



وزارة البحث العلمي والتعليم العالي
MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET
DE LA RECHERCHE SCIENTIFIQUE



جامعة عبد الحميد بن باديس مستغانم
Université Abdelhamid Ibn Badis Mostaganem
كلية العلوم و التكنولوجيا
Faculté des Sciences et de la Technologie
DEPARTEMENT DE GENIE ELECTRIQUE

MEMOIRE

Présenté pour obtenir le diplôme de

MASTER EN ÉLECTRONIQUE

Par :

MOKHTARI abderrahmen

BELDJILALI meriem zohra

Spécialité: Électronique des Systèmes Embarqués

Intitulé du sujet

Mise en œuvre d'un système d'anonymisation et de traçabilité des copies d'examen par code-barres et intelligence artificielle

Soutenu le 06 / 07/2026 devant le jury composé de :

Président	ABDERRAHMEN Abdelkader	Grade MCB	Université de Mostaganem
Examineur	BENTOUMI Mohamed	Grade MCA	Université de Mostaganem
Représentant de l'incubateur	BOUZIANE Larbi	Grade MCB	Université de Mostaganem
Encadreur	AMAMRA Laid	Grade MA	École Normale Supérieure de Mostaganem
Co-encadreur	BENOUALI Abdelhak	Grade MAA	Université de Mostaganem
Partenaire économique	HASSINI Mustapha	Grade MCA	UFC Mostaganem

Année Universitaire 2025/2026

RESUME

Dans ce mémoire, nous présentons la conception, le développement et l'utilisation d'ANONYMOS OS, une plateforme intelligente dédiée à la gestion automatisée, sécurisée et anonyme des examens. Ce travail s'inscrit dans le cadre de la modernisation des processus administratifs au sein des établissements de l'enseignement supérieur et vise à répondre à deux problématiques majeures : l'absence d'un mécanisme garantissant l'anonymat lors de la correction des copies d'examen et la lenteur des traitements manuels liés à la gestion des notes. La solution proposée repose sur l'intégration de trois briques technologiques. La première est une couche matérielle basée sur le module embarqué ESP32-S3-CAM (intégré d'un capteur d'image OV3660) chargé de la numérisation et de la transmission sans fil des copies d'examen via le protocole Wi-Fi. La deuxième est une couche logicielle développée en Python, comprenant un serveur Flask destiné à la réception et au traitement des images, une base de données SQLite qui assure la centralisation des données académiques et une interface web conviviale réalisée à l'aide du framework Streamlit. Enfin, la troisième brique est reposée sur un ensemble d'algorithmes spécialisés, notamment la génération sécurisée des codes-barres d'anonymisation basés sur le HMAC d'une part et l'extraction automatique des notes par le biais de l'analyse de la densité de pixels (Pixel Density Analysis) sur les grilles de notation d'autre part.

Le fonctionnement global du système s'articule autour de cinq phases principales. La première consiste en la préparation du système et la génération des identifiants anonymes sous forme de codes-barres. La deuxième phase concerne l'organisation de l'examen et l'acquisition des images des copies à l'aide du module ESP32-S3-CAM. La troisième phase est dédiée au décodage automatique des codes-barres et à l'extraction des notes. La quatrième étape porte sur les opérations de délibération et de levée de l'anonymat. Enfin, la cinquième phase assure la génération automatique des procès-verbaux au format PDF ainsi que leur transmission aux enseignants responsables via le protocole SMTP.

Les résultats obtenus démontrent l'efficacité fonctionnelle et la fiabilité de la plateforme dans la gestion des examens universitaires tout en garantissant la confidentialité des étudiants et l'automatisation des traitements administratifs. Par ailleurs, ce travail ouvre des perspectives prometteuses, notamment l'évolution vers une architecture Cloud multi-utilisateurs et l'intégration de techniques avancées d'intelligence artificielle, telles que les réseaux de neurones convolutifs, afin d'améliorer davantage les performances et les capacités d'analyse du système.

الملخص

في هذه المذكرة، نقدم تصميم وتطوير واستخدام منصة ANONYMOS OS وهو منصة ذكية للإدارة الآلية والأمن والمجهولة الهوية للامتحانات. يندرج هذا العمل في إطار تحديث العمليات الإدارية داخل مؤسسات التعليم العالي، ويهدف إلى معالجة إشكاليتين رئيسيتين تتمثلان في غياب آلية تضمن إخفاء هوية الطلبة أثناء تصحيح أوراق الامتحانات، وبطء المعالجات اليدوية المرتبطة بإدارة العلامات. تعتمد الحلول المقترحة على دمج ثلاث لبنات تكنولوجية متكاملة. تتمثل الأولى في طبقة مادية تعتمد على الوحدة المدمجة (ESP32-S3-CAM) المزودة بمستشعر صور (OV3660) والمسؤولة عن المسح الضوئي اللاسلكي لأوراق الامتحانات عبر بروتوكول Wi-Fi. أما الثانية فهي طبقة برمجية مطورة بلغة Python، تضم خادم مبنياً على إطار العمل Flask مخصصاً لاستقبال الصور ومعالجتها، وقاعدة بيانات SQLite تضمن مركزية البيانات الأكاديمية وإدارتها، إضافة إلى واجهة ويب تفاعلية وسهلة الاستخدام تم تطويرها باستخدام إطار العمل Streamlit. وأخيراً، تم بناء اللبنة الثالثة فتعتمد على مجموعة من الخوارزميات المتخصصة في المعالجة: تشمل التوليد الآمن لرموز الشريطية الخاصة بإخفاء الهوية استناداً إلى HMAC، بالإضافة إلى الاستخراج التلقائي للدرجات من خلال تحليل كثافة البكسل (Pixel Density Analysis) المطبق على شبكات التقييم.

يرتكز التشغيل العام للنظام على خمس مراحل رئيسية. تتمثل المرحلة الأولى في إعداد النظام وتوليد المعرفات المجهولة على شكل رموز شريطية. أما المرحلة الثانية فتخصص لتنظيم الامتحان واقتناء صور أوراق الإجابة باستخدام وحدة ESP32-S3-CAM. وتُعنى المرحلة الثالثة بفك ترميز الرموز الشريطية واستخراج العلامات بصورة آلية. بينما تتعلق المرحلة الرابعة بعمليات المداولات ورفع السرية عن الهوية. وأخيراً، تضمن المرحلة الخامسة التوليد التلقائي لمحاضر النتائج بصيغة PDF وإرسالها إلى الأستاذة المسؤولين عبر بروتوكول SMTP.

تُظهر النتائج المتحصل عليها الفعالية الوظيفية والموثوقية العالية للمنصة في إدارة الامتحانات الجامعية، مع ضمان سرية هوية الطلبة وأتمتة مختلف المعالجات الإدارية. كما يفتح هذا العمل آفاقاً واعدة للتطوير مستقبلاً، من خلال الانتقال إلى بنية سحابية متعددة المستخدمين (Cloud Multi-Users) ودمج تقنيات متقدمة في مجال الذكاء الاصطناعي، مثل الشبكات العصبية الالتفافية (Convolutional Neural Networks - CNNs)، بهدف تحسين الأداء وتعزيز قدرات التحليل والمعالجة داخل النظام..

Summary

In this master's thesis, we present the design, development, and implementation of **ANONYMOS OS**, an intelligent platform dedicated to the automated, secure, and anonymous management of examinations. This work is part of the modernization of administrative processes within higher education institutions and aims to address two major issues: the absence of a mechanism ensuring anonymity during the grading of examination papers and the slowness of manual procedures related to grade management.

The proposed solution is based on the integration of three technological components. The first is a hardware layer built around the ESP32-S3-CAM embedded module, equipped with an OV3660 image sensor, which is responsible for the digitization and wireless transmission of examination papers through the Wi-Fi protocol. The second is a software layer developed in Python, consisting of a Flask server for image reception and processing, an SQLite database for the centralization of academic data, and a user-friendly web interface developed using the Streamlit framework. Finally, the third component relies on a set of specialized algorithms, including the secure generation of anonymization barcodes based on HMAC technology and the automatic extraction of grades through Pixel Density Analysis applied to grading grids.

The overall operation of the system is organized into five main phases. The first phase consists of system preparation and the generation of anonymous identifiers in the form of barcodes. The second phase involves examination management and the acquisition of examination paper images using the ESP32-S3-CAM module. The third phase is dedicated to the automatic decoding of barcodes and the extraction of grades. The fourth phase concerns deliberation procedures and the removal of anonymity. Finally, the fifth phase ensures the automatic generation of PDF reports and their transmission to the responsible instructors via the SMTP protocol.

The obtained results demonstrate the functional effectiveness and reliability of the platform in managing university examinations while ensuring student confidentiality and automating administrative processes. Furthermore, this work opens promising perspectives, particularly the evolution toward a multi-user cloud-based architecture and the integration of advanced artificial intelligence techniques, such as convolutional neural networks, to further improve the system's performance and analytical capabilities.

Remerciement

Nous tenons à exprimer notre profonde gratitude à notre encadreur, **M. AMAMRA Laid**, ainsi qu'à notre co-encadreur, **M. BENOUALI Abdelhak**, pour leur accompagnement, leurs précieux conseils, leur disponibilité et leur soutien tout au long de la réalisation de ce projet. Nous adressons également nos remerciements à notre partenaire économique, **M. HASSINI Mustapha**, pour son accompagnement et son soutien qui ont contribué à la réussite de ce travail. Leurs orientations et leur expertise ont largement contribué à l'aboutissement de ce projet.

Nous adressons également nos sincères remerciements aux membres du jury, **M. ABDERRAHMEN Abdelkader**, président du jury, **M. BENTOUMI Mohamed**, examinateur, et le représentant de l'incubateur **M. BOUZIANE Larbi**, pour l'honneur qu'ils nous font en acceptant d'évaluer notre travail ainsi que pour leurs remarques et recommandations constructives.

Nous exprimons notre reconnaissance À mes amis **BOUDANI Mohamed Toufik** et **HAMCHRIF Amir Youcef** pour leur aide, leur disponibilité et leur précieuse contribution à l'organisation et au bon déroulement de ce projet.

Enfin, nous adressons nos plus sincères remerciements à nos familles et à nos proches pour leur patience, leurs encouragements et leur soutien indéfectible tout au long de cette période.

Dédicace Par Abderrahmane Mokhtari

Ce travail est le fruit d'un long cheminement qui n'aurait pu aboutir sans le soutien précieux de nombreuses personnes. C'est avec une profonde gratitude que je leur dédie ces quelques lignes.

À mes chers parents,

Qui ont été mon soutien dans les moments de doute,

Qui ont partagé mes efforts et nos nuits de travail par leur présence silencieuse,

Et qui n'ont jamais cessé de croire en moi, même lorsque je doutais moi-même.

Votre amour inconditionnel a été ma plus grande force.

À mes amis,

Pour leur soutien, leurs encouragements, leurs paroles réconfortantes et leur présence tout au long de ce parcours.

Leur aide, de près comme de loin,

a été précieuse et a contribué à la réalisation de ce travail.

À mes enseignants et mentors,

Qui ont éveillé en moi le goût du savoir,

Nourri ma curiosité intellectuelle

Et m'ont guidé tout au long de mon parcours.

Enfin, à tous ceux qui, de près ou de loin,

Ont contribué à la réalisation de ce projet,

J'exprime ma sincère reconnaissance.

Ce travail est autant le vôtre que le mien.

Dédicace Par Beldjilali Meriem Zohra

À mes chers parents,

*Pour leur amour infini et leurs sacrifices silencieux,
pour chaque regard bienveillant et chaque mot d'encouragement qui m'ont portée jusqu'ici.*

Vous avez été ma lumière dans les moments les plus sombre.

Ce travail est, avant tout, le vôtre.

À ma famille,

Pour sa chaleur, sa tendresse et son soutien indéfectible.

Savoir que vous étiez là a rendu chaque effort plus léger.

À mes amis,

*Pour leur fidélité et leur soutien sans faille, pour avoir partagé ce chemin moi et illuminé les
moments les plus difficiles de leur simple présence.*

À mes enseignants,

*Qui ont semé en moi le gout du savoir et de la rigueur,
et dont la passion pour la transmission a guidé chacun de mes pas*

À toutes les personnes qui ont croisé mon chemin,

De près ou de loin, d'une façon ou d'une autre,

Et qui ont laissé une trace dans ce parcours,

Je dédie ce travail avec toute ma reconnaissance et ma gratitude.

Liste des abréviations

- ADC / CAN : Analog-to-Digital Converter / Convertisseur Analogique-Numérique
- AES : Advanced Encryption Standard
- BLE : Bluetooth Low Energy
- CAM : Camera Module (Module Caméra)
- CCD : Charge-Coupled Device
- CMOS : Complementary Metal Oxide Semiconductor
- CPU : Central Processing Unit (Unité Centrale de Traitement)
- DAC / CNA : Digital-to-Analog Converter / Convertisseur Numérique-Analogique
- DMA : Direct Memory Access
- DVP : Digital Video Port
- ECC : Elliptic Curve Cryptography (Cryptographie à Courbe Elliptique)
- EEPROM : Electrically Erasable Programmable Read-Only Memory
- ESP32-S3 : Espressif Systems Platform 32 – Série S3 (SoC Xtensa LX7)
- FPGA : Field-Programmable Gate Array
- FPU : Floating Point Unit (Unité à Virgule Flottante)
- GPIO : General Purpose Input/Output (Entrée/Sortie à Usage Général)
- I2C : Inter-Integrated Circuit (Bus de Communication Série)
- I2S : Inter-IC Sound (Interface Audio Numérique)
- IoT : Internet of Things (Internet des Objets)
- LED : Light-Emitting Diode (Diode Électroluminescente)
- MCU : Micro-Controller Unit (Microcontrôleur)
- OV3660 : OmniVision 3660 — Capteur d'image CMOS couleur
- PSRAM : Pseudo Static Random Access Memory
- PWM : Pulse Width Modulation (Modulation de Largeur d'Impulsion)
- RAM : Random Access Memory (Mémoire Vive)
- RFID : Radio Frequency Identification (Identification par Radiofréquence)
- RISC-V : Reduced Instruction Set Computing – 5ème génération
- ROM : Read-Only Memory (Mémoire Morte)
- RSA : Rivest–Shamir–Adleman (Algorithme de Chiffrement Asymétrique)

- SDIO : Secure Digital Input Output
- SHA : Secure Hash Algorithm (Algorithme de Hachage Sécurisé)
- SoC : System on Chip (Système sur Puce)
- SPI : Serial Peripheral Interface (Interface Périphérique Série)
- SRAM: Static Random Access Memory (Mémoire Statique Vive)
- ULP : Ultra Low Power (Coprocesseur à Ultra Faible Consommation)
- USB : Universal Serial Bus (Bus Série Universel)
- USB-OTG : Universal Serial Bus On-The-Go
- UART : Universal Asynchronous Receiver-Transmitter (Émetteur-Récepteur Asynchrone Universel)
- API : Application Programming Interface (Interface de Programmation Applicative)
- CNN : Convolutional Neural Network (Réseau de Neurones Convolutif)
- EAN : European Article Number (Numéro d'Article Européen)
- ERP : Enterprise Resource Planning (Progiciel de Gestion Intégrée)
- FreeRTOS : Free Real-Time Operating System (Système d'Exploitation Temps Réel Libre)
- GS1 : Global Standards 1 (Organisation Mondiale de Standardisation des Codes-Barres)
- HMAC : Hash-based Message Authentication Code (Code d'Authentification de Message Basé sur le Hachage)
- HTTP : HyperText Transfer Protocol (Protocole de Transfert Hypertexte)
- IA : Intelligence Artificielle (Artificial Intelligence – AI)
- IEC : International Electrotechnical Commission (Commission Électrotechnique Internationale)
- IEEE : Institute of Electrical and Electronics Engineers
- ISO : International Organization for Standardization (Organisation Internationale de Normalisation)
- IU : Interface Utilisateur (User Interface – UI)
- IUG : Interface Utilisateur Graphique (Graphical User Interface – GUI)
- JPEG : Joint Photographic Experts Group (Format d'Image Numérique Compressé)
- OS : Operating System (Système d'Exploitation)

- PC : Personal Computer (Ordinateur Personnel)
- PDF : Portable Document Format
- QR Code : Quick Response Code (Code de Réponse Rapide — Code 2D)
- REST : Representational State Transfer (Architecture de Services Web)
- RNG : Random Number Generator (Générateur de Nombres Aléatoires)
- RTOS : Real-Time Operating System (Système d'Exploitation Temps Réel)
- SNR : Signal-to-Noise Ratio (Rapport Signal sur Bruit)
- SQL : Structured Query Language (Langage de Requête Structuré)
- SMTP : Simple Mail Transfer Protocol (Protocole Simple de Transfert de Courrier)
- UML : Unified Modeling Language (Langage de Modélisation Unifié)
- UPC : Universal Product Code (Code Produit Universel)
- YOLO : You Only Look Once (Algorithme de Détection d'Objets en Temps Réel)
- CC : Contrôle Continu (Continuous Assessment)
- ID : Identifiant unique (Unique Identifier)

Liste des figures

Figure I-1 : présente quelques produits embarqués. Ce n'est qu'un échantillon parmi l'immense galaxie des systèmes embarqués.	5
Figure I-2 : Architecture matérielle interne de l'ESP32.....	10
Figure I-3 : schema Block Esp32-s3	13
Figure I-4 : ESP32-S3-Cam.....	15
Figure I-5 : Structure d'un code-barre 1D.....	22
Figure I-6 : Type de codes-barres et symbologies	25
Figure I-7 : La différence entre les codes-barres 1D et 2D	26
Figure I-8 : Chaîne d'acquisition d'image	30
Figure I-9 : Principe de fonctionnement d'un capteur CCD	32
Figure I-10 : Principe de fonctionnement du capteur CMOS	34
Figure III-1:UML DATABASE	65
Figure III-2 ESP32-S3-CAM OV 3660	68
Figure III-3 Aperçu du fichier PDF de présence des étudiants	73
Figure III-4 Chaîne de traitement appliquée aux images des copies d'examen pour la détection des zones de notation et l'identification automatique des notes.	76
Figure III-5: Interface de sécurité du système.	78
Figure III-6: Interface d'exécution et de contrôle du système.	80
Figure III-7: Interface de réception des copies d'examen.	81
Figure III-8: Interface de consultation des données.	82
Figure III-9: Consultation des fichiers PDF dans la base de données	83
Figure III-10: Table de contrôle des résultats dans la base de données	84
Figure III-11: Interface de support et de contrôle du système.	85
Figure III-12 : Diagramme de Cas d'Utilisation du systeme	87
Figure III-13: Diagramme d'activités du système.	89
Figure III-14: Diagramme de séquence du système.....	92
Figure IV-1 Interface d'authentification des utilisateurs	99
Figure IV-2 : Table des examens de la base de données	100
Figure IV-3 : Interface avant l'initialisation des codes-barres.....	101
Figure IV-4 : Interface après l'initialisation des codes-barres	101
Figure IV-5 : Table des examens après la mise à jour des données	102
Figure IV-6 : Aperçu PDF avant la génération des documents	103
Figure IV-7 Interface après la génération du fichier PDF.	103
Figure IV-8Interface avant la génération du fichier PDF	103
Figure IV-9 Aperçu PDF après la génération du fichier PDF	104
Figure IV-10 contrôleur de réception des copies avant le démarrage du traitement.	105
Figure IV-11 contrôleur de réception des copies après le démarrage du processus	105
Figure IV-12 Interface de consultation des copies reçues.	106
Figure IV-13 Image d'une copie d'examen à traiter	107
Figure IV-14 Interface de consultation des copies archivées	108
Figure IV-15 Interface après le décodage des copies d'examen.	109

Figure IV-16 Interface avant le lancement du décodage.....	109
Figure IV-17 Table des notes dans la base de données avant délibération.	111
Figure IV-18 Interface avant le lancement de la délibération.	112
Figure IV-19 Interface après le lancement de la délibération.	112
Figure IV-20 Table des notes dans la base de données après délibération	113
Figure IV-21 Interface après l'envoi des résultats.	114
Figure IV-22 Interface avant l'envoi des résultats.....	114
Figure IV-23 Liste des fichiers PDF des modules archivés.	115
Figure IV-24 fichier PDF généré après sélection du module.....	115
Figure IV-25 Courriel envoyé au professeur via Gmail	116

Liste des tableaux

Tableau 1 : ESP32-S3 Cam OV3660 vs Arduino	19
Tableau 2 Investissement.....	129
Tableau 3 Masse salariale	129
Tableau 4 Chiffre d'affaires.....	129
Tableau 5 Grille tarifaire Des Licences Annuelles	130
Tableau 6 Charges externes	130
Tableau 7 Besoin en Fonds de Roulement.....	130
Tableau 9 Flux de Trésorerie	131
Tableau 8 compte de résultat	131

Sommaire

RESUME	I
Remerciement	IV
Dédicace Par Abderrahmane Mokhtari.....	V
Dédicace Par Beldjilali Meriem Zohra	VI
Liste des abréviations.....	VII
Liste des figures	X
Liste des tableaux.....	XII
Sommaire	XIII
Introduction générale	XVI
Chapitre I: Généralités sur les systèmes embarqués et la technologie des codes-barres	1
I.1 Introduction	1
I.2 Les systèmes embarqués	1
I.2.1 Définition	1
I.2.2 Caractéristiques des systèmes embarqués	3
I.2.3 Domaine d'application des systèmes embarqués	4
I.2.4 Comparaison entre un système embarqué et un ordinateur à usage général.....	5
I.3 La plateforme ESP32	7
I.3.1 Architecture ESP32	8
I.3.2 Caractéristiques techniques d'ESP32 :	10
I.3.3 Module ESP32-S3-CAM.....	12
I.3.4 Caractéristiques :	13
I.3.5 AVANTAGES :	15
I.3.6 Comparaison avec Arduino :	17
I.4 Technologie des codes-barres :	19
I.4.1 Définition et principe de fonctionnement :	20
I.4.2 Types de codes-barres et symbologies :	23
I.4.3 Différence entre code-barres et QR Code	25
I.4.4 Applications pratiques des codes-barres dans la gestion d'actifs	27
I.4.5 Applications sectorielles des codes-barres	28
I.5 Acquisition d'image dans les systèmes embarqués :	29
I.5.1 Capteurs d'image	30
I.5.2 Contraintes liées à l'éclairage	34
I.5.3 Résolution et qualité d'image	35

I.6	Travaux existants et solutions similaires.....	38
I.6.1	Système utilisé en Algérie :.....	38
I.6.2	Limites des solutions actuelles :	39
I.6.3	Limites liées à la connectivité et à l'architecture système	41
I.6.4	Limitations des solutions mobiles grand public	41
I.6.5	Faible diffusion des technologies IA dans les solutions locales	41
I.7	Conclusion :.....	42
I.7.1	Perspectives de travaux futurs :.....	42
Chapitre II: Analyse et spécifications du système		44
II.1	Introduction	44
II.2	Présentation de la problématique	45
II.3	Cahier des charges fonctionnel.....	46
II.4	Solution proposée	47
II.4.1	Préparation du système :	51
II.4.2	Gestion de l'examen	53
II.4.3	Post-examen et délibération	55
II.5	Spécifications fonctionnelles.....	56
II.6	Spécifications non fonctionnelles	57
II.7	Description globale du système	57
II.8	Conclusion.....	58
Chapitre III: Conception du système Anonymos OS.....		60
III.1	Introduction :.....	60
III.2	Architecture globale du système :	61
III.2.1	Schema bloc du système :	61
III.2.2	Interactions entre les différents modules :.....	66
III.3	Conception Matérielle :.....	67
III.3.1	Schema de L'ESP32-S3-CAM :.....	67
III.3.2	Alimentation :.....	68
III.3.3	Interface camera :	69
III.4	Conception logicielle :	69
III.4.1	Architecture du frimwar embarqué :	69
III.4.2	Architecture logicielle coté pc :	71
III.5	Modélisation UML :.....	87

III.6	Communication entre l'ESP32 et le PC:	93
III.6.1	Protocole de communication :	94
III.6.2	Format des données échangées.....	95
III.7	Conclusion :.....	97
Chapitre IV: Protocol test ANONYMOS OS		98
IV.1	Introduction	98
IV.2	Phase de sécurisation, d'authentification et d'accès global à l'application ANONYMOSOS	98
IV.3	Procédure d'Exécution du Module de Génération de Codes-barres D'anonymisation (Barcode)	99
IV.3.1	Phase d'avant-exécution : État initial du système	100
IV.4	Protocole d'exécution du module de génération des fiches PDF.....	102
IV.5	Protocole d'activation du serveur de réception d'images (ESP32-S3-CAM)	104
IV.6	Protocole d'exécution du module de traitement d'image, de décodage et d'extraction des notes automatique :	108
IV.7	Protocole d'exécution du module de délibération et de levée d'anonymat	111
IV.8	Protocole d'exécution du module d'exportation PDF par module et d'expédition automatisée (SEND RESULTS)	113
IV.9	Perspectives d'évolution et futurs développements d'ANONYMOS OS	116
IV.9.1	Migration vers le Cloud et architecture relationnelle multi-utilisateurs	116
IV.9.2	Optimisation du traitement d'image par réseaux de neurones convolutifs et numérisation industrielle	117
IV.9.3	Amélioration du terminal IoT et standardisation de l'acquisition visuelle.....	117
IV.9.4	Intégration d'une extension mobile multiplateforme pour les usagers.....	117
IV.10	Conclusion :.....	118
Références.....		120
ANNEXE		125
Core de programme		125
Business Model Canvas (modèle d'affaire)		128
Business Plan.....		129
Détection des erreurs « ambigu » dans l'analyse des pixels.....		132

Introduction générale

L'évolution des technologies numériques et des systèmes embarqués a profondément modifié les méthodes de gestion de l'information dans de nombreux domaines, notamment celui de l'enseignement supérieur. Aujourd'hui, les universités cherchent à moderniser leurs processus administratifs afin d'améliorer leur efficacité, de réduire les erreurs humaines et de garantir davantage de transparence dans le traitement des données académiques. Parmi les activités concernées par cette transformation figure la gestion des examens, qui constitue un élément essentiel du processus d'évaluation des étudiants.

Dans de nombreux établissements universitaires algériens, la gestion des examens repose encore largement sur des procédures manuelles. La correction des copies s'effectue généralement sans mécanisme d'anonymisation, tandis que la saisie et le traitement des notes nécessitent des opérations répétitives pouvant entraîner des erreurs et allonger les délais de publication des résultats. Cette situation soulève des préoccupations liées à l'équité de l'évaluation, à la traçabilité des informations et à l'efficacité globale du processus administratif.

Les avancées réalisées dans le domaine des systèmes embarqués et des technologies d'identification automatique offrent aujourd'hui des solutions adaptées à ces problématiques. Les modules intelligents tels que l'ESP32-S3-CAM permettent l'acquisition et la transmission de données en temps réel, tandis que les codes-barres facilitent l'identification et le suivi automatisé des documents. Associées aux outils logiciels modernes et aux techniques de traitement d'image, ces technologies contribuent à l'automatisation et à la sécurisation des opérations académiques.

C'est dans ce contexte que s'inscrit ce projet de fin d'études, consacré à la conception et au développement d'ANONYMOS OS, une plateforme intelligente destinée à la gestion automatisée et sécurisée des examens universitaires. La solution proposée vise à garantir l'anonymat des copies, à automatiser leur numérisation, à extraire automatiquement les notes et à centraliser les données académiques au sein d'un environnement logiciel unique. Elle permet également la génération et l'envoi automatisés des procès-verbaux, améliorant ainsi la fiabilité et l'efficacité du processus d'évaluation.

Ce mémoire est structuré en quatre chapitres. Le premier présente les notions générales relatives aux systèmes embarqués et aux codes-barres. Le deuxième expose l'analyse et les spécifications du système proposé. Le troisième détaille la conception et l'architecture de la plateforme ANONYMOS OS. Enfin, le quatrième chapitre est consacré aux tests réalisés ainsi qu'à la validation de la solution développée.

Chapitre I: Généralités sur les systèmes embarqués et la technologie des codes-barres

Chapitre I:Généralités sur les systèmes embarqués et la technologie des codes-barres

I.1 Introduction

Dans un monde de plus en plus connecté, les systèmes embarqués occupent une place centrale dans notre quotidien, allant des appareils domestiques aux équipements industriels et médicaux. Ces systèmes, constitués d'un ensemble matériel et logiciel optimisé pour une tâche spécifique, permettent d'automatiser, de contrôler et de faciliter de nombreuses opérations sans nécessiter une intervention humaine permanente. Le développement de microcontrôleurs puissants et économes en énergie, tels que la plateforme ESP32, a ouvert la voie à des applications avancées, notamment dans le domaine de l'Internet des objets (IoT) et de la vision embarquée.

Parallèlement, la technologie des codes-barres s'est imposée comme un standard mondial de gestion d'inventaire et traçabilité des produits, en offrant une solution fiable, économique et universelle pour l'identification et le suivi des actifs. Bien que le code-barres reste largement dominant en Algérie, notamment à travers le standard GS1 et le format EAN-13, ses limites en matière de traçabilité, de sécurité et de compatibilité avec les technologies émergentes (RFID, Intelligence Artificielle (IA), blockchain) deviennent de plus en plus évidentes.

Ce chapitre présente les notions fondamentales relatives aux systèmes embarqués, leurs caractéristiques, leurs architectures et leurs domaines d'applications. Il aborde ensuite la technologie des codes-barres, en détaillant ses types, son fonctionnement, ses avantages et ses limites, ainsi que son rôle dans le contexte algérien. L'objectif est de fournir une base conceptuelle solide permettant de mieux comprendre les défis et les opportunités liés à l'intégration des systèmes embarqués dans les solutions modernes de gestion et de suivi des produits.

I.2 Les systèmes embarqués

I.2.1 Définition

Un système embarqué peut être défini comme un système informatique spécialisé, intégré dans un dispositif électronique ou électromécanique, conçu pour exécuter une fonction précise et prédéfinie. Contrairement à un ordinateur à usage général, destiné à exécuter une grande variété d'applications, le système embarqué est dédié à une tâche spécifique. Cette spécialisation permet une optimisation poussée des ressources matérielles et logicielles, garantissant ainsi un fonctionnement fiable, déterministe et souvent en temps réel.

Chapitre I: Généralités sur les systèmes embarqués et la technologie des codes-barres

La présence conjointe de matériel et de logiciel fait d'un système embarqué une forme particulière d'ordinateur, mais il n'exécute qu'un ensemble limité de fonctions. Ainsi, l'on exclut le PC du monde des systèmes embarqués et il est considéré comme un ordinateur à usage général. Par conséquent, un système embarqué est une unité de calcul « à usage spécifique », dotée d'un processeur et d'un logiciel associé. Le logiciel lié à l'application est généralement stocké dans une mémoire non volatile (ROM); il est donc plus approprié de l'appeler un « micrologiciel » (firmware).[\[1\]](#)

Sur le plan matériel, un système embarqué repose généralement sur un microcontrôleur ou un microprocesseur, accompagné de différentes mémoires comme la RAM pour le stockage temporaire des données, la mémoire Flash pour le programme, et parfois une mémoire EEPROM pour la conservation de paramètres permanents. À cela s'ajoutent des interfaces d'entrée/sortie (GPIO, ADC, DAC, SPI, I2C, UART) permettant l'interaction avec des capteurs, des actionneurs et d'autres périphériques. Le logiciel embarqué, souvent appelé firmware, constitue la couche logicielle chargée de contrôler le matériel et d'implémenter les fonctionnalités du système.

Les systèmes embarqués sont bien des systèmes informatiques, mais leur interface utilisateur (IU) peut varier de manière importante, allant de l'absence totale d'IU – par exemple, sur les appareils conçus pour une seule tâche – jusqu'à des interfaces utilisateur graphiques (IUG) complexes, comme sur les appareils mobiles. Les interfaces utilisateur peuvent comprendre des boutons, des écrans à diodes électroluminescentes (DEL) et des écrans tactiles. Certains systèmes utilisent également des interfaces utilisateur à distance.[\[1\]](#)

Selon Global Markets Insight, la valeur du marché des systèmes embarqués s'élevait à 110,3 milliards de dollars en 2023 et devrait dépasser les 190 milliards de dollars d'ici à l'année 2032. Parmi les entreprises technologiques de pointe qui fabriquent des puces pour les systèmes embarqués, on compte Apple, IBM, Intel et Texas Instruments. Cette augmentation attendue s'explique en partie par le maintien des investissements dans le domaine de l'intelligence artificielle (IA), de l'informatique mobile et de la nécessité de circuits intégrés pour un traitement de données hautement performant.[\[2\]](#)

Chapitre I: Généralités sur les systèmes embarqués et la technologie des codes-barres

I.2.2 Caractéristiques des systèmes embarqués

- **Exécuter une fonction précise :**

Les systèmes embarqués sont conçus pour réaliser une fonction spécifique et dédiée. Ni plus ni moins. Un ordinateur de bureau peut lancer de nombreuses applications simultanément, alors qu'un contrôleur embarqué dans un système de freinage a une seule mission : détecter les conditions et réagir au quart de tour. Toute décision architecturale s'inscrit dans cet unique objectif.

- **Travailler dans le cadre de contraintes strictes :**

Les systèmes embarqués fonctionnent également avec de très faibles ressources. Leur puissance de traitement est souvent limitée, leur mémoire minimale et leurs budgets énergétiques restreints. Les ingénieurs doivent écrire un code très performant, tout en trouvant un équilibre pointu entre la performance et les limites du matériel. L'optimisation n'est pas un luxe. C'est une obligation.

- **Les délais d'exécution :**

le système embarqué doit répondre aux événements critiques dans un délai strictement défini, appelé **échéance (deadline)**. Une garantie de respect de tous les délais ne peut être donnée que si le comportement du système d'exploitation peut être prévisible. En d'autres termes, le temps de réponse doit être déterministe.

Dans un système à temps réel dur (hard real-time), toute réponse tardive est considérée comme une défaillance critique, comme c'est le cas dans les systèmes de contrôle automobile ou médicaux. Dans un système à temps réel souple (soft real-time), un léger retard peut être toléré sans compromettre gravement le fonctionnement global.

- **Fiabilité sur le long terme :**

Les systèmes embarqués se caractérisent également par la fiabilité et le déterminisme. Ces systèmes fonctionnent souvent pendant des années sans redémarrage, et ce dans des conditions difficiles. Leur comportement doit être prévisible, tolérer les pannes et rester stable durant tout le cycle de vie. Les systèmes embarqués sont généralement déployés dans des environnements où les mises à jour sont complexes, coûteuses ou soumises à une réglementation stricte, contrairement aux logiciels grand public qui peuvent être fréquemment mis à jour.

Chapitre I: Généralités sur les systèmes embarqués et la technologie des codes-barres

- **Intégration matérielle étroite :**

Enfin, les systèmes embarqués sont fortement intégrés au matériel. Ils contrôlent directement les capteurs, les actionneurs, les moteurs et les interfaces de communication. Le matériel et le logiciel sont conçus et optimisés de façon conjointe, puis sont validés comme un système unifié. Cette intégration poussée et les graves conséquences d'une défaillance distinguent fondamentalement les systèmes embarqués de l'informatique générale. [3]

I.2.3 Domaine d'application des systèmes embarqués

Comme vous l'avez sans doute compris : les systèmes embarqués réalisent des fonctions, au sein d'un dispositif. Mais de quelles fonctions parle-t-on exactement ? En fait, tout dépend de l'équipement dans lequel le système a été intégré : matériel de santé, appareils ménagers, ordinateur, informatique, transports... On retrouve les systèmes embarqués dans de nombreux domaines, au point qu'il nous arrive de les côtoyer tous les jours sans nous en apercevoir. Voici quelques exemples de domaines d'application :

- **Électronique grand public :** appareils photo, lecteurs de musique, téléviseurs, lecteurs DVD, fours à micro-ondes, machines à laver, réfrigérateurs et télécommandes.
- **Appareils électroménagers / systèmes de sécurité domestique :** climatiseurs, systèmes d'alarme contre les intrusions et les incendies domestiques.
- **Contrôle automobile :** système de freinage antiblocage (ABS), contrôle du moteur et de la transmission, commande des portes et des essuie-glaces, etc.
- **Appareils portables :** téléphones mobiles, assistants numériques personnels (PDA), lecteurs MP3, appareils photo numériques, etc.
- **Équipements médicaux :** scanners, appareils ECG et EEG, équipements de test et de surveillance.
- **Secteur bancaire :** distributeurs automatiques de billets (DAB), compteuses de billets, etc.
- **Périphériques informatiques :** imprimantes, scanners, webcams, etc.
- **Réseaux :** routeurs, commutateurs (switches), concentrateurs (hubs), etc.
- **Usines :** systèmes de contrôle, d'automatisation, d'instrumentation et d'alarme.
- **Aviation :** commandes d'avion, systèmes de guidage et d'instrumentation.

Chapitre I: Généralités sur les systèmes embarqués et la technologie des codes-barres

- **Militaire** : contrôle et surveillance des équipements militaires.
- **Robotique** : utilisée dans les usines, les foyers et les activités de loisir.
- Jouets. [4]



Figure I-1 : présente quelques produits embarqués. Ce n'est qu'un échantillon parmi l'immense galaxie des systèmes embarqués.

Cette liste est non exhaustive et, en la parcourant, on pourrait avoir l'impression que tout ce qui implique un contrôle électronique moderne est un « système embarqué ». Ce n'est effectivement pas très éloigné de la réalité. En fait, le seul équipement électronique généralement exclu de cette catégorie est l'ordinateur personnel domestique (PC).

I.2.4 Comparaison entre un système embarqué et un ordinateur à usage général

Qu'est-ce qui définit un ordinateur à usage général ?

Un ordinateur à usage général est une machine capable de réaliser de nombreuses tâches, tels que des calculs, du traitement de texte, des communications ou des analyses. Ces systèmes font

Chapitre I:Généralités sur les systèmes embarqués et la technologie des codes-barres

fonctionner des systèmes d'exploitation "standards" comme Windows, macOS ou des distributions Linux, conçus pour fonctionner avec un maximum de matériel plutôt que pour répondre aux exigences d'un industriel.

Les ordinateurs généraux sont basés sur du matériel standard, comme les processeurs, la mémoire et les systèmes de stockage grand public, qui favorisent une large compatibilité plutôt qu'un rendement spécifique. Ces systèmes font l'objet de mises à jour et de correctifs réguliers du système d'exploitation, ce qui, bien qu'excellent pour la sécurité, peut entraîner des perturbations imprévisibles dans les environnements industriels. Elles sont conçues pour être polyvalentes et compatibles avec un grand nombre de logiciels, ce qui en fait un excellent choix pour les environnements de bureau où l'on privilégie la flexibilité aux fonctionnalités spécialisées. Cette approche a toutefois pour inconvénient une durée de vie généralement plus courte, de 3 à 5 ans, les composants devenant obsolètes et le support diminuant.[\[5\]](#)

Qu'est-ce qui différencie les systèmes embarqués ?

Un système embarqué est un système informatique spécialisé et techniquement conçu pour exécuter des tâches spécifiques ou un ensemble restreint de fonctions dans une solution dédiée. Par exemple, dans le cas d'un lecteur de codes-barres basé sur ESP32, le système est uniquement destiné à capturer une image, traiter les données capturées et transmettre le résultat via une interface de communication. Cette spécialisation permet de réduire considérablement la complexité logicielle et matérielle.

La différence principale tient à la conception, les systèmes embarqués industriels étant conçus de façon pointue pour accomplir des tâches ou des besoins spécifiques, assurant ainsi leur fiabilité et la constance de leur fonctionnement.

Les systèmes embarqués industriels fonctionnent de manière tout à fait différente, grâce à un matériel dédié et optimisé pour des applications et des environnements d'exploitation précis. En général, ces systèmes tournent sous des systèmes d'exploitation personnalisés ou renforcés, qui privilégient la stabilité et la prévisibilité à des mises à jour fréquentes. Leur cycle de vie prolongé, allant de 10 à 15 ans, correspond à leur utilisation dans les déploiements industriels à long terme, et leur résistance environnementale garantit un fonctionnement fiable même en présence de

Chapitre I: Généralités sur les systèmes embarqués et la technologie des codes-barres

températures extrêmes, de vibrations et d'interférences électromagnétiques.[5] Cette philosophie de conception privilégie les capacités de traitement en temps réel et un comportement prévisible, ce qui les rend idéales pour les applications critiques où la fiabilité est primordiale et prime sur la flexibilité.

La principale différence entre un ordinateur et un système embarqué réside dans la gestion des ressources. Les ordinateurs disposent de ressources importantes (mémoire, puissance de calcul, stockage), tandis que les systèmes embarqués fonctionnent avec des ressources limitées, nécessitant une optimisation rigoureuse des logiciels et des ressources matérielles.

La consommation énergétique constitue également un critère essentiel : les ordinateurs consomment beaucoup d'énergie et sont alimentés en continu, alors que les systèmes embarqués, souvent alimentés par batterie, doivent minimiser leur consommation grâce à des mécanismes spécifiques (modes veille, gestion dynamique de la fréquence du processeur).

Concernant le système d'exploitation, les ordinateurs utilisent des systèmes complets et polyvalents, tandis que les systèmes embarqués fonctionnent soit sans système d'exploitation (**mode bare-metal**), soit avec un système temps réel (RTOS) garantissant des délais de réponse déterministes.[5]

Enfin, les systèmes embarqués se distinguent par leur robustesse, leur fiabilité, leur faible coût et leur forte intégration, car ils doivent fonctionner de manière autonome et durable, parfois dans des conditions difficiles, avec peu d'intervention humaine.

I.3 La plateforme ESP32

L'ESP32, développé par Espressif Systems et fabriqué par TSMC selon un procédé de gravure en 40 nm, est une famille de microcontrôleurs économiques et à faible consommation intégrant nativement les technologies Wi-Fi et Bluetooth, conçue comme une évolution de l'ESP8266. Cette puce se distingue par la diversité de ses architectures de traitement, pouvant embarquer un microprocesseur Tensilica Xtensa LX6 (en version monocœur ou bicœur), un Xtensa LX7 bicœur ou encore un cœur RISC-V monocœur, ce qui lui confère une grande flexibilité en fonction des besoins applicatifs.

Chapitre I: Généralités sur les systèmes embarqués et la technologie des codes-barres

L'ESP32 intègre également de nombreux composants radio et analogiques essentiels, tels que des commutateurs d'antenne, un balun RF, des amplificateurs de puissance, des récepteurs à faible bruit, des filtres et des modules avancés de gestion de l'alimentation, réduisant ainsi le nombre de composants externes ainsi que l'encombrement du circuit imprimé.

Conçu pour fonctionner dans des environnements industriels exigeants, il supporte une large plage de températures allant de $-40\text{ }^{\circ}\text{C}$ à $+125\text{ }^{\circ}\text{C}$ et dispose de mécanismes d'étalonnage dynamique permettant de compenser les variations et imperfections des circuits externes.

Optimisé pour les applications embarquées, mobiles et IoT, il propose une consommation énergétique très faible grâce à une gestion fine de l'horloge, à plusieurs modes d'alimentation et à une adaptation dynamique de la puissance.

Généralement intégré sur des cartes dédiées ou des kits de développement dotés de nombreuses interfaces GPIO, il peut fonctionner de manière autonome ou comme périphérique esclave d'un microcontrôleur hôte via des interfaces telles que SPI, SDIO, I2C ou UART, offrant ainsi une solution complète, compacte et performante pour les systèmes communicants sans fil.

[6]

I.3.1 Architecture ESP32

L'architecture ESP32, conçue par Espressif Systems, est parfaitement compatible avec la plateforme logicielle ESP-IDF (Internet-of-Things Development Framework) et peut également fonctionner sous Arduino. Cette flexibilité permet d'utiliser l'ESP32 dans divers systèmes embarqués et applications IoT.

Côté performances, l'ESP32 standard utilise un microprocesseur double cœur 32 bits Tensilica Xtensa LX6, offrant des capacités de traitement relativement bonnes, notamment pour les applications multithreads. Le dispositif intègre un système d'exploitation temps réel, FreeRTOS, qui assure la gestion des tâches en temps réel.

L'architecture mémoire de l'ESP32 est de type Harvard. Il dispose d'environ 520 Ko de RAM statique interne, extensible via une mémoire Flash externe jusqu'à 16 Mo et une RAM externe jusqu'à 4 Mo.

Chapitre I:Généralités sur les systèmes embarqués et la technologie des codes-barres

Côté connectivité, l'ESP32 intègre un module radio compatible Wi-Fi 802.11 b/g/n (2,4 GHz) et Bluetooth v4.2 (BR/EDR et BLE), ce qui le rend idéal pour le développement d'objets connectés sans fil bimodes.

De plus, il dispose de nombreux périphériques intégrés, réduisant ainsi le besoin en composants externes. Parmi eux, on trouve plus de 30 broches GPIO, des capteurs tactiles capacitifs, des convertisseurs analogique-numérique (CAN) 12 bits, des convertisseurs numérique-analogique (CNA) 8 bits, ainsi que des interfaces de communication telles que UART, SPI, I2C et I2S. Un appareil basé sur l'ESP32 peut ainsi intégrer de nombreuses fonctionnalités, offrant une plus grande liberté de conception.

L'une des particularités de l'ESP32 est l'intégration d'un coprocesseur à très faible consommation (ULP), qui peut être utilisé pour contrôler des capteurs et des périphériques spécifiques lorsque la puce principale est en mode veille profonde, réduisant ainsi la consommation d'énergie.

L'ESP32 intègre des moteurs cryptographiques matériels offrant une protection avancée, notamment AES, SHA-2, RSA, ECC et un générateur de nombres aléatoires (RNG).

Il existe différentes versions architecturales de la puce :

- L'ESP32 standard avec le processeur double cœur Xtensa LX6 ;
- L'ESP32-S (ESP32-S2, ESP32-S3) avec un processeur Xtensa LX7 plus rapide (240 MHz), conçu pour des performances accrues et les applications d'apprentissage automatique ;

Chapitre I: Généralités sur les systèmes embarqués et la technologie des codes-barres

- L'ESP32-C (C3, C6) utilisant le processeur RISC-V 32 bits pour les applications basse consommation et représentant une version moderne de l'ESP8266. [5]

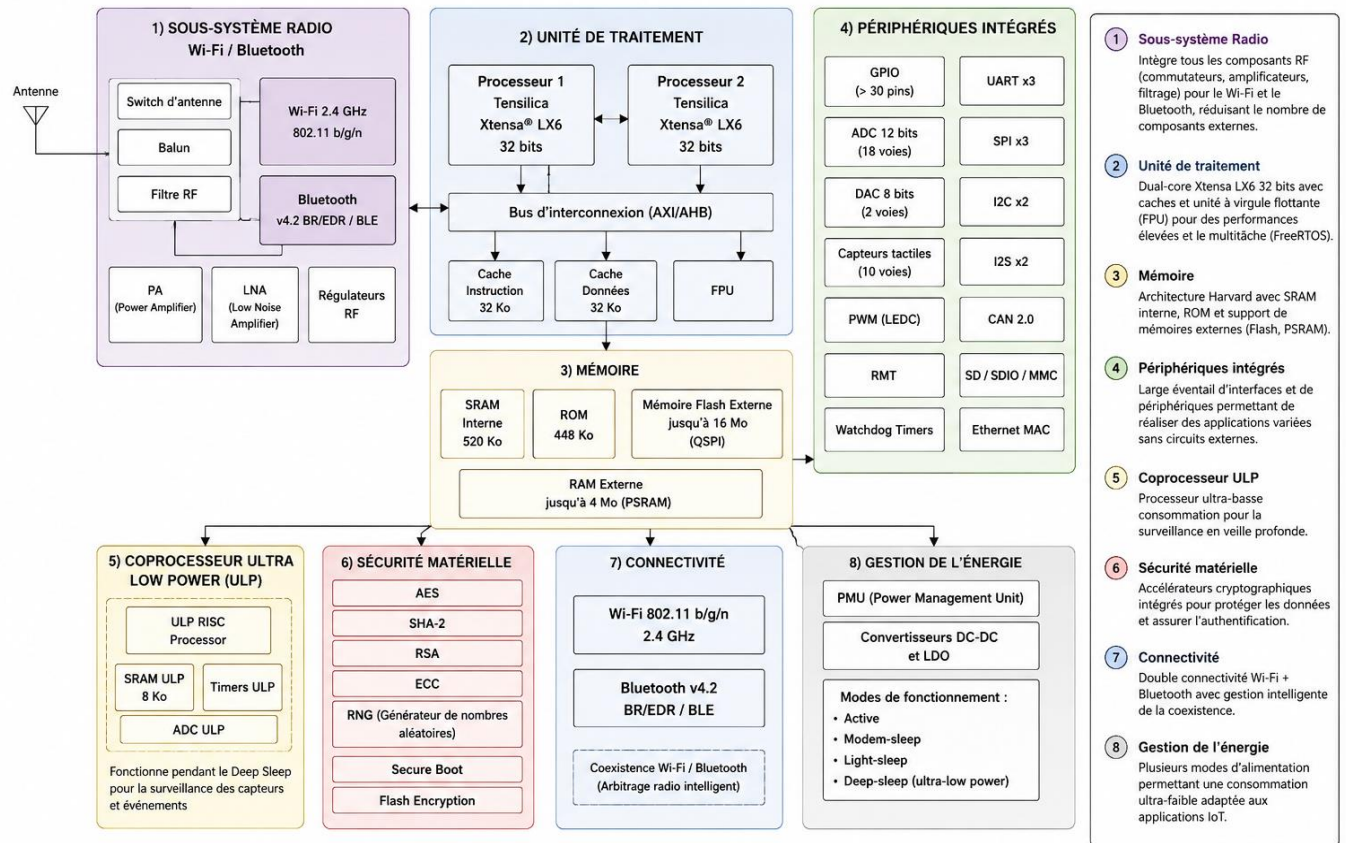


Figure I-2 : Architecture matérielle interne de l'ESP32

I.3.2 Caractéristiques techniques d'ESP32 :

Les ESP32 comprennent les caractéristiques techniques suivantes :

- CPU : Xtensa double-cœur (ou simple-cœur), microprocesseur LX 32 bits, fonctionnant à 160 ou 240 MHz et fournissant jusqu'à 600 DMIPS ;
- coprocesseur ultra basse consommation (ULP) ;
- Mémoire :
- 32 Mo SRAM ;
- Connectivité sans-fil :
- Wi-Fi : 802.11 b/g/n ;
- Bluetooth : v 4.2 BR/EDR et BLE jusqu'à v 5.0 et v 5.1 ;
- Interfaces de périphériques :

Chapitre I:Généralités sur les systèmes embarqués et la technologie des codes-barres

- Segmentation 12-bit sur les ADC (SAR ADC) jusqu'à 18 canaux ;
- 2×8 bit DAC ;
- $10 \times$ capteurs de toucher (GPIO de capteur capacitif (en)) ;
- $4 \times$ SPI ;
- $2 \times$ interfaces I²S ;
- $2 \times$ interfaces I²C ;
- $3 \times$ UART ;
- Contrôleur hôte SD/SDIO/CE-ATA (en)/MMC/eMMC ;
- Contrôleur esclave SDIO/SPI ;
- Interface MAC Ethernet avec DMA dédié et support du protocole de temps précis IEEE 1588 ;
- Bus de données CAN 2.0 ;
- Contrôleur infrarouge distant (TX/RX, jusqu'à 8 canaux) ;
- Moteur PWM ;
- LED PWM (jusqu'à 16 canaux) ;
- Capteur à effet Hall ;
- pré-amplificateur analogique ultra-basse consommation ;
- Sécurité :
 - Standard de sécurité supportant complètement IEEE 802.11, incluant WPA/WPA2 et WAPI de WFA ;
 - Secure boot (démarrage sécurisé) ;
 - Chiffrement de la Flash ;
 - 1024-bit OTP, jusqu'à 768 bit pour les clients ;
 - Accélération matérielle du chiffrement : AES, SHA-2, RSA, Cryptographie sur les courbes elliptiques (ECC), Générateur de nombres aléatoires (RNG) ;
- Gestion de l'énergie :
 - low-dropout regulator (en) interne.
- Domaines d'alimentation individuels pour le RTC
- Alimentation en sommeil profond de 5 μ A ;

Chapitre I:Généralités sur les systèmes embarqués et la technologie des codes-barres

- Réveil depuis des interruption GPIO, timer, mesure ADC, interruption du capteur de touché capacitif.

I.3.3 Module ESP32-S3-CAM

L'ESP32-S3 est une version améliorée de la famille de microcontrôleurs ESP32 conçue par Espressif Systems. C'est un système sur puce (SoC) conçu spécifiquement pour les applications embarquées intelligentes qui exigent à la fois des capacités de communication sans fil, de traitement de données et de vision artificielle. Le développement de cette carte donne une connectivité totale qui comprend une entrée/issue USB Type-C pour UART (Universel émetteur-récepteur asynchrone), une entrée/issue USB Type-C pour USB OTG (Universal Serial Bus On-The-Go), une led RGB, une caméra OV3660, un lecteur de carte SD et des entrées/issus GPIO. L'espacement de 25,4 mm entre les deux rangées d'entrées/issus garantit la compatibilité avec les protocoles de test, ce qui facilite le processus de prototypage et de développement. Cette carte est principalement utilisée dans divers domaines comme la domotique, l'automatisation industrielle, le domaine médical, l'électronique grand public et l'agriculture intelligente. [\[6\]](#)[\[8\]](#)

Chapitre I: Généralités sur les systèmes embarqués et la technologie des codes-barres

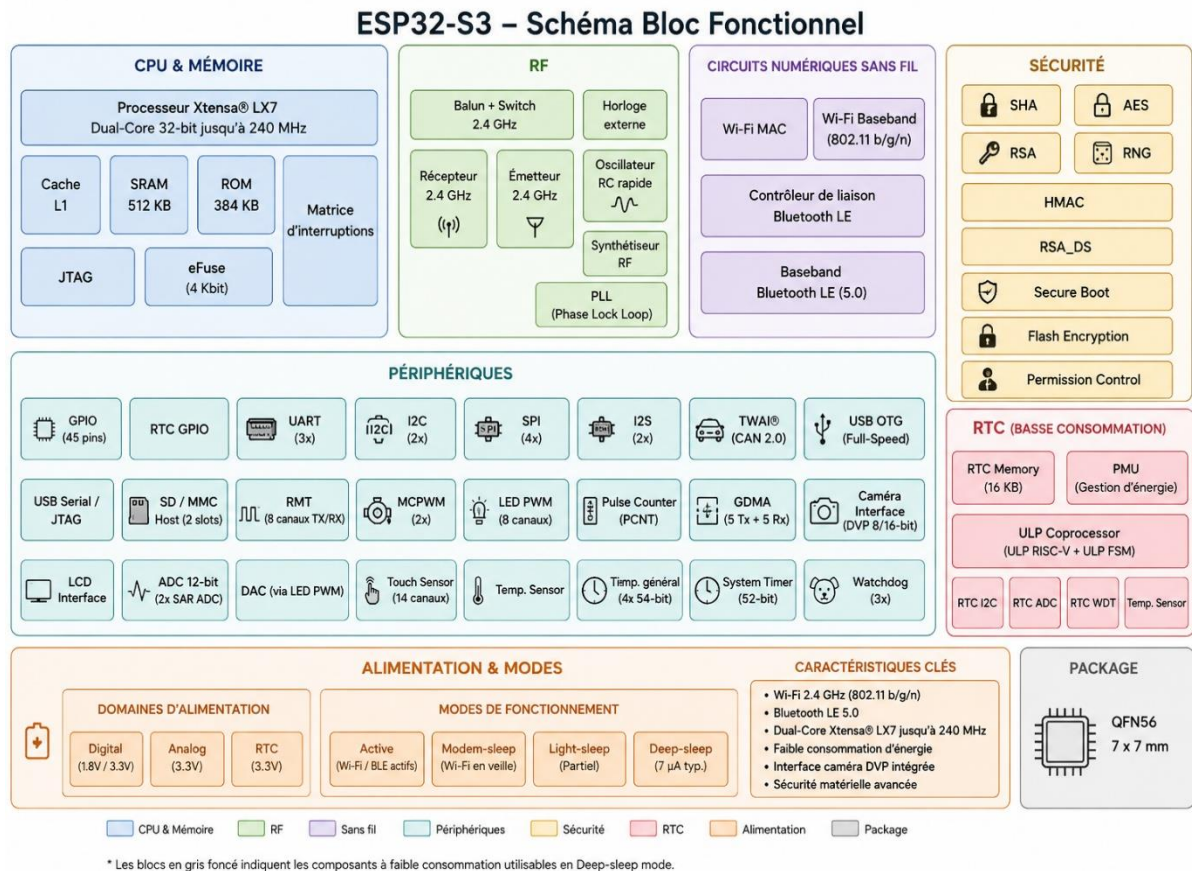


Figure I-3 : schema Block Esp32-s3

I.3.4 Caractéristiques :

- Wi-Fi
- Conforme aux normes IEEE 802.11b/g/n
- Prend en charge les bandes passantes de 20 MHz et 40 MHz dans la bande 2,4 GHz
- Mode 1T1R avec un débit de données jusqu'à 150 Mbit/s
- Wi-Fi Multimedia (WMM)
- A-MPDU et A-MSDU en émission/réception
- Accusé de réception immédiat par bloc (ACK)
- Fragmentation et défragmentation
- Surveillance automatique des balises (TSF matériel)

Chapitre I: Généralités sur les systèmes embarqués et la technologie des codes-barres

- Quatre interfaces Wi-Fi virtuelles
- Prise en charge simultanée des BSS d'infrastructure en modes Station, SoftAP ou Station + SoftAP. Remarque : lors d'une analyse en mode Station par l'ESP32-S3, le canal SoftAP change en même temps que le canal Station.
- Diversité d'antenne
- 802.11mc FTM
- Bluetooth®
- Bluetooth LE : Bluetooth 5, Bluetooth Mesh
- Mode haute puissance avec une puissance d'émission jusqu'à 20 dBm
- Vitesse : 125 Kbps, 500 Kbps, 1 Mbps, 2 Mbps
- Extensions publicitaires LE
- Plusieurs ensembles de publicités
- Algorithme de sélection de canal LE n° 2
- Mécanisme de coexistence interne entre Wi-Fi et Bluetooth pour partager la même antenne
- Processeur et mémoire
- Microprocesseur Xtensa® LX7 bicœur 32 bits
- Fréquence d'horloge : jusqu'à 240 MHz
- Score CoreMark® : – Deux cœurs à 240 MHz : 1329,92 CoreMark ; 5,54 CoreMark/MHz
- Pipeline à cinq étages
- Bus de données 128 bits et instructions SIMD dédiées
- Unité de calcul en virgule flottante simple précision (FPU)
- Coprocesseurs ultra basse consommation (ULP) : – Coprocesseur ULP-RISC-V – Coprocesseur ULP-FSM
- Contrôleur DMA général, avec 5 canaux d'émission et 5 canaux de réception
- Cache L1
- ROM : 384 Ko
- SRAM : 512 Ko
- SRAM dans l'horloge temps réel (RTC) : 16 Ko
- Mémoire eFuse 4 096 bits, jusqu'à 1 792 bits pour l'utilisateur

Chapitre I: Généralités sur les systèmes embarqués et la technologie des codes-barres

- Protocoles SPI pris en charge : interfaces SPI, Dual SPI, Quad SPI, Octal SPI, QPI et OPI permettant la connexion à la mémoire flash, à la RAM externe et à d'autres périphériques SPI
- Prise en charge du contrôleur flash avec cache
- Prise en charge de la programmation flash en circuit (ICP)
- Sécurité
- Démarrage sécurisé – contrôle des autorisations Accès à la mémoire interne et externe
- Chiffrement Flash - chiffrement et déchiffrement de la mémoire
- Accélération matérielle cryptographique :
 - Accélérateur SHA (FIPS PUB 180-4)
 - Accélérateur AES (FIPS PUB 197)
 - Accélérateur RSA
 - Accélérateur HMAC
 - Périphérique de signature numérique RSA (RSA_DS)
 - Générateur de nombres aléatoires (GNA) [\[6\]](#)[\[8\]](#)

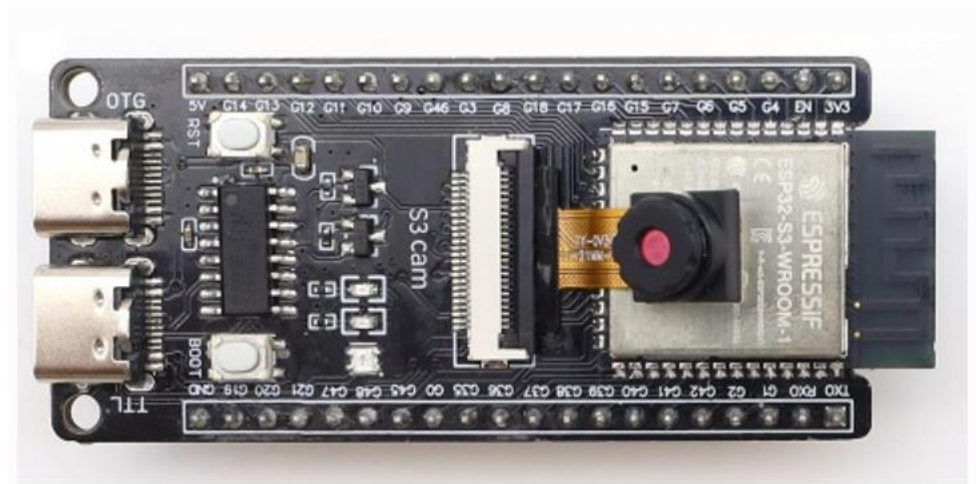


Figure I-4 : ESP32-S3-Cam

I.3.5 AVANTAGES :

- Intégration de la connectivité : Le circuit-imprimé comporte des technologies Wi-Fi et BLE intégrées qui facilitent une communication sans fil avec d'autres appareils ou serveurs.

Chapitre I:Généralités sur les systèmes embarqués et la technologie des codes-barres

- Bonne performance : En raison de sa fréquence maximale de 240 MHz et d'un processeur à cœur dual basé sur 32 bits, ESP32-S3 a une bonne puissance de traitement.
- Caméra à haute résolution :Le circuit-imprimé comprend un capteur de caméra OV3660 qui peut prendre des photos de haute résolution de 3 mégapixels pour :
- La reconnaissance faciale,
- La surveillance,
- La détection d'objet,
- La vision artificielle
- Faible consommation d'énergie : Le circuit-imprimé a été développé pour des utilisations dans les systèmes embarqués ou en tant que système autonome à partir de piles grâce à diverses fonctions d'économies d'énergie.
- Appui à l'intelligence artificielle : Le microcontrôleur a des instructions spécialisées pour le traitement :
- TinyML,
- La reconnaissance vocale,
- Le traitement d'image,
- L'intelligence artificielle embarquée.
- Support de différentes plateformes : Le circuit-imprimé est compatible avec plusieurs plateformes :
- Arduino IDE,
- ESP-IDF,
- MicroPython,
- PlatformIO.

Cela facilite le codage.

- Présence de la puce microSD : Il y a une interface avec la carte de stockage SD pour le stockage local :
- Des images,
- Des vidéos,
- Des données collectées par le système.

Chapitre I:Généralités sur les systèmes embarqués et la technologie des codes-barres

- Petit encombrement et peu coûteux : La caméra ESP32-S3 CAM OV3660 est un système polyvalent à petit encombrement et pas cher.
- Facilité d'intégration : En raison de sa multitude d'interfaces (GPIO, UART, SPI, I2C), elle s'intègre facilement avec :
 - Des capteurs,
 - Des écrans,
 - Des modules externes,
 - Des systèmes domotiques ou robotiques.
- Idéale pour les projets IoT : Cette carte est parfaite pour :
 - Les systèmes de surveillance intelligents,
 - La maison connectée,
 - Les robots intelligents,
 - Les applications de vision embarquée,
 - Les projets éducatifs et de recherche.

I.3.6 Comparaison avec Arduino :

Le choix de la plateforme embarquée est une étape essentielle dans notre projet de gestion intelligente des examens, car il influence directement les performances du système, la rapidité d'acquisition des images et la communication avec le serveur de traitement. Pour justifier le choix matériel effectué, une comparaison a été faite entre l'ESP32-S3 CAM OV3660 et la carte Arduino Uno.

L'ESP32-S3 est un système sur puce (SoC) développé par Espressif qui inclut un processeur Xtensa LX7 double cœur 32 bits, cadencé jusqu'à 240 MHz, et qui offre une connectivité Wi-Fi 2,4 GHz et Bluetooth Low Energy intégrées [6]. Il dispose également d'une grande mémoire SRAM et d'un support matériel dédié aux applications de vision embarquée grâce à son contrôleur LCD_CAM.

À l'inverse, la carte Arduino Uno est basée sur un microcontrôleur ATmega328P 8 bits cadencé à 16 MHz, avec seulement 2 KB de SRAM et sans connectivité Wi-Fi native [9]. Elle convient parfaitement aux projets électroniques simples et aux systèmes de commande, mais

Chapitre I: Généralités sur les systèmes embarqués et la technologie des codes-barres

ses ressources sont insuffisantes pour les applications qui exigent l'acquisition et la transmission d'images.

Dans notre projet, la plateforme embarquée doit pouvoir gérer plusieurs tâches en même temps : capturer les copies d'examen grâce au capteur OV3660, compresser les images, et enfin les envoyer vers le serveur Flask par Wi-Fi. Ces contraintes exigent une puissance de calcul accrue ainsi qu'une connectivité réseau native, ce que l'ESP32-S3 offre[6].

En outre, l'ESP32-S3 possède une interface matérielle de caméra compatible avec les capteurs de type DVP, tel que le module OV3660, ce qui permet une communication directe entre la caméra et le microcontrôleur [10]. Cette fonctionnalité n'existe pas sur Arduino Uno, qui aurait besoin de composants externes supplémentaires pour contrôler une caméra, augmentant ainsi la complexité matérielle du montage.

Le capteur OV3660 du module ESP32-S3 CAM utilisé dans notre projet offre une résolution de 3 mégapixels et supporte divers formats d'image tels que YUV et RGB, le rendant idéal pour les applications embarquées de vision [10]. Le serveur utilise ensuite les images saisies afin de décoder les codes-barres et d'analyser automatiquement les annotations manuscrites.

Ainsi, pour les besoins du système développé (acquisition d'image, communication réseau, traitement intelligent déporté et intégration compacte), l'ESP32-S3 semble être une solution plus performante et mieux adaptée qu'Arduino Uno. Sa conception permet de répondre efficacement aux besoins du projet tout en maintenant un coût réduit et une faible consommation énergétique [7].

Critère	ESP32-S3 CAM OV3660	Arduino Uno
Architecture du processeur	Xtensa LX7 Dual-Core 32 bits	ATmega328P 8 bits
Fréquence de fonctionnement	Jusqu'à 240 MHz	16 MHz
Mémoire SRAM	512 KB	2 KB
Connectivité Wi-Fi	Intégrée (2.4 GHz)	Non intégrée (module externe requis)
Connectivité Bluetooth	Bluetooth Low Energy (BLE) intégré	Non disponible
Interface caméra	Oui (DVP – LCD_CAM intégré)	Non disponible

Chapitre I:Généralités sur les systèmes embarqués et la technologie des codes-barres

Capture et traitement d'image	Oui – Supporte les capteurs OV3660, OV2640...	Très limité, non adapté à la vision embarquée
Communication réseau	Wi-Fi, HTTP, TCP/IP natifs	Module Wi-Fi externe nécessaire (ex : ESP8266)
Puissance de calcul	Élevée – Dual core jusqu'à 240 MHz	Faible – Processeur 8 bits
Capacité de traitement	Adaptée au traitement d'images et à l'IA embarquée	Non adaptée aux traitements complexes
Consommation énergétique	Optimisée (modes économie d'énergie)	Faible mais sans optimisation avancée
Coût	Faible à moyen	Très faible
Adaptation à notre projet	Parfaitement adapté (vision, Wi-Fi, envoi d'images, etc.)	Non adapté aux besoins du système

Tableau 1 : ESP32-S3 Cam OV3660 vs Arduino

I.4 Technologie des codes-barres :

L'Histoire du Code-Barres trouve sa source dans le lieu le plus improbable : la plage de Floride en 1948. En effet, Norman Joseph Woodland, élève ingénieur en mécanique de l'université Drexel de Philadelphie, dessine inconsciemment des lignes parallèles dans le sable en traçant celles-ci avec ses doigts. Initialement familier avec le code Morse grâce à ses expériences de scout, il comprend tout à coup qu'il était possible d'utiliser ce même principe : les points et les traits pouvant véhiculer des informations. Il utilise ainsi avec son co-fondateur Bernard Silver pour qui il développe un concept, les barres verticales d'épaisseurs différentes et lisibles par machine. Ces derniers soumettent leur brevet le 20 octobre 1949 et obtiennent celui-ci le 7 octobre 1952. Le système breveté se composait alors de cercles concentriques en forme d'oeil de boeuf lisibles par un faisceau lumineux.[\[11\]](#)

C'est une idée visionnaire, mais cependant, elle est avant son temps. Woodland rejoindra IBM en 1951 avec l'intention de mettre au point sa création, mais encore vingt ans seront nécessaires afin que les technologies de lecture par laser puissent être mises en œuvre pour l'analyser. Le brevet de son inventeur sera, quant à lui, vendu pour une somme dérisoire de 15 000 dollars. IBM commencera alors ses efforts dans cette direction à la fin des années 1960 en créant un groupe d'ingénieurs travaillant au perfectionnement de la barre code, dont Woodland lui-même faisait partie. Ce fut de cette initiative que découla le UPC, le Code Universel des Produits.[\[12\]](#)

Le tout premier scan de code-barres a eu lieu le 26 juin 1974 dans un supermarché Marsh's à Troy, dans l'Ohio, lorsqu'un code-barres figurant sur un paquet de chewing-gum Wrigley's Juicy Fruit a été scanné. Ce simple événement a marqué le début d'un processus irréversible qui a

Chapitre I: Généralités sur les systèmes embarqués et la technologie des codes-barres

profondément transformé notre économie mondiale. Le paquet de ce chewing-gum est aujourd'hui conservé au Musée national d'histoire américaine du Smithsonian à Washington, en témoignage de cette révolution discrète. Aujourd'hui, des milliards de codes-barres de ce type sont scannés chaque jour dans le monde, à un rythme d'environ 116 000 par seconde. Xpert + 2 [13]

I.4.1 Définition et principe de fonctionnement :

Ces codes-barres omniprésents dans nos vies quotidiennes sont l'exemple même des innovations modestes mais puissantes qui ont profondément modifié le fonctionnement du monde tel que nous le connaissons. Ces codes-barres se rencontrent dans nos placards, nos salles de bains, nos bibliothèques, nos hôpitaux, nos aéroports, nos entrepôts et tous nos magasins. Néanmoins, malgré une telle omniprésence, ces codes-barres ne passent pas toujours inaperçus. Or c'est justement parce qu'ils sont si invisibles qu'ils sont vraiment efficaces : un outil vraiment efficace est celui qu'on oublie peu à peu grâce à son utilité. Mais qu'est-ce qu'un code-barres, précisément ?

Par définition, un code-barres est une représentation visuelle de données qui se présente habituellement sous la forme d'un carré ou d'un rectangle, constitué d'une série de barres verticales noires de largeur et de hauteur différentes, de blancs et, si nécessaire, de chiffres. Ces éléments réunis ensemble créent un code unique qui aide dans l'identification des données concernant un certain article, individu ou objet. Les ordinateurs avec les scanners liés à eux sont capables de lire ces codes et utiliser les barres, les espaces et les chiffres pour obtenir les données correspondantes dans une base de données. [14]

Le fonctionnement du code-barres est basé sur une règle de base qu'on appelle la symbologie. Symbologie, dans le sens le plus littéral, fait référence à la structuration du code-barres : c'est-à-dire la manière par laquelle sont interprétées et interprétées les données codées. En pratique, elle aide le lecteur à comprendre où commence le chiffre ou le caractère et où il prend fin comme une forme de notation binaire. La largeur des barres noires est généralement interprétée comme une valeur de 0 ou 1, alors que la combinaison de celles-ci signifie un chiffre allant de 0 à 9. Une machine ordinateur reliée au scanner pourra alors additionner, multiplier et diviser ces nombres pour trouver exactement le produit codé qui s'affiche automatiquement à l'écran. [15]

Chapitre I:Généralités sur les systèmes embarqués et la technologie des codes-barres

Au niveau technique, les traits de la barre de codes diffèrent en termes d'encodage des largeurs. Certaines méthodes utilisent deux largeurs de traînées pour distinguer le caractère, d'autres font usage de différentes largeurs des traînées. Le choix d'un ou l'autre style encodant dépend entièrement de la nature et de la nécessité de l'application. [\[15\]](#)

Pour extraire les informations contenues dans un code-barres, ce dernier doit d'abord être scanné au moyen d'une machine spécifique, qui peut soit se présenter sous forme de machine portable portée à la main, soit être fixe et installé au sein d'un guichet de paiement ou d'une station de travail. La machine de scanne effectue une mesure lumineuse des reflets provenant des lignes du code-barres, distinguant alors la zone noire des zones blanches. Parmi les types de lasers les plus utilisés pour l'opération figurent ceux d'hélium-néon qui présentent un très faible besoin d'énergie tout en étant très efficaces. Il est important de calibrer de manière rigoureuse le laser et le système pour une lecture précise du code-barres, en fonction de la symbologie, du format et de la structure des largeurs de lignes du code.[\[14\]](#)

Parmi ses nombreux avantages, la machine de scanne est notamment très utile pour éviter la saisie manuelle des données. Où un employé serait normalement chargé de renseigner en temps réel tous les renseignements sur un article, une opération qui requiert beaucoup de temps et est sujette à des erreurs, la machine permet d'extraire les données en quelques secondes et ce de façon très précise.

Chapitre I: Généralités sur les systèmes embarqués et la technologie des codes-barres

Depuis leur mise en place il y a plus de soixante-dix ans, les codes-barres ont donc été pour les entreprises un atout majeur, les aidant à faire fonctionner leurs activités d'une façon plus efficace, ainsi qu'à suivre l'acheminement de leurs produits à chaque étape du cycle de production, de vente et d'après-vente : de la conception, de la fabrication, de l'achat et de la vente à la réparation.[\[16\]](#)

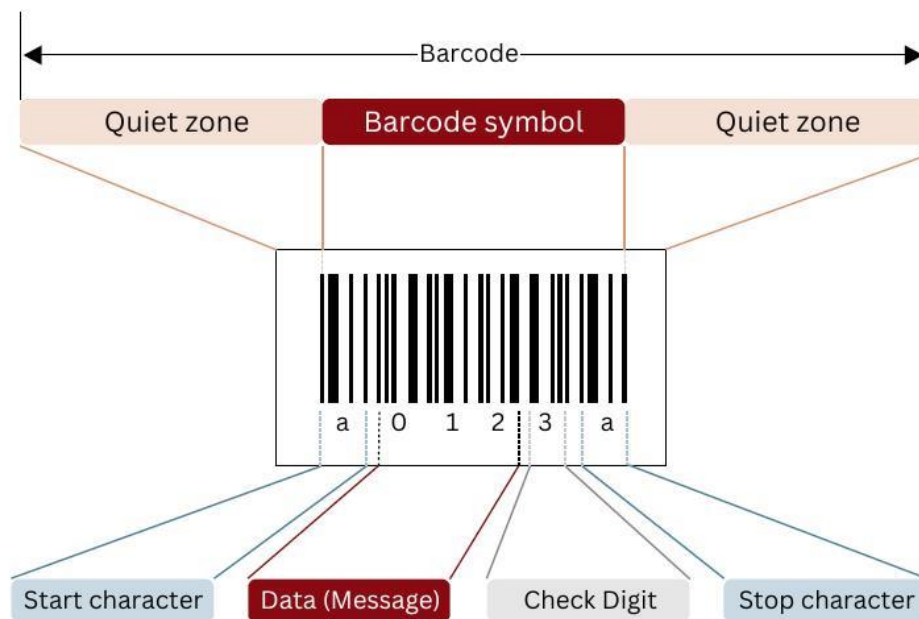


Figure I-5 : Structure d'un code-barre 1D

Chapitre I:Généralités sur les systèmes embarqués et la technologie des codes-barres

I.4.2 Types de codes-barres et symbologies :

Techniquement, le système des codes-barres est loin d'être uniforme. En effet, la technologie des codes-barres constitue une gamme variée de formats qui ont leurs propres caractéristiques, et chaque format a été créé dans le but d'assurer l'accomplissement de tâches précises. Il existe deux catégories principales de codes-barres : les 1D et les 2D.[\[17\]](#)

- **Les codes-barres 1D (linéaires) :**

Les codes-barres linéaires ou unidimensionnels, constituent les plus anciennes et les plus largement utilisées. Il s'agit précisément de ceux que nous sommes capables de reconnaître naturellement : une série de lignes verticales parallèles noires/ blanches qui s'étirent sur une base rectangulaire. La lecture de ces codes se fait grâce à un balayage unique horizontal puisque l'information est insérée de façon homogène sur toute la longueur de celui-ci. Codes-barres linéaires continuent d'être la base de tous les systèmes de caisses et de gestion de stock dans le monde entier [\[18\]](#).

En ce qui concerne leur structure interne, les codes-barres linéaires peuvent être divisés en deux sous-catégories [\[16\]](#) :

- Symbologies continues : lors desquelles les symboles s'enchaînent successivement les uns aux autres avec comme particularité d'avoir un premier symbole noir suivi par un dernier blanc.
- Symbologies discrète : lors desquelles chaque symbole est inscrit de façon autonome avec un espace intermédiaire bien délimité entre chaque.

Parmi les formats linéaires les plus utilisés à l'échelle mondiale, plusieurs standards se distinguent :

- UPC (Universal Product Code) : Largement utilisé dans le commerce de détail aux États-Unis et au Canada, contenant généralement 12 chiffres. [\[19\]](#)
- EAN (European Article Number) : Version internationale de l'UPC, comportant 13 chiffres (EAN-13) ou 8 chiffres (EAN-8).[\[20\]](#)
- Code 39 : Capable d'encoder des caractères alphanumériques, couramment utilisé dans les secteurs non-commerciaux.[\[18\]](#)

Chapitre I:Généralités sur les systèmes embarqués et la technologie des codes-barres

- Code 128 : Très dense, permettant d'encoder tous les caractères ASCII sur un espace réduit.[\[18\]](#)

Chacune de ces symbologies présente des caractéristiques et des avantages spécifiques qui la rendent adaptée à certains contextes d'utilisation.

- Les codes-barres 2D

En raison de certaines limites des codes linéaires, en particulier leurs limites de capacité de stockage de seulement une vingtaine de caractères, ont vu le jour les codes-barres 2D [\[18\]](#). Tandis que les codes-barres 1D ne stockent les informations qu'à une dimension horizontale, ceux 2D utilisent deux dimensions, à savoir la largeur et la hauteur du symbole. En effet, alors que le code-barres 1D encode 20 à 85 caractères, un symbole 2D est en mesure d'en contenir 100 à plus de 4 000 en fonction du type de code utilisé [\[18\]](#). De fait, la lecture des symboles 2D ne peut pas se faire par balayage avec un simple laser, mais nécessite un scanner à base d'image [\[17\]](#).

- DataMatrix : Capable de stocker jusqu'à 2 335 caractères alphanumériques dans un petit espace, souvent utilisé pour les petits composants électroniques.[\[18\]](#)
- PDF417 : Un code-barres empilé pouvant contenir jusqu'à 1,1 ko de données, utilisé notamment sur les permis de conduire.[\[21\]](#).
- Aztec Code : Ne nécessitant pas de zone de silence, idéal pour les espaces restreints comme les billets électroniques.

Ces formats avancés offrent une capacité de stockage supérieure tout en conservant certains avantages des codes-barres traditionnels.

Chapitre I: Généralités sur les systèmes embarqués et la technologie des codes-barres

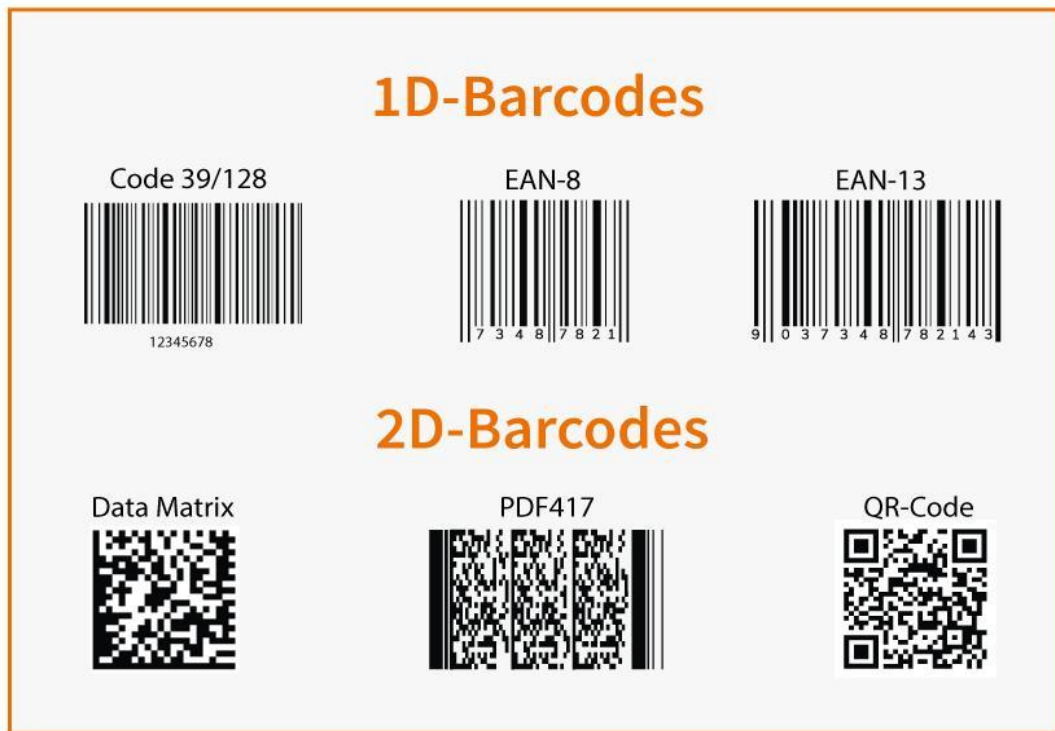


Figure I-6 : Type de codes-barres et symbologies

I.4.3 Différence entre code-barres et QR Code

Bien qu'il soit vrai que les codes-barres et le code QR relèvent tous les deux du large groupe de codes à lecture optique, il ne faut pas perdre de vue le fait que ces codes sont fondamentalement différents dans ce qui concerne leur structuration, leur potentiel de stockage, leur façon de lire les informations, leur résistance et leur utilisation.

- Structure et capacité de stockage

Le code-barres (1D) stocke l'information uniquement sur un axe horizontal sous forme de barres verticales de tailles différentes [18]. L'utilisation d'un système linéaire permet à ce type de code de pouvoir contenir de 20 à 85 caractères selon le format employé. Ce format simple est à la fois un avantage et une limitation pour ce code.

En effet, le code QR stocke l'information sur deux axes horizontaux et verticaux dans une matrice carrée comportant des modules noirs et blancs [18]. Cette double dimensionalité du code QR permet d'enrichir sa capacité de stockage qui est de 7 089 caractères pour les informations numériques ou de 4 296 caractères alphanumériques [18], soit environ cent fois plus que dans un

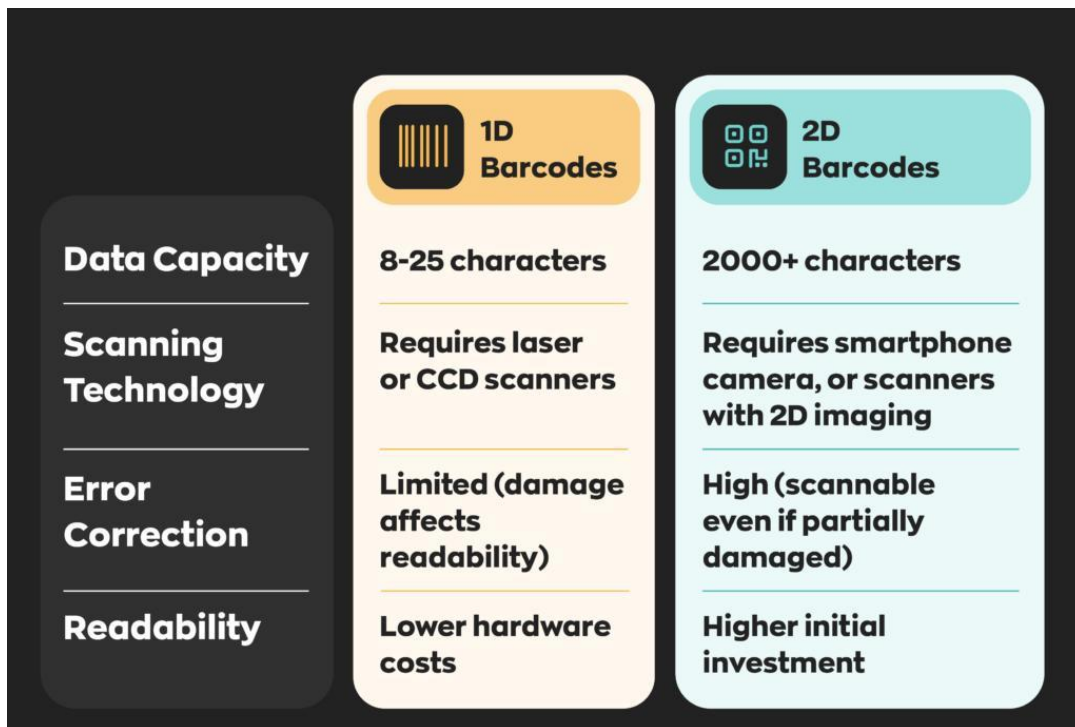
Chapitre I:Généralités sur les systèmes embarqués et la technologie des codes-barres

code-barres simple. On peut donc stocker de l'information plus importante comme des liens internet ou des informations multimédia [22]

- Mode de lecture et robustesse

Le code barre en une dimension est lu par un laser simple et rapide et peu cher qui requiert pour la lecture un bon alignement du code [16]. De son côté, le QR code nécessite un scanneur image, c'est-à-dire un téléphone mobile, qui va être capable de lire le code QR quelle que soit sa position, ce qui est rendu possible grâce à la présence des trois marqueurs de position placés aux quatre coins du code [21]. En outre, le QR code dispose d'une fonction de correction des erreurs grâce aux codes de Reed-Solomon, il est donc possible de lire le QR code même s'il y a un dommage de jusqu'à 30 % [23][24]

En conclusion, les deux systèmes ne se trouvent donc pas en concurrence mais plutôt en complémentarité. L'identification par le code barre demeure un standard incontournable dans les milieux où la vitesse et le cout minimum doivent être pris en compte, alors que le code QR est la solution adaptée à l'ère de l'information [21]



	 1D Barcodes	 2D Barcodes
Data Capacity	8-25 characters	2000+ characters
Scanning Technology	Requires laser or CCD scanners	Requires smartphone camera, or scanners with 2D imaging
Error Correction	Limited (damage affects readability)	High (scannable even if partially damaged)
Readability	Lower hardware costs	Higher initial investment

Figure I-7 : La différence entre les codes-barres 1D et 2D

Chapitre I:Généralités sur les systèmes embarqués et la technologie des codes-barres

I.4.4 Applications pratiques des codes-barres dans la gestion d'actifs

- Traçabilité et inventaire

Dans le domaine de la gestion des stocks et de l'inventaire, les codes-barres offrent une solution éprouvée pour le suivi des actifs :

- Identification unique : Chaque équipement peut recevoir un identifiant distinct.
- Rapidité d'inventaire : Les opérations d'inventaire physique sont considérablement accélérées, réduisant les erreurs manuelles.
- Contrôle des mouvements : Les entrées et sorties de matériel peuvent être enregistrées instantanément.
- Réduction des erreurs : La lecture automatisée élimine les erreurs de saisie manuelle.

Pour les entreprises gérant un parc informatique, l'étiquetage par codes-barres permet une identification rapide et précise de chaque composant, facilitant ainsi les audits et le suivi des déplacements d'équipements.[\[25\]](#)

- Optimisation de la maintenance

Les codes-barres contribuent efficacement à l'amélioration des processus de maintenance :

- Accès aux historiques : Un simple scan permet d'accéder à l'historique complet de maintenance d'un équipement.
- Planification préventive : Les interventions peuvent être planifiées selon des intervalles prédéfinis.
- Suivi des pièces détachées : La gestion des stocks de pièces de rechange est optimisée.
- Documentation des interventions : Chaque opération de maintenance peut être enregistrée et associée à l'équipement concerné.
- Gestion des consommables et réapprovisionnement

Les codes-barres excellents dans la gestion quotidienne des consommables :

- Suivi des stocks en temps réel : Chaque prélèvement ou ajout est immédiatement comptabilisé.
- Automatisation des commandes : Des alertes peuvent être générées lorsque les seuils minimaux sont atteints.

Chapitre I:Généralités sur les systèmes embarqués et la technologie des codes-barres

- Analyse des consommations : Les données collectées permettent d'optimiser les niveaux de stock.
- Réduction des ruptures de stock : La visibilité accrue permet d'anticiper les besoins.

Pour les secteurs comme le médical et paramédical, où la disponibilité des consommables est critique, cette fonctionnalité représente un avantage considérable en termes d'efficacité et de sécurité.

I.4.5 Applications sectorielles des codes-barres

- Secteur médical et hospitalier

Dans le domaine de la santé, les codes-barres sont devenus indispensables pour :

- La traçabilité des dispositifs médicaux : Suivi complet du cycle de vie et des stérilisations.
- La gestion des médicaments : Contrôle des dates de péremption et des conditions de stockage.
- L'identification des patients : Bracelets avec codes-barres pour réduire les erreurs médicales.
- La gestion des échantillons de laboratoire : Traçabilité précise des prélèvements.

Ces applications contribuent significativement à l'amélioration de la sécurité des patients et à l'optimisation des processus dans les établissements de santé.

- Construction et BTP

Sur les chantiers et dans les entreprises de construction, les codes-barres transforment la gestion du matériel :

- Suivi des outils et équipements : Réduction des pertes et vols.
- Gestion des stocks de matériaux : Anticipation des besoins et réduction des ruptures.
- Attribution aux équipes et projets : Responsabilisation et meilleure allocation des ressources.
- Maintenance préventive : Suivi des heures d'utilisation des machines.

Cette approche permet d'optimiser considérablement la gestion du matériel de construction, souvent coûteux et critique pour le respect des délais des projets.

Chapitre I:Généralités sur les systèmes embarqués et la technologie des codes-barres

- Administration publique et collectivités

Les codes-barres offrent aux administrations publiques des solutions efficaces pour :

- L'inventaire des actifs mobiliers : Bureaux, chaises, armoires dans les différents services.
- La gestion du parc informatique : Ordinateurs, imprimantes, téléphones.
- Le suivi du matériel technique : Équipements d'entretien, outils, véhicules.
- La gestion des archives et documents : Traçabilité des dossiers physiques.

Ces applications permettent aux administrations publiques d'améliorer leur efficacité tout en répondant aux exigences de transparence et de bonne gestion des ressources publiques.

I.5 Acquisition d'image dans les systèmes embarqués :

L'acquisition d'image est la première étape de toute chaîne de vision embarquée. C'est le processus de capture d'une scène réelle qui sera transformée en données numériques exploitables par le système. Sans cette première étape, on ne peut effectuer aucune analyse visuelle, détection d'objet, lecture automatique ou prise de décision. La qualité de l'image capturée a un impact direct sur les performances des traitements qui lui sont appliqués par la suite, notamment en matière de reconnaissance, de localisation ou de décodage.

Dans un système embarqué, l'acquisition d'image est soumise à des contraintes spécifiques dues aux ressources limitées de l'appareil. Un système embarqué, à la différence d'un ordinateur traditionnel, fonctionne le plus souvent avec une capacité de calcul réduite, une mémoire limitée, une consommation énergétique maîtrisée et un besoin de traitement en temps réel. Le choix des éléments de la chaîne d'acquisition doit donc faire des compromis entre performance, rapidité, coût et consommation énergétique.

Pour notre système, l'étape d'acquisition d'image est indispensable, car elle détermine la fiabilité du traitement numérique et la qualité des résultats obtenus. Cette acquisition est directement affectée par plusieurs facteurs, tels que le capteur d'image, les conditions

Chapitre I: Généralités sur les systèmes embarqués et la technologie des codes-barres

d'éclairage, la résolution de l'image et, depuis peu, l'incorporation de méthodes d'intelligence artificielle.[26]

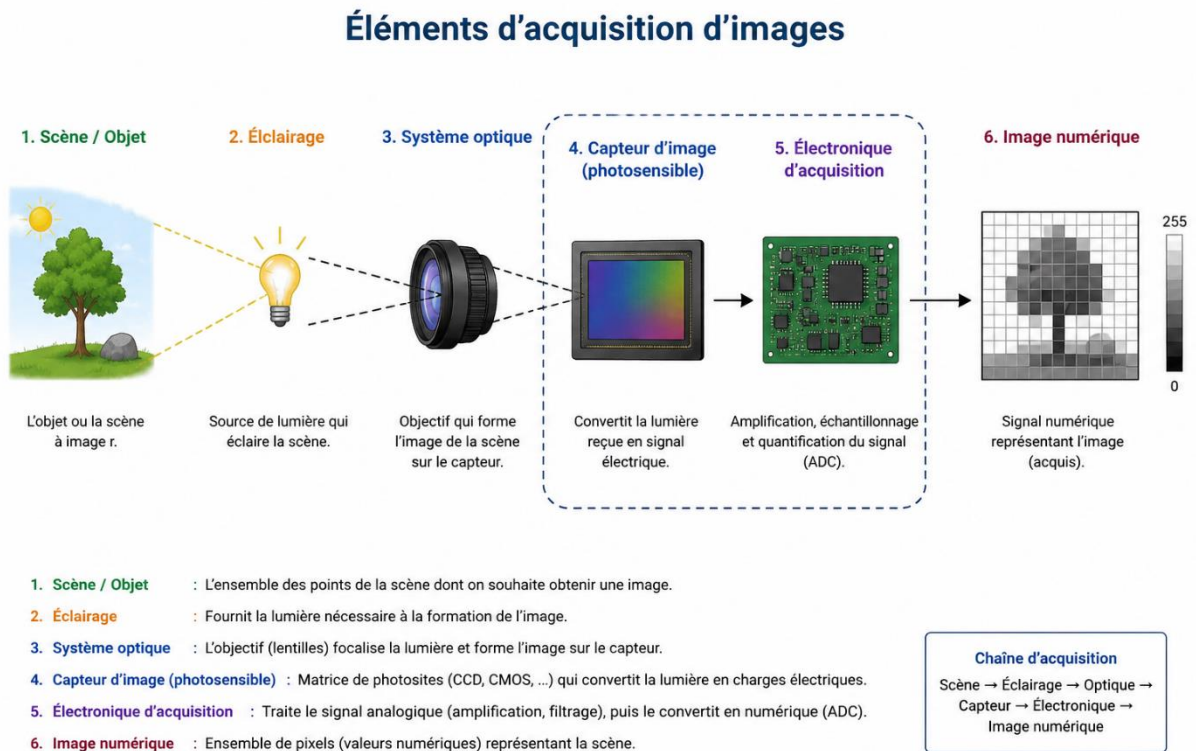


Figure I-8 : Chaîne d'acquisition d'image

I.5.1 Capteurs d'image

L'élément d'entrée de la chaîne d'acquisition d'image est le capteur, dont le rôle est de convertir en signaux électriques et numériques les photons entrant sur celui-ci. On distingue deux technologies de capteurs d'images qui permettent cette conversion : les CCD (Charge-Coupled Device) et CMOS. Les CCD ont été inventés en 1970 par Willard Boyle et George Smith au sein des laboratoires Bell. Ils fonctionnent sur le même principe de transfert de charge que la photodiode. Lorsque le photon arrive sur la diode, il génère des charges qui migrent en cascade dans un registre de décalage jusqu'à se retrouver à la sortie du capteur à un seul endroit, appelé « sortant ». Ce processus permet d'atteindre une grande homogénéité des pixels, peu de bruit de lecture et une grande dynamique pour ces capteurs.[27]

Il existe plusieurs types de structures de capteurs CCD. Par exemple, pour les CCD full frame, la lumière est détectée à l'aide de toute la surface de la structure. Cette méthode donne une très

Chapitre I:Généralités sur les systèmes embarqués et la technologie des codes-barres

grande sensibilité et un coefficient de remplissage très élevé, cependant, l'étape de lecture se fait lentement ligne par ligne avec l'aide de registres de décalage. Pour éviter la distorsion de l'image due au flou de mouvement, il faut créer une certaine condition de pose. CCD frame transfer sont constitués par la séparation de la structure de deux parties identiques – la zone photo-sensitive et la zone mémoire non exposée à la lumière. Les charges électriques sont transférées rapidement dans une partie masquée, ce qui permet de réaliser l'étape de lecture simultanément avec une nouvelle acquisition de lumière.

En ce qui concerne les CCD interline transfer, ces dernières possèdent un registre de transfert optiquement masqué pour chaque photo-diode, facilitant le déplacement de charges dans des microsecondes. Cela rend les images CCD interline transfer capables d'obtenir des expositions rapides sans avoir recours à un obturateur mécanique. Par contre, ces structures masquées entravent l'augmentation du facteur de remplissage puisque l'espace utilisé pour ces structures diminue l'aire photosensible. La solution pour remédier à ce désavantage pourrait être l'installation de microlentilles sur ces caméras.[\[27\]](#)

En termes généraux, les capteurs CCD sont reconnus pour avoir un taux de remplissage extrêmement élevé qui approche 100 %, pour la haute qualité de leurs images et l'étendue dynamique qu'ils peuvent fournir grâce au ratio entre la capacité de charge des photosites et les bruits d'obscurité et de lecture. Cependant, étant donné que leur fonctionnement est basé sur le transfert de charge, ils sont très gourmands en énergie, coûteux et difficiles à intégrer aux systèmes actuels pour divers usages comparativement aux capteurs CMOS.[\[23\]](#)

Chapitre I: Généralités sur les systèmes embarqués et la technologie des codes-barres

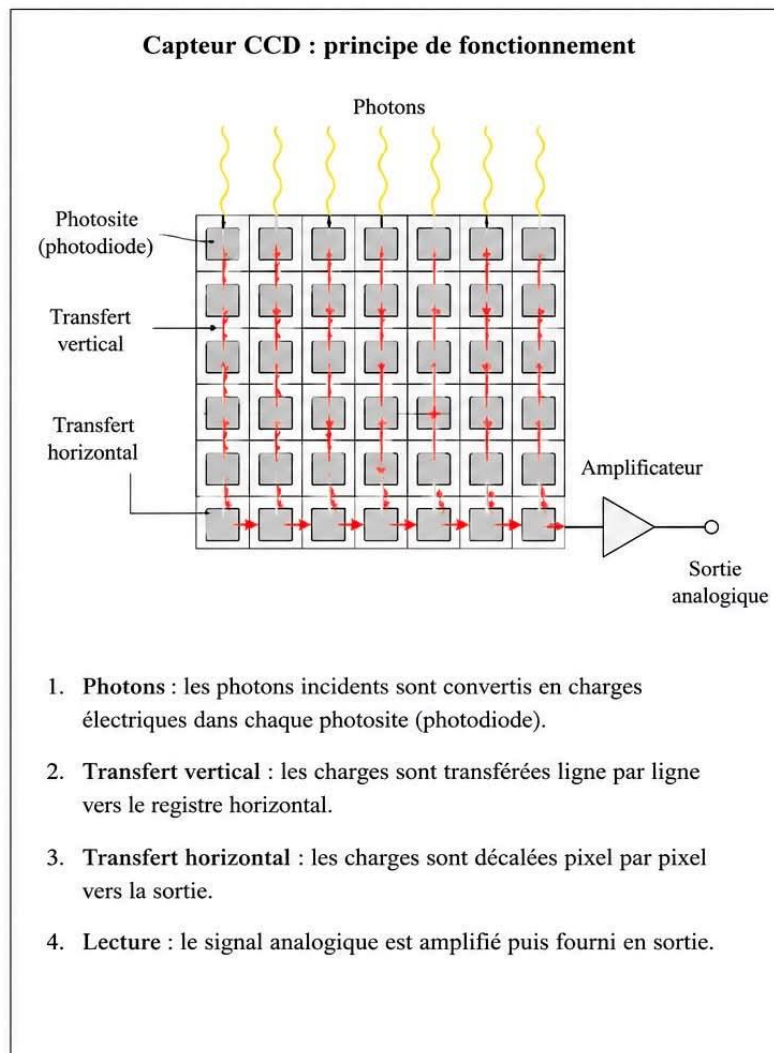


Figure I-9 : Principe de fonctionnement d'un capteur CCD

- Le capteur CMOS (Complementary Metal Oxyde Semiconductor)

Le capteur CMOS, et plus précisément sa variante à **pixels actifs (APS)**, a constitué une avancée majeure dans le domaine de l'imagerie embarquée. Développés dès 1993 au Jet Propulsion Laboratory (JPL) par **Eric R. Fossum** et son équipe, les capteurs CMOS APS intègrent dans chaque pixel son propre circuit de lecture, permettant un accès direct et parallèle à l'ensemble des pixels de la matrice. Fossum a démontré que cette architecture permettait d'atteindre des performances comparables aux CCD, tout en offrant des avantages décisifs pour les systèmes embarqués [27]

Chapitre I: Généralités sur les systèmes embarqués et la technologie des codes-barres

- Faible consommation énergétique : le CMOS peut fonctionner sous des tensions aussi basses que 1,2 V, réduisant considérablement la dissipation thermique [28].
<https://ieeexplore.ieee.org/document/551910/>
- Intégration sur puce : la compatibilité avec les procédés de fabrication CMOS standard permet d'intégrer sur un même circuit le capteur, l'ADC, le traitement du signal et les interfaces de communication [28].
- Vitesse de lecture élevée : les capteurs CMOS modernes peuvent atteindre des vitesses de lecture de plus de 1 000 mégapixels par seconde, contre moins de 70 pour les CCD [28].
- Coût réduit : la fabrication CMOS est mutualisée avec celle des circuits logiques standard, ce qui permet des économies d'échelle considérables.

Aujourd'hui, le capteur CMOS est devenu la technologie dominante dans les systèmes embarqués : smartphones, drones, systèmes de vision industrielle, caméras embarquées dans les véhicules autonomes. Des entreprises comme Sony et Samsung ont massivement investi dans cette technologie dès les années 2000, ce qui a précipité son adoption universelle [28].

Chapitre I: Généralités sur les systèmes embarqués et la technologie des codes-barres

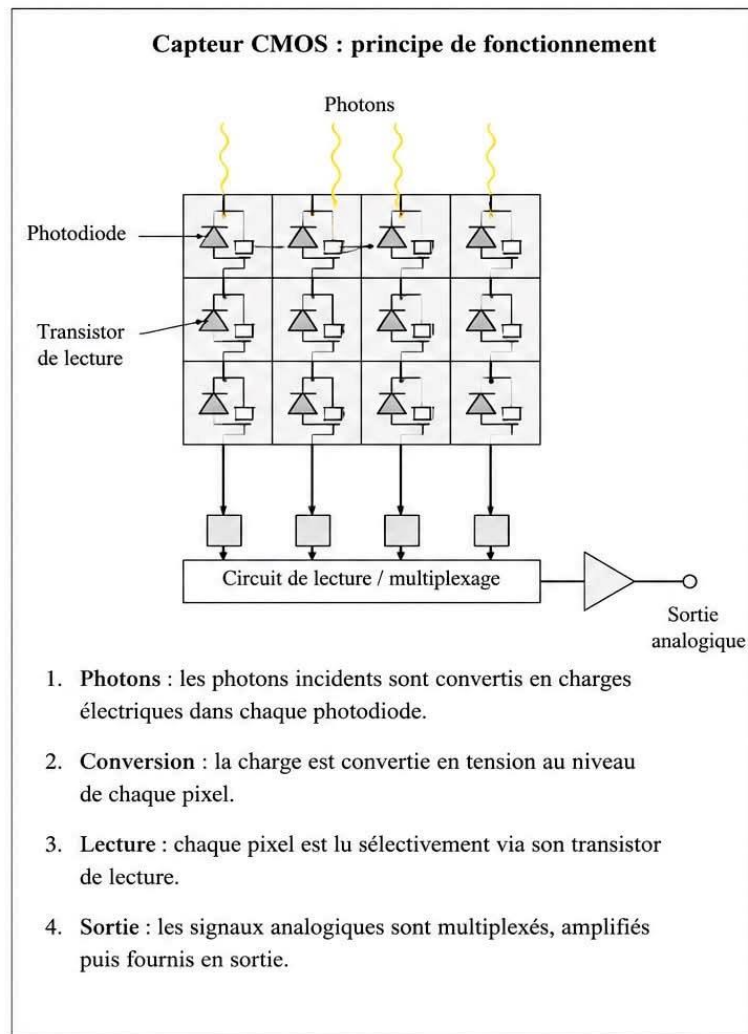


Figure I-10 : Principe de fonctionnement du capteur CMOS

I.5.2 Contraintes liées à l'éclairage

Là où le capteur serait l'œil du système, l'éclairage constitue sa source de lumière. De même qu'il n'est pas possible de distinguer une image sans un bon éclairage pour l'œil humain, la qualité de l'image vue dépend de l'environnement lumineux dans lequel elle est analysée. Pour les systèmes embarqués conçus pour lire des codes-barres ou des codes QR, l'éclairage n'est pas optionnel : il s'agit d'un paramètre crucial qui affecte directement l'efficacité du décodage [29].

- **Importance du contraste**

Un code-barres ou code QR se base fondamentalement sur le contraste entre ses parties noires et blanches. Un éclairage insuffisant, non-homogène ou causant des phénomènes de réflexion spéculaire pourrait entraîner la perte du contraste nécessaire à la lecture de certaines barres ou

Chapitre I:Généralités sur les systèmes embarqués et la technologie des codes-barres

modules. Une étude a montré que la présence d'une interférence lumineuse issue de LED environnantes, modulées en largeur d'impulsion (PWM), qui produit un léger scintillement invisible par l'œil humain mais détectable par le capteur, pouvait causer un échec de lecture récurrent dans des environnements industriels [18]

- **Types d'éclairage et géométries**

L'importance de la géométrie d'éclairage n'est pas moins grande que celle même du capteur. La figure qui illustre ces différentes géométries [29]

Eclairage incident (Frontal) : la source de lumière illumine l'objet de la même façon que l'axe optique de la caméra. Très indiqué pour la lecture des codes-barres et codes QR imprimés sur une surface mate, il maximise le contraste des traits de cet objet. C'est la géométrie la plus fréquente dans les lecteurs embarqués.

Eclairage diffuse (dôme) : un éclairage homogène illumine l'objet de partout, éliminant tous les reflets. Particulièrement recommandé pour les objets brillants et/ou courbes comme les étuis plastiques et les étiquettes métalliques.

Rétroéclairage (Contrejour) : l'objet se trouve devant une source lumineuse uniforme. Lorsqu'elle est suffisamment forte, elle donne une silhouette tranchée bien visible, mais qui ne convient pas à la lecture de code-barres.

- **Sources lumineuses embarquées**

Dans les petits systèmes embarqués, les LED ont été le choix standard pour le faisceau d'éclairage. Elles ont la qualité d'avoir une excellente relation entre la performance et le coût, une longévité élevée, une faible consommation de puissance et une commande électronique possible de l'intensité et de la synchronisation avec l'acquisition [29]. L'usage des longueurs d'onde infrarouges est courant dans les applications industrielles, car il aide à éviter les interférences par lumière visible extérieur

I.5.3 Résolution et qualité d'image

La résolution d'un capteur se réfère au nombre de pixels utilisables pour représenter la scène photographiée. Elle est un critère fondamental, mais fréquemment mal compris car une résolution

Chapitre I: Généralités sur les systèmes embarqués et la technologie des codes-barres

plus élevée n'implique pas automatiquement une qualité d'image supérieure, notamment lorsque le calculateur est sous-dimensionné [30].

- Résolution spatiale et déchiffrement de codes

Dans le cas des codes-barres linéaires, la résolution du système doit être suffisamment élevée pour permettre la discrimination des plus fines barres du code, dont le minimum possible de largeur, appelé module X, est à peu près de quelques dixièmes de millimètres en cas de formats denses tels que le Code 128 [27]. En effet, si la résolution est trop faible pour pouvoir distinguer la fine barre, celle-ci sera floutée et donc non reconnue par le décodeur.

Contrairement aux barcodes, pour les QR Codes, le paramètre qui impose une contrainte est l'imagerie du code. Il faut que chaque module de la matrice soit imprimé en utilisant au minimum de deux à trois pixels pour que le code soit facilement décodable. Généralement, une résolution minimale de 640 x 480 pixels (VGA) suffit pour une lecture jusqu'à 50 centimètres, alors que des résolutions supérieures sont nécessaires dans d'autres cas.

- Le rapport signal sur bruit (SNR)

L'aspect qualitatif d'une image est multidimensionnel. Le rapport signal sur bruit (Signal to Noise ratio ou SNR) mesure de manière importante le niveau de confiance de l'information contenue dans l'image. C'est la relation entre l'amplitude du signal utile et l'amplitude du bruit dans l'image [22].

Pour les systèmes intégrés de vision industrielle, l'admissibilité au SNR dans une image est de 30 décibels pour les applications courantes, alors que les activités nécessitant une grande précision, telles que la biologie biométrique et l'optique, requièrent un rapport signal sur bruit qui dépasse 40 décibels [31]. Un rapport signal sur bruit faible entraîne du bruit sur les pixels, une perte de contraste ainsi qu'un risque élevé de fausses lectures des codes [14].

Parmi les paramètres qui peuvent affecter le rapport signal sur bruit dans les capteurs CMOS figurent les suivants [32]

- Taille des pixels : des grands pixels absorbent plus de lumière et génèrent un rapport signal bruit meilleur, bien que la qualité de résolution soit compromise.

Chapitre I: Généralités sur les systèmes embarqués et la technologie des codes-barres

- Temps d'exposition : plus de temps d'exposition améliore le rapport signal sur bruit, mais augmenterait le risque de flou de mouvement pour un objet mobile.
- Qualité de l'éclairage : Un éclairage uniforme et intense améliore le rapport signal/bruit car il amplifie le signal utile.
- Température : Les capteurs CMOS produisent du bruit thermique, qui augmente avec la température.
- Proximité d'image et netteté

La proximité d'image (DoF, Depth of Field) est l'intervalle des distances où l'image demeure nette. Dans les systèmes embarqués avec objectif fixe, elle est calculée à partir de la valeur de l'ouverture et du coefficient focal. Une faible DoF peut rendre le système vulnérable à toute variation de la distance entre le code et la caméra, ce qui est courant lorsque le code est en mouvement [23].

- **L'utilisation des IA.**

L'intelligence artificielle (IA) et surtout le deep learning ont complètement bouleversé la conception des techniques d'analyse d'image dans les systèmes embarqués. Au lieu de méthodes historiques de vision par ordinateur qui se basaient sur l'utilisation explicite des techniques de seuillage, détecteur de contours, gradient, etc., la nouvelle génération d'algorithmes apprend des bases de données massives d'images annotées [33]

- **Réseaux de neurones convolutifs (CNN)**

Réseaux de neurones convolutifs (CNN, Convolutional Neural Network) sont un élément essentiel pour la nouvelle génération de systèmes de vision par ordinateur. Structure du réseau est basée sur le fonctionnement du cortex visuel humain : une série de filtres convolutifs sont appliqués aux images afin de trouver des caractéristiques plus et plus complexes de ces dernières, comme des bords, puis des formes et enfin des objets entiers [30].

Dans le cas de la lecture de barcodes et de codes QR, les CNN ont montré leur efficacité pour localiser et reconnaître les codes, malgré les difficultés : images floues, inégale luminosité, code partiellement caché, rotation importante et/ou fond d'image complexe. Des architectures telles que AlexNet, ResNet et DenseNet201 ont donc été modifiées afin d'être utilisées pour la classification et la reconnaissance des codes, avec des taux de précision largement supérieurs aux techniques classiques [34].

Chapitre I: Généralités sur les systèmes embarqués et la technologie des codes-barres

- **YOLO et la détection en temps réel**

Le fait de devoir réaliser une analyse de l'image dans un temps inférieur à quelques millisecondes est un des grands problèmes rencontrés dans la conception d'IA embarquées. Effectuer la reconnaissance avec un réseau neuronal profond pourrait en effet prendre plusieurs dizaines de millisecondes, trop longtemps lorsque le nombre de trames par seconde doit être plusieurs dizaines. La solution apportée par Redmon et al. au travers de l'architecture YOLO (You Only Look Once) était de prendre en compte toute l'image à chaque passage dans le réseau, permettant ainsi une détection en temps réel [\[35\]](#)

Dernièrement, les performances de YOLOv5, YOLOv8 et Faster R-CNN ont été évaluées sur des jeux de données de codes-barres dans des environnements industriels complexes. YOLOv8 parvient notamment à obtenir un mAP (mean Average Precision) de 92,4 % pour la détection de codes-barres 1D et 2D dans des conditions qui imitent un environnement d'entrepôt avec éclairages variables et occlusions partielles [\[35\]](#) De tels algorithmes, optimisés et quantifiés pour leur déploiement sur des processeurs embarqués d'architecture ARM ou FPGA, constituent une technologie robuste et accessible aux lecteurs intelligents [\[36\]](#)

- **L'IA au service de la robustesse**

L'IA ne se limite pas à une simple détection, mais contribue également à lutter contre le bruit d'une image afin d'en améliorer la qualité au moment de son passage à l'étape suivante. Des débruiteurs neuronaux sont capables de réduire de manière temporelle les défauts présents dans une image perturbée pour l'envoyer ensuite au décodeur [\[30\]](#). Pour lire des codes barres au-delà de leurs limites habituelles ou pour augmenter leurs capacités de lecture en présence de mauvais optiques, les modèles de super-résolution reconstituent les détails manquants d'une image. Finalement, une technique d'intégration entre les deux méthodes qui utilise YOLO pour localiser et des CNN pour classer offre la possibilité de prendre en compte plusieurs types de code dans une même image, ce qui donne lieu aux lecteurs universels [\[32\]](#).

I.6 Travaux existants et solutions similaires

I.6.1 Système utilisé en Algérie :

L'Algérie, comme beaucoup d'autres pays en cours de passage à la numérisation, a entrepris depuis la fin du vingtième siècle une progression progressive vers l'introduction des technologies

Chapitre I:Généralités sur les systèmes embarqués et la technologie des codes-barres

de codage automatique. Ce processus s'est tout d'abord buté sur de multiples obstacles structurels : coûts des matériels, insuffisance des formations, faiblesse des infrastructures numériques ; il aura néanmoins été fortement boosté par les initiatives des pouvoirs publics et la mise en œuvre des normes internationales.

L'acteur principal de la normalisation des codes-barres en Algérie est l'Association Algérienne de Codification des Articles GS1 Algeria qui a été mise en place sous la supervision du Ministère du Commerce. Il s'agit de l'unique organisme officiel chargé d'allouer des codes-barres à toutes les entreprises conformément aux normes techniques internationales sous le préfixe « 613 » réservé aux produits algériens [\[37\]](#).

Jusqu'à une certaine époque, l'utilisation des codes-barres était optionnelle. Mais le virage a été pris en 2020 avec la prise de conscience des pouvoirs publics pour accélérer le processus. Le ministre du Commerce avait annoncé que 10 000 entreprises économique allaient utiliser le code à barres emballeur dont 450 000 produits algériens sont codés par GS1 Algeria [\[38\]](#).

Au cours du mois de décembre 2022, le gouvernement algérien a marqué un tournant avec l'annonce de l'imposer la mention de code-barres à tous les produits pour la consommation humaine au départ du 29 mars 2023. Au même moment, 11 410 entreprises ont obtenu le code-barres mentionné sur plus de 500 000 produits.

" Toute marchandise qui n'est pas dotée d'un code barre n'est plus juridique." Le présent règlement légal est une force motrice importante dans le sens où il aide à accélérer la numérisation commerciale en Algérie.

D'après le Ministère du Commerce, le processus de codification aidera à assurer la sécurité des produits, à protéger les consommateurs, à garantir la traçabilité, et à combattre la contrefaçon, ainsi qu'à favoriser le développement du e-commerce en Algérie. [\[38\]](#)– Plus de 11 000 entreprises ont obtenu le code-barres

I.6.2 Limites des solutions actuelles :

Bien que de grands progrès aient été faits par rapport aux méthodes traditionnelles de lecture des codes-barres et des QR codes, ces derniers, actuellement employés, ont néanmoins une série

Chapitre I:Généralités sur les systèmes embarqués et la technologie des codes-barres

de limitations structurales qui font apparaître le besoin d'un développement de nouvelles technologies, en particulier en ce qui concerne les systèmes embarqués intelligents.

- Dépendance aux équipements étrangers et importés

Pratiquement tous les scanners professionnels employés en Algérie sont des appareils importés et leur prix se situe entre 50 et 300 euros pour des modèles professionnels câblés [39]. Les défauts qui en résulteront seront nombreux : coûts d'acquisition élevés pour les PME algériennes, dépendance de ces mêmes entreprises vis-à-vis des fabricants étrangers pour la maintenance et le remplacement de pièces, et sensibilité face aux variations de l'évolution de la monnaie nationale et des conditions d'importation. Actuellement il n'existe pas encore de produits nationaux de lecture embarqués.

- Limites des scanners laser 1D

Les douchettes lasers traditionnels, bien que fiables et rapides, ont néanmoins une série de limites techniques propres à leur conception [40] :

- Uni directionnalité de lecture : la balise laser doit être disposée de façon horizontale en relation avec le code-barres, qui doit donc être disposé dans l'alignement exact de la balise pour que celle-ci puisse le lire. L'inclinaison du code ou sa présentation sur une surface courbe peut nécessiter plusieurs essais.[40]
- Incapacité de lire les codes 2D : les scanners laser 1D sont incapables de lire les codes QR, les matrices data ou autres formes bidimensionnelles de code-barres. Ceci, à l'époque où ces formes de codage prennent de l'importance, oblige alors les entreprises à posséder plusieurs systèmes de scan.
- Fragilité face aux conditions d'impression dégradées : les scanner lasers sont particulièrement sensibles à la qualité d'impression, et ne permettent de lecture que des codes-barres clairs et parfaitement lisibles, car ces scanners n'ont aucun moyen de rectification d'erreur [16].
- Absence d'intelligence embarquée : les scanners ne font qu'exporter les informations collectées vers l'ordinateur sans réaliser aucune analyse ni prise de décision.

Chapitre I: Généralités sur les systèmes embarqués et la technologie des codes-barres

I.6.3 Limites liées à la connectivité et à l'architecture système

L'infrastructure de lecture installée en Algérie se fonde principalement sur des architectures **client/serveur centralisées** : la balise diffuse les données vers un ordinateur qui interroge à distance une base de données contenant l'information associée. Cette architecture souffre notamment de limites importantes :

D'une forte dépendance vis-à-vis de la connectivité réseau, puisque tout dysfonctionnement Internet ou réseau local entraîne la désactivation complète du dispositif de lecture.

De délais d'information liés à l'interrogation à distance de la base de données, insuffisants pour certains usages temps réel nécessitant une forte cadence.

D'absence totale de fonctionnalité hors connexion, qui n'est pas envisagée pour l'instant.

I.6.4 Limitations des solutions mobiles grand public

L'utilisation de la technologie smartphone pour lire des codes-barres via application mobile, malgré son universalisation, comporte plusieurs inconvénients lorsqu'elle est mise en œuvre dans un cadre professionnel [\[40\]](#) [\[16\]](#) :

- Variabilité de la performance : la lisibilité varie en fonction du téléphone utilisé, de l'application qui permet de lire les codes-barres et des conditions lumineuses.
- Difficulté d'intégration au SI de l'entreprise : il n'existe aucune interface naturelle entre ces applications grand public et les ERP, WMS ou systèmes d'inventaires professionnels.
- Problème de sécurité des données : le scan via smartphone est en général associé à l'envoi des données vers un serveur externe et non géré par l'entreprise.
- Inadaptées à des conditions industrielles : un téléphone portable n'est pas conçu pour être solide et résister aux conditions de stockage ou d'usine (choc, poussières, humidité).

I.6.5 Faible diffusion des technologies IA dans les solutions locales

Bien que l'efficacité de l'architecture de Deep Learning utilisée pour la reconnaissance de codes soit confirmée par diverses recherches scientifiques internationales [\[33, 35\]](#) parmi celles utilisant des modèles tels que YOLO ou CNN, leur utilisation dans les systèmes en place est presque absente. Ces solutions ne comportent que des algorithmes de décodage traditionnels, qui

Chapitre I:Généralités sur les systèmes embarqués et la technologie des codes-barres

ne sont pas capables de prendre en compte des images abîmées, des angles de reconnaissance impossibles, des codes masqués partiellement ou non-conformes.

I.7 Conclusion :

Ce premier chapitre a permis de mettre en lumière le rôle déterminant des systèmes embarqués dans la transformation numérique des secteurs industriels, économiques et institutionnels, en soulignant leur contribution essentielle à l'automatisation, à l'optimisation des processus et au contrôle intelligent des dispositifs électroniques. À travers l'étude de la plateforme ESP32 et de son module ESP32-CAM, il a été démontré que les microcontrôleurs modernes offrent désormais des capacités avancées en matière de traitement des données, de connectivité et d'acquisition d'images, ouvrant ainsi la voie à des applications embarquées innovantes intégrant des fonctionnalités de vision artificielle et d'analyse en temps réel.

En parallèle, l'analyse de la technologie des codes-barres a confirmé son statut de solution mature, largement déployée dans les systèmes de gestion d'actifs et d'inventaire, notamment en Algérie à travers le standard GS1. Toutefois, malgré ses avantages indéniables en termes de simplicité, de coût et de fiabilité, cette technologie présente des limitations structurelles, notamment en matière de capacité de stockage, de sécurité des données et d'interopérabilité avec les architectures numériques émergentes.

Ainsi, ce chapitre établit un cadre conceptuel et technologique solide, indispensable à la compréhension des enjeux liés à la conception de systèmes embarqués dédiés à la gestion intelligente des actifs. Il constitue également un socle fondamental pour la suite de ce travail, qui s'attachera à concevoir et à développer une solution embarquée innovante, adaptée aux spécificités du contexte algérien, en combinant les apports des technologies étudiées.

I.7.1 Perspectives de travaux futurs :

Dans la continuité de ce travail, plusieurs axes de recherche et de développement peuvent être envisagés :

- **Intégration de l'intelligence artificielle embarquée :** développement d'algorithmes de vision par ordinateur (reconnaissance d'objets, détection automatique de codes-barres) directement sur des microcontrôleurs.

Chapitre I: Généralités sur les systèmes embarqués et la technologie des codes-barres

- **Hybridation des technologies (codes-barres / RFID)** : conception de systèmes combinant les avantages des deux approches pour une traçabilité plus complète et flexible.
- **Optimisation énergétique** : amélioration de la consommation des systèmes embarqués afin de prolonger leur autonomie dans des environnements contraints.
- **Développement de solutions IoT avancées** : intégration avec des plateformes cloud pour le suivi en temps réel, l'analyse de données et la prise de décision intelligente.
- **Sécurisation des données** : mise en place de mécanismes de chiffrement et d'authentification pour garantir l'intégrité et la confidentialité des informations échangées.
- **Adaptation au contexte algérien** : conception de solutions adaptées aux contraintes locales (coût, infrastructure, environnement industriel), favorisant leur adoption à grande échelle.

En définitive, ce travail constitue une contribution à la compréhension et à l'exploitation des systèmes embarqués dans le domaine de la traçabilité intelligente. Il ouvre la voie à des solutions innovantes, capables de répondre aux exigences croissantes des environnements modernes en matière de gestion automatisée, connectée et sécurisée des ressources.

Chapitre II:Analyse et spécifications du système

Chapitre II:Analyse et spécifications du système

II.1 Introduction

La gestion des examens et de leurs résultats fait partie intégrante des activités réalisées au sein des institutions d'enseignement supérieur, étant donné qu'elle affecte directement la qualité du processus d'évaluation pédagogique, la fiabilité des résultats et l'intégrité de l'institution universitaire. En effet, une bonne évaluation devrait se caractériser par une rapidité dans le traitement des travaux de contrôle, une minimisation des erreurs humaines, une protection des données et une équité dans la notation entre les étudiants. Cependant, dans le contexte algérien, on constate que beaucoup d'universités font appel à des méthodes traditionnelles de gestion de ces activités, qui reposent essentiellement sur des procédures manuelles [\[41\]](#)[\[42\]](#)[\[43\]](#).

Toutefois, ces techniques traditionnelles comportent plusieurs défauts. L'utilisation manuelle des copies de test entraîne une augmentation significative du risque d'erreurs administratives, notamment lors de la reconnaissance de l'identité des élèves, l'entrée des scores ou l'archivage des formulaires. En outre, l'emploi d'un processus manuel est coûteux en termes de temps, surtout dans les écoles comportant beaucoup d'élèves, provoquant souvent des retards dans la distribution des scores et une charge de travail supplémentaire pour les professeurs et le personnel administratif. Ces facteurs peuvent causer des problèmes de gestion, une absence d'homogénéité dans les procédures d'évaluation ainsi que réduire l'efficacité globale du système d'évaluation.

D'autre part, le développement des systèmes universitaires modernes révèle l'accentuation de la nécessité d'intégration de la transformation numérique dans le secteur éducatif. En effet, malgré les tentatives des universités algériennes pour moderniser leur infrastructure et pour améliorer la qualité des services universitaires, l'utilisation de systèmes numériques modernes spécifiquement conçus pour le travail académique reste insuffisante dans nombre d'institutions [\[42\]](#)[\[43\]](#). Le système utilisé ne se limite généralement qu'à la gestion administrative globale sans prévoir un ensemble complet qui permettrait une automatisation efficace du processus d'examen et notation.

Dans un tel contexte, le recours à des solutions technologiques reposant sur l'automatisation et l'anonymisation du traitement des copies paraît particulièrement pertinent afin de favoriser l'efficacité, la transparence et la sécurité des procédures académiques. En effet, la mise en œuvre de technologies intégrées, d'outils d'identification automatisée et de procédés centralisés de

Chapitre II:Analyse et spécifications du système

traitement des informations permettrait d'éviter les interventions humaines, de faciliter la collecte et le traitement des copies, et de réduire les risques d'erreur ou de fraude. Par ailleurs, l'anonymisation des copies favorise la notion d'impartialité dans les procédures de notation.

La solution proposée dans ce travail repose sur une plateforme intelligente nommée **ANONYMOS OS**. Elle est basée sur l'utilisation de technologies embarquées associées à un système de gestion de données centralisé. Cette plateforme permet de garantir une identification rapide des copies d'examen, un suivi automatique des processus de traitement ainsi qu'une gestion optimisée des résultats académiques. Le but est aussi d'optimiser les délais de traitement des examens, de faciliter la gestion administrative et le stockage des données dans les institutions universitaires.

II.2 Présentation de la problématique

Dans le cas de l'approche de gestion de l'examen qui est encore majoritairement manuelle dans beaucoup d'universités algériennes, différentes contraintes d'organisation et administratives entravent le bon déroulement de ce processus. Le fait que ce dernier n'est pas automatisé, au niveau de la collecte, du traitement et de la gestion des notes provoque souvent des retards importants dans la publication des résultats et cela affecte directement le bon déroulement de certaines activités pédagogiques et administratives [\[41\]](#). Les activités de distribution de copies, de collecte des copies, de correction, de saisie des notes et de délibération sont effectuées manuellement et prennent beaucoup de temps et d'efforts humains.

Ce fait contribue à créer un surcroît de travail significatif pour les enseignants ainsi que pour le personnel administratif. En effet, outre la tâche consistant à évaluer manuellement les copies d'examen, il faut également faire face à celle relative à la saisie des notes dans divers documents administratifs ou sites web, parfois même plusieurs fois. Ceci n'est pas sans augmenter grandement le temps nécessaire pour évaluer l'examen et multiplier les erreurs en cours de transcription des notes, des moyennes et même des procès-verbaux. Dans le cas d'universités nombreuses, ces aspects sont rendus plus difficiles encore.

D'autre part, une des problématiques notées dans les systèmes traditionnels est l'absence d'anonymat lors du processus de correction. D'une manière générale, les copies des examens sont porteuses des informations personnelles de l'étudiant, à savoir son nom et son numéro

Chapitre II:Analyse et spécifications du système

d'inscription. Ceci peut avoir un impact psychologique sur le correcteur et nuire aux principes de justice, de neutralité et d'objectivité qui doivent prévaloir lors de tout processus de correction académique [42]. Cela tient au fait que même sans la volonté de faire preuve de partialité, la simple connaissance de l'étudiant pourrait induire un biais mental chez le correcteur.

En outre, les procédures manuelles contribuent considérablement aux erreurs humaines. Les erreurs de saisie, la confusion des notes, le calcul incorrect des moyennes ainsi que l'oubli de faire des copies sont parmi les défis couramment observés dans les systèmes traditionnels [43]. Ces erreurs peuvent causer des contradictions administratives, des réclamations des étudiants et des retards additionnels causés par les vérifications et les corrections requises. De plus, elles ont un impact sur la qualité des résultats académiques et peuvent ternir la réputation de l'université.

La nécessité de la présence d'un système centralisé et automatisé cause également une difficulté pour l'archivage, le suivi et la consultation de données académiques. Des données sur les examens sont dispersées dans différents documents papier et documents informatiques, ce qui entraîne des problèmes de rapidité de recherches de données, de suivi de copies et de formation automatique de statistiques pédagogiques. Cela empêche une analyse efficace et un bon contrôle par les dirigeants pédagogiques et administratifs.

En réponse à ces difficultés, beaucoup de chercheurs dans la spécialité de gestion de la qualité dans l'enseignement supérieur soulignent la nécessité de modernisation des systèmes universitaires au moyen de l'introduction d'une solution numérique et automatisée [41][42]. L'automatisation de processus de travail avec examens non seulement aidera à minimiser le temps nécessaire à la réalisation de cette tâche et éliminer le danger de fautes humaines, mais aussi améliorera la sécurité, la transparence et l'efficacité du processus en général. De plus, mécanismes de masquage des copies et d'identification automatique peuvent être introduits pour garantir objectivité des données académiques.

II.3 Cahier des charges fonctionnel

Le système ANONYMOS OS repose sur un ensemble de fonctionnalités permettant d'assurer une gestion automatisée et anonymisée des examens. Chaque étudiant reçoit un code-barres unique pour chaque module, garantissant l'anonymat des copies lors du processus de correction.

Chapitre II:Analyse et spécifications du système

Afin d'assurer la continuité du fonctionnement en cas d'imprévu, trois exemplaires de chaque code-barres sont générés pour chaque étudiant et pour chaque module, garantissant ainsi la disponibilité de copies supplémentaires en cas de besoin, dont un exemplaire réservé à l'enseignant pour le suivi et le contrôle.

Le module ESP32-CAM S3, qui est un microcontrôleur doté d'une caméra intégrée et d'une connectivité Wi-Fi, capture automatiquement les images des codes-barres présents sur les copies corrigées. Ces images sont ensuite transmises vers un système central chargé du traitement des informations collectées [6]. Au niveau du système central, les images capturées sont analysées afin de décoder les codes-barres correspondants. Chaque code est ensuite associé à l'étudiant correspondant dans la base de données, ce qui permet d'enregistrer les notes de manière structurée et sécurisée. Les données ainsi stockées permettent le calcul automatique des moyennes, la génération de rapports et la consultation organisée des résultats par les utilisateurs autorisés.

L'organisation des différentes fonctionnalités du système s'appuie sur les principes recommandés par les méthodes de génie logiciel, qui soulignent l'importance d'une structuration claire des fonctions et d'une gestion cohérente des interactions entre les composants matériels et logiciels [19][44].

II.4 Solution proposée

La solution proposée ANONYMOS OS repose sur la mise en place d'un système automatisé visant à améliorer la gestion des examens et le traitement des notes au sein des établissements universitaires. Cette solution s'appuie sur l'intégration d'une base de données centralisée, de technologies d'identification par codes-barres et d'un système d'acquisition d'images basé sur un module embarqué. L'objectif principal est d'assurer une gestion plus rapide, plus fiable et plus transparente du processus d'évaluation académique.

Le fonctionnement du système ANONYMOS OS débute par l'exploitation des informations stockées dans la base de données académique. Cette base de données regroupe plusieurs catégories d'informations nécessaires à l'organisation des examens, notamment les données relatives aux étudiants, les modules d'enseignement, l'emploi du temps des examens, les salles

Chapitre II:Analyse et spécifications du système

d'examen ainsi que les enseignants responsables des modules. L'ensemble de ces informations permet de structurer et d'organiser efficacement le processus de gestion des examens.

À partir de ces données, le système génère automatiquement des codes-barres uniques pour chaque étudiant et pour chaque module. Cette génération est réalisée directement à partir des informations stockées dans la base de données, afin d'assurer une correspondance précise entre les étudiants, les modules et les examens concernés. Les codes-barres ainsi générés servent d'identifiants anonymes pour les copies d'examen, ce qui permet de dissocier l'identité de l'étudiant du processus de correction et de garantir l'objectivité de l'évaluation.

Chaque code-barres est ensuite imprimé en plusieurs exemplaires afin de faciliter son utilisation lors des examens et d'assurer la disponibilité de copies supplémentaires en cas de perte ou de détérioration. Les étudiants utilisent ces codes-barres pour identifier leurs copies d'examen sans mentionner directement leur identité. Cette méthode contribue à renforcer l'anonymat durant la phase de correction et à améliorer l'équité dans l'évaluation des étudiants.

Après la correction des copies par les enseignants, le système permet de collecter automatiquement les informations associées aux copies corrigées. Pour cela, un module ESP32-CAM S3, équipé d'une caméra intégrée et d'une connectivité réseau, est utilisé pour capturer les images des codes-barres présents sur les copies. L'utilisation de ce module permet d'automatiser le processus d'acquisition des données et de réduire considérablement les opérations de saisie manuelle.

Les images capturées sont ensuite transmises vers un système central via une connexion Wi-Fi. Le système central reçoit ces images et applique des techniques de décodage afin d'extraire les informations contenues dans les codes-barres. Les données obtenues sont ensuite comparées avec celles enregistrées dans la base de données, afin d'identifier l'étudiant correspondant, le module concerné ainsi que les informations associées à l'examen.

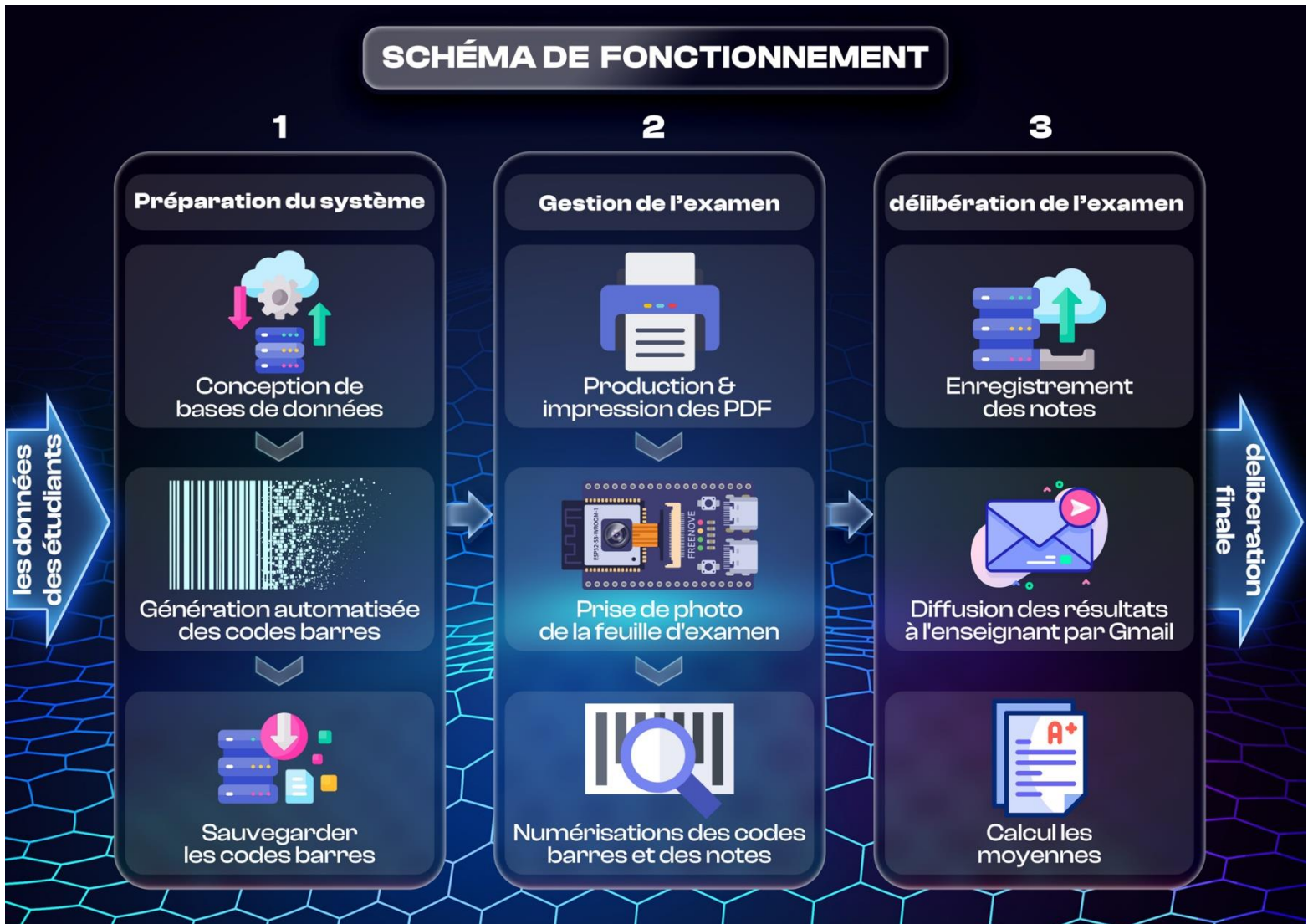
Une fois cette correspondance établie, les notes attribuées aux copies corrigées sont enregistrées dans la base de données centralisée. Cette centralisation des informations permet d'organiser les résultats académiques de manière structurée et facilite leur exploitation ultérieure. Les données stockées peuvent ensuite être exploitées pour effectuer différentes opérations

Chapitre II:Analyse et spécifications du système

administratives, telles que le calcul automatique des moyennes, la préparation des délibérations ou encore la génération de rapports statistiques.

En intégrant l'ensemble des informations relatives aux étudiants, aux modules, aux enseignants, aux salles d'examen et à l'emploi du temps des examens, le système assure une gestion cohérente et complète du processus d'évaluation académique. Cette approche permet d'améliorer la fiabilité du traitement des données, de réduire les erreurs humaines et d'accélérer la publication des résultats.

Ainsi, la solution proposée combine l'utilisation d'une base de données académique, la génération automatique des codes-barres, l'utilisation du module ESP32-CAM S3 pour l'acquisition des images, ainsi qu'un système central dédié au traitement et au stockage des informations. Cette architecture permet de mettre en place un système efficace, fiable et adapté aux besoins des établissements universitaires modern.



Chapitre II:Analyse et spécifications du système

II.4.1 Préparation du système :

La phase de préparation du système ANONYMOS OS constitue l'étape initiale et essentielle de notre projet. Elle vise à mettre en place toutes les structures nécessaires au bon fonctionnement de la gestion des examens, depuis la création et l'organisation des données jusqu'à la génération des supports utilisés par les étudiants.

Cette étape comprend trois volets principaux : la conception et l'initialisation de la base de données, la génération automatisée des codes-barres, et la sauvegarde de ces derniers. L'objectif est d'assurer la fiabilité, la traçabilité et la rapidité des opérations avant le déroulement de l'examen, en facilitant le traitement automatique des informations tout au long du processus

- Conception de la base de données

La conception de la base de données est l'un des points cruciaux dans la mise en place du système de ce travail. En effet, la totalité des opérations effectuées en rapport avec l'organisation des examens est basée sur une bonne structuration des données de manière à assurer leur fiabilité, leur traçabilité et surtout leur automatisation lors de leur manipulation.

Au sein de ce projet, la base de données est considérée comme l'ossature principale du système ANONYMOS OS. Elle sera chargée de collecter, classer et utiliser toutes les données nécessaires concernant les étudiants, les modules, les examens, les enseignants, les salles de classe et les scores de notation obtenus après le traitement automatique des copies des examens.

Au regard de ce projet, il a été choisi le système SQLite qui est un SGBD relationnel léger, portable et facile à implémenter pour le langage Python. Voici quelques raisons pour justifier ce choix :

- Facilité d'utilisation .
- Absence de serveur centralisé.
- Faible utilisation de ressources informatiques.
- Rapide exécution
- Compatibilité nativement avec le langage Python via la librairie sqlite3.

Chapitre II:Analyse et spécifications du système

Cette approche serait très utile pour une application locale comprenant plusieurs modules logiciels interdépendants.

Le modèle de la base de données est conçu pour assurer l'incapacité à identifier anonymement les copies d'examen, l'intégrité des données, le contrôle des procédures d'examen, l'automatisation de l'enregistrement des notes par les enseignants, et aussi la commodité du contrôle et délibérations de copies par les enseignants.

L'utilisation de relations dans le schéma de la base de données permet non seulement éviter la répétitivité des données, mais aussi effectuer les actions liées à l'utilisation de clés primaires et étrangères.

La base de données est structurée autour de neuf tables principales, chacune ayant un rôle spécifique et interconnecté, afin de garantir la cohérence et l'intégrité des données.

- Génération automatisée des codes-barres

La génération de codes-barres est conçue pour faciliter la gestion et le suivi des informations relatives aux examens. Chaque enregistrement d'examen est associé à un code-barre unique qui encode des identifiants essentiels. Une clé de sécurité est intégrée lors de la génération, ce qui empêche toute personne non autorisée de visualiser le contenu du code-barres. Le processus est entièrement automatisé : le système génère un code-barres pour chaque examen dans la base de données et l'enregistre directement dans la colonne correspondante. Cela permet une récupération et une vérification rapides. De plus, l'intégration de la génération de codes-barres améliore la sécurité des données des examens. En centralisant la création des codes-barres dans le système, le projet garantit la cohérence de tous les enregistrements d'examen et simplifie les processus en aval

- Sauvegarde des codes-barres

Une fois générés, les codes-barres sont enregistrés et stockés directement dans la base de données, dans le champ correspondant à chaque enregistrement d'examen. Cette étape est importante car elle permet de conserver chaque code-barres de manière structurée et organisée, tout en l'associant aux informations relatives à l'étudiant et à l'examen concerné.

Chapitre II:Analyse et spécifications du système

La sauvegarde des codes-barres dans la base de données garantit également leur disponibilité pour les étapes suivantes du système, notamment lors de la phase de numérisation et de traitement des copies. Grâce à cette centralisation, le système peut facilement accéder aux codes-barres et récupérer les informations nécessaires sans risque de perte de données.

De plus, cette méthode de stockage contribue à renforcer la traçabilité des examens. Chaque code-barres est lié à un enregistrement précis, ce qui permet d'identifier rapidement une copie et de vérifier les informations associées. Ainsi, la sauvegarde des codes-barres dans la base de données participe à la fiabilité globale du système et facilite la gestion automatique des informations tout au long du processus d'examen.

II.4.2 Gestion de l'examen

Dans **ANONYMOS OS**, la gestion de l'examen constitue l'une des phases les plus critiques du système, car elle englobe l'ensemble des processus nécessaires pour assurer le suivi, l'organisation et la sécurisation des copies des étudiants depuis leur préparation jusqu'à l'enregistrement des notes. Elle joue un rôle central dans la cohérence globale du système, garantissant que toutes les informations relatives aux étudiants, aux modules et aux sessions d'examen sont correctement reliées et protégées.

La gestion inclut plusieurs étapes interconnectées : la production des listes des étudiant sous format PDF avec le code-barres, la capture photographique des copies et la numérisation des informations pour leur enregistrement dans la base de données. L'intégration de ces fonctionnalités permet non seulement de réduire les risques d'erreurs humaines, mais aussi d'améliorer la rapidité, la fiabilité et la sécurité du traitement des examens. Chaque étape est conçue pour s'assurer que les copies sont correctement identifiées, suivies et validées tout au long du processus.

- Production et impression des listes

La première étape consiste à générer automatiquement les listes sous format PDF à partir des données existantes dans la base de données. Ces documents contiennent les codes-barres uniques associés à chaque étudiant et chaque examen, permettant une identification fiable et un suivi précis des copies. La génération automatisée des **listes** de code-barres garantit que chaque étudiant reçoit une copie correctement liée à son identité et à son examen.

Chapitre II:Analyse et spécifications du système

Tout étudiant de la liste d'examen reçoit une série de trois codes-barres uniques pour permettre une identification précise et un bon suivi de la copie de l'examen. Le premier code-barres est appliqué sur la copie principale de l'étudiant et il sert d'identification pendant toute la procédure de collecte, de tri et de correction. Les deux autres codes-barres sont destinés aux feuilles intercalaires et copies supplémentaires utilisées si l'espace sur la copie principale ne suffit pas. Ainsi grâce à cette technique, toutes les feuilles liées à un seul étudiant peuvent être liées et identifiées facilement sans risque de confusion ou de perte. C'est une très bonne technique qui facilite grandement la gestion des examens en réduisant les erreurs humaines dans le suivi des copies. De plus, elle permet une numérisation rapide et la correction des copies, surtout dans les institutions qui reçoivent de nombreux étudiants.

- Prise de photo des feuilles d'examen

Traditionnellement, la numérisation des copies d'examen implique l'utilisation de scanners, ce qui peut être long et complexe, notamment pour un grand nombre d'étudiants. Dans ce système, chaque copie est photographiée directement par le programme. Cette approche permet de capturer rapidement toutes les copies tout en conservant la qualité nécessaire pour l'analyse ultérieure.

La prise de vue assure que toutes les images sont uniformes en termes de luminosité, de cadrage et de résolution, ce qui est crucial pour la lecture fiable des codes-barres et des notes. De plus, cette méthode minimise les interventions manuelles, réduit le risque de perte ou de mélange des copies et permet un traitement simultané de plusieurs documents.

- Numérisation des codes-barres et des notes

Une fois les images capturées, le programme procède à l'extraction automatique des informations contenues dans les codes-barres et les notes. Cette étape repose sur des algorithmes de reconnaissance d'image et d'intelligence artificielle capables d'identifier les codes-barres, de lire les notes manuscrites et de détecter d'éventuelles anomalies.

Le système vérifie la cohérence des données extraites avec celles déjà présentes dans la base de données, garantissant que chaque copie est correctement associée à l'étudiant, au module et à l'examen correspondant. Les erreurs de saisie ou de lecture sont automatiquement signalées ou corrigées lorsque cela est possible, ce qui augmente la fiabilité globale du processus. Cette

Chapitre II:Analyse et spécifications du système

automatisation réduit considérablement les risques d'erreurs humaines et accélère le traitement des examens tout en maintenant un niveau élevé de sécurité et de traçabilité des informations.

II.4.3 Post-examen et délibération

Après la tenue des examens, la phase post-examen constitue une étape cruciale du système, car elle permet de consolider, vérifier et diffuser les résultats de manière fiable et sécurisée. Cette étape ne se limite pas à l'enregistrement des notes ; elle englobe également le calcul des moyennes, le contrôle de la cohérence des données et la transmission des informations aux parties concernées tout en garantissant la confidentialité et l'intégrité du processus d'évaluation. L'ensemble de ces opérations vise à offrir un suivi académique précis et transparent, à prévenir les erreurs humaines et à sécuriser toutes les informations relatives aux performances des étudiants.

- Enregistrement des notes

Après la numérisation des copies et la lecture des codes-barres, le système procède à l'enregistrement des notes directement dans la base de données en les associant à chaque code-barre unique. Chaque code-barre agit comme un identifiant sécurisé et spécifique pour une copie donnée, garantissant que la note enregistrée correspond exactement à l'étudiant et au module concerné.

Les notes sont insérées dans la colonne dédiée à chaque code-barre dans la table de la base de données. Cette méthode assure que toutes les informations sont centralisées, facilement traçables et reliées de manière fiable à chaque copie. L'utilisation du code-barre comme clé permet de vérifier automatiquement l'intégrité des données, d'éviter les doublons et de prévenir toute confusion entre les copies. Ainsi, chaque note enregistrée est directement associée à son code-barre, ce qui renforce la précision et la sécurité du processus de gestion des résultats.

Cette approche facilite également la mise à jour automatique des notes en cas de correction ou de modification, car le code-barre unique garantit que toute modification se fait sur la copie correcte sans risque d'altérer les données des autres étudiants.

- Diffusion des résultats à l'enseignant par courriel

Après l'enregistrement final des notes dans la base de données, le système assure la diffusion sécurisée des résultats en les envoyant exclusivement au compte Gmail de l'enseignant

Chapitre II:Analyse et spécifications du système

responsable de la session d'examen. Cette approche garantit que toutes les informations restent confidentielles et centralisées, évitant toute diffusion accidentelle ou non autorisée à des tiers.

Le processus automatisé crée un rapport regroupant toutes les notes enregistrées pour chaque module et chaque étudiant, puis l'envoie directement à l'adresse de l'enseignant.

- Calcul des moyennes

Parallèlement à l'enregistrement et à la diffusion des résultats, le système calcule automatiquement les moyennes de chaque étudiant pour chaque module. Ce calcul prend en compte à la fois les notes des contrôles continus déjà présentes dans la base et les notes des examens nouvellement enregistrées via les codes-barres.

L'automatisation du calcul des moyennes permet d'obtenir rapidement la moyenne finale de chaque étudiant tout en garantissant l'exactitude et la cohérence des données. Le système effectue également des vérifications pour détecter d'éventuelles anomalies ou incohérences dans les notes et les moyennes, assurant ainsi que les performances académiques reflètent fidèlement le travail et les résultats des étudiants.

II.5 Spécifications fonctionnelles

ANONYMOS OS assure la lecture automatique des codes-barres à partir des images capturées par le module ESP32-CAM S3. Les images sont ensuite analysées afin de détecter et de décoder les symboles visuels correspondant aux codes-barres.

Une fois le code-barre identifié, le système établit la correspondance avec l'étudiant concerné à partir des informations stockées dans la base de données. Les notes associées aux copies corrigées sont ensuite enregistrées de manière structurée dans le système central.

La transmission des données entre le module d'acquisition et le système central s'effectue via une connexion réseau, garantissant la continuité du flux d'informations. Les données reçues sont stockées dans une base de données sécurisée, permettant leur consultation et leur exploitation ultérieure.

Le système réalise également plusieurs opérations automatiques, telles que le calcul des moyennes, la génération de statistiques et l'affichage des résultats pour les étudiants et le

Chapitre II:Analyse et spécifications du système

personnel administratif. L'ensemble de ces fonctionnalités permet d'assurer une gestion plus efficace et cohérente des examens [\[28\]\[44\]\[19\]](#).

II.6 Spécifications non fonctionnelles

Les spécifications non fonctionnelles définissent les caractéristiques de qualité, de performance et de fiabilité du système. La fiabilité constitue un élément essentiel, car ANONYMOS OS traite des données académiques sensibles et doit garantir l'exactitude des informations enregistrées. Chaque étape du traitement, depuis la capture des images jusqu'à l'enregistrement des notes dans la base de données, doit fonctionner de manière précise et cohérente.

La précision du processus de décodage des codes-barres garantit l'intégrité des informations associées aux étudiants et contribue à assurer la fiabilité des résultats académiques.

Le système présente également un coût relativement faible, grâce à l'utilisation de composants matériels accessibles tels que le module ESP32-CAM S3. L'automatisation du traitement permet en outre de réduire considérablement le temps nécessaire à la gestion des examens et à la publication des résultats.

Enfin, l'interface et le fonctionnement du système sont conçus pour rester simples et intuitifs, afin de permettre leur utilisation par le personnel administratif et les enseignants sans nécessiter de compétences techniques avancées, ce qui facilite son adoption dans l'environnement universitaire [\[28\]\[19\]\[16\]](#).

II.7 Description globale du système

ANONYMOS OS se compose de plusieurs composants matériels et logiciels qui interagissent afin d'assurer le traitement complet des examens. Les principaux acteurs impliqués dans ce processus sont les étudiants, les enseignants, le personnel administratif et le système central chargé du traitement automatisé des données

Les principaux acteurs impliqués dans ce processus sont les étudiants, les enseignants, le personnel administratif ainsi que le système central chargé du traitement automatisé des données.

Chapitre II:Analyse et spécifications du système

Le fonctionnement du système débute par la génération des codes-barres attribués à chaque étudiant et à chaque module d'enseignement. Ces codes sont ensuite imprimés en plusieurs exemplaires afin d'assurer la disponibilité de copies supplémentaires si nécessaire.

Après la correction des copies d'examen, le module ESP32-CAM S3 capture les images des codes-barres présents sur les copies. Les images sont ensuite transmises vers le système central, où elles sont analysées et décodées. Les informations obtenues permettent d'associer les notes aux étudiants correspondants et de les enregistrer dans une base de données centralisée.

Les données stockées peuvent ensuite être consultées et exploitées pour la préparation des délibérations ainsi que pour l'affichage des résultats. L'ensemble de ce flux d'informations permet d'assurer une gestion plus rapide, plus fiable et plus transparente des examens.

II.8 Conclusion

Nous avons présenté l'analyse du système ANONYMOS OS et les différentes spécifications requises pour son bon fonctionnement. L'étude de la problématique a permis de mettre en évidence les limites des méthodes traditionnelles utilisées dans la gestion des examens, notamment en termes de fiabilité, de rapidité de traitement et de risque d'erreurs humaines. L'étude de la problématique a permis de mettre en évidence les limites des méthodes traditionnelles utilisées dans la gestion des examens, notamment en termes de fiabilité, de rapidité de traitement et de risque d'erreurs humaines.

Pour répondre à ces contraintes, une solution fondée sur l'automatisation du processus de gestion des examens a été proposée. Cette solution innovante repose sur l'utilisation d'une base de données centralisée regroupant les informations relatives aux étudiants, aux modules, aux enseignants, aux salles d'examen et à l'emploi du temps des épreuves. À partir de ces informations, le système génère des codes-barres uniques permettant de garantir l'identification et l'anonymat des copies d'examen.

Le chapitre a également présenté les principales fonctionnalités du système, telles que la génération des codes-barres, la capture des images à l'aide du module ESP32-CAM S3, le décodage des informations et l'enregistrement des notes dans la base de données. Les spécifications fonctionnelles et non fonctionnelles ont permis de définir les capacités du système ainsi que les critères de performance et de fiabilité attendus.

Chapitre II:Analyse et spécifications du système

Ainsi, l'analyse et les spécifications présentées dans ce chapitre constituent une base fondamentale pour la compréhension et la mise en œuvre du système proposé. Le chapitre suivant sera consacré à la conception et à la réalisation du système, en détaillant l'architecture, les outils utilisés et les différentes étapes de développement.

Chapitre III: Conception du système Anonymos OS

Chapitre III: Conception du système Anonymos OS

III.1 Introduction :

Après l'analyse détaillée du problème et l'identification des exigences fonctionnelles et non fonctionnelles présentées dans le chapitre précédent, il convient désormais d'aborder la phase de conception du système. Cette étape constitue un jalon fondamental dans le processus de développement, puisqu'elle permet de traduire les besoins exprimés en une architecture cohérente, capable de répondre efficacement aux objectifs fixés. La conception ne se limite pas à une simple représentation technique du système ; elle constitue une démarche méthodique visant à définir les choix architecturaux, technologiques et organisationnels qui encadreront l'implémentation de la solution. Elle permet ainsi d'établir une vision globale du fonctionnement de la plateforme, tout en garantissant la cohérence et l'intégration de ses différents composants.

Dans ce chapitre présent, nous présentons la conception complète de notre plateforme **ANONYMOS OS**, dédiée à la gestion sécurisée des examens universitaires. Cette solution intègre plusieurs fonctionnalités essentielles, notamment les fonctions d'anonymisation automatique des copies, de traçabilité des opérations, la gestion sécurisée des données ainsi que l'automatisation du processus de délibération.

L'architecture proposée repose sur une combinaison de composants matériels et logiciels interconnectés. La partie matérielle s'appuie sur un module ESP32-S3-CAM équipé d'une caméra OV3660, utilisé pour la capture et la numérisation des copies d'examen. La partie logicielle comprend un ensemble de scripts développés en Python destinés au traitement des données, à la génération sécurisée des codes-barres et à l'automatisation des différentes opérations de gestion. Ces composants communiquent à travers une API REST développée avec le framework Flask, assurant l'échange et la synchronisation des données entre les différents modules du système. L'ensemble des informations relatives aux étudiants, aux examens, aux modules d'enseignement et aux résultats est stocké et géré au sein d'une base de données SQLite, garantissant simplicité, fiabilité et rapidité d'accès aux données.

Afin de fournir une vision complète de la solution proposée, ce chapitre adopte une approche multidimensionnelle. Dans un premier temps, nous présenterons l'architecture générale du système ainsi que les interactions entre ses différentes composantes. Dans un second temps, nous détaillerons la conception matérielle, en décrivant les caractéristiques du module ESP32-S3-CAM, son alimentation électrique et son interface de capture d'images. Enfin, nous aborderons

Chapitre III: Conception du système Anonymos OS

la conception logicielle à travers la présentation de l'architecture applicative, du firmware embarqué sur l'ESP32 ainsi que des différents modules logiciels assurant le fonctionnement global de la plateforme.

Cette étude de conception constitue ainsi la base technique et méthodologique sur laquelle repose la réalisation de la solution proposée, tout en garantissant sa robustesse, son évolutivité et sa capacité à répondre aux exigences de modernisation et de sécurisation des processus d'évaluation universitaire.

III.2 Architecture globale du système :

III.2.1 Schema bloc du système :

L'architecture générale du système proposé repose sur une structure modulaire comprenant plusieurs éléments matériels et logiciels interconnectés. Cette approche permet d'organiser proprement le système et de répartir clairement les responsabilités entre les différents modules, ce qui facilite sa maintenance, son évolution et sa compréhension globale.

L'objectif principal de cette architecture est d'assurer une automatisation complète du processus de gestion des examens, incluant la création d'identifiants anonymes, l'enregistrement et l'affichage des notes, ainsi que la capture et le traitement des copies d'examen.

Le système est organisé en plusieurs blocs fonctionnels, chacun jouant un rôle précis dans la chaîne de traitement.

- **Génération de codes-barres :**

Ce premier module constitue l'étape initiale du système. Il est développé en Python et permet de générer des identifiants uniques pour les étudiants sous forme de codes-barres.

Plutôt que de recourir à une génération aléatoire, ce module s'appuie sur une méthode sécurisée. Les données relatives à l'examen (identifiant, module, numéro de matricule, salle, enseignant, etc.) sont combinées puis traitées à l'aide d'un algorithme cryptographique construit autour de HMAC-SHA256. Le résultat obtenu est ensuite encodé en Base32 et raccourci afin de produire une chaîne compacte pouvant être utilisée comme valeur d'un code-barres.

Chapitre III: Conception du système Anonymos OS

Cette méthode garantit :

- L'unicité des codes générés.
- Leur caractère non prédictible.
- Une meilleure résistance contre les tentatives de falsification.

Les codes-barres sont ensuite générés sous forme d'images (format Code128), puis stockés directement dans la base de données SQLite sous forme de données binaires (BLOB).

- **Génération des fiches d'examen :**

Après la génération des codes-barres, un second module assure la création automatique de documents PDF contenant les informations des étudiants. Ce module, basé sur la bibliothèque ReportLab, exploite les données stockées dans la base de données et les organise par matière, enseignant et salle d'examen.

Pour chaque étudiant, le document comprend :

- Le nom et le prénom.
- Le numéro d'authentification.
- Plusieurs reproductions du code-barres créé.

La disposition des codes-barres sur une même ligne facilite leur utilisation lors des examens (collage sur les copies, listes de présence, etc ...).

Ces fiches constituent un élément essentiel du dispositif, car elles garantissent l'anonymat des copies en remplaçant l'identité de l'étudiant par un identifiant codé.

- **Capture des-copies :**

La phase suivante intervient après la correction des copies. Le système utilise un module embarqué ESP32-S3-CAM équipé d'une caméra OV3660 pour la capture des images des feuilles d'examen. Ce matériel a été choisi en raison de sa compacité, de son faible coût et de sa connectivité Wi-Fi. Il permet de numériser les copies contenant à la fois le code-barres d'identification et la note attribuée par l'enseignant. Dès la capture de l'image, celle-ci est transmise à un serveur intermédiaire au moyen d'une requête HTTP. Le rôle de l'ESP32-S3-CAM se limite ainsi à l'acquisition et à la transmission des données, sans effectuer de traitements complexes.

Chapitre III: Conception du système Anonymos OS

- **Serveur relais et API**

Le serveur intermédiaire assure la liaison entre le module embarqué et le système de traitement. Développé à l'aide du framework Flask, il expose une API REST permettant de recevoir et de gérer les images transmises.

Ce serveur joue plusieurs rôles :

- Réception des images envoyées par l'ESP32
- Gestion des requêtes réseau
- Transfert des données au module de traitement Python

Cette séparation des responsabilités permet de déporter les traitements intensifs vers le système central tout en conservant un module embarqué léger et performant.

- **Module de traitement et de reconnaissance (Python)**

Cet ensemble de méthodes d'intelligence artificielle implémentées pour la vision par ordinateur va analyser les images issues des copies d'examen et y extraire les informations dont nous avons besoin de façon entièrement automatisée. Il s'agit donc de transformer les images récupérées avec l'ESP32-S3-CAM en informations nécessaires pour pouvoir identifier l'étudiant et lui attribuer sa note automatiquement.

Ainsi, après le passage de l'image sur le serveur Flask, le module d'intelligence artificielle se lance dans la phase d'analyse visuelle consistant à repérer le code-barres contenu sur cette image (module PyZbar). Cette opération nous permet non seulement de repérer l'étudiant en question, mais aussi de respecter le principe d'anonymat qui guide notre système.

L'un des points forts de ce module est sa capacité à s'adapter aux différents paramètres de capture. En effet, on peut supposer que la feuille d'examen peut être mal orientée, légèrement tournée ou mal placée face à la caméra. Cette opération permet de redresser la feuille avant toute phase d'analyse et garantit ainsi la robustesse du traitement quelles que soient les conditions de prise de vue.

Lorsque l'image est normalisée, on applique plusieurs méthodes avancées de vision par ordinateur qui nous aident à identifier les informations utiles. On passe d'abord par l'étape de passage en niveaux de gris (Grayscale Conversion), puis au filtre pour supprimer tout bruit qui

Chapitre III: Conception du système Anonymos OS

pourrait s'y trouver (Gaussian Blur). Suite à cela, une analyse d'images est effectuée à travers les méthodes Canny Edge Detection et Contour Detection pour identifier automatiquement les structures géométriques des tableaux de notation.

Le processus se poursuit par l'étape d'analyse des tableaux de notation détectés. Les tableaux sont divisés en cellules correspondant aux différents critères de notation (Segmentation par contours et découpage matriciel). Une analyse de densité est ensuite faite dans chaque cellule avec comme objectif de déterminer automatiquement le point d'intersection du tableau (Pixel Density Analysis). En utilisant cette méthode, le système est capable de reproduire la logique utilisée par l'humain lorsqu'il lit une grille de notation.

A partir informations identifiées, il peut alors interpréter les valeurs des deux parties, soit celle de la partie entière et celle de la partie décimale, de la note attribuée (Reconnaissance des cellules cochées). Ensuite, il est possible de fusionner ces deux valeurs pour obtenir une note automatique de l'étudiant (Fusion des valeurs et calcul automatique). Le système effectue alors un lien intelligent entre la note et l'étudiant associé grâce au code-barres détecté et enregistre immédiatement cette information dans la base de données.

Il en résulte donc que ce module d'intelligence artificielle est capable d'assurer en même temps plusieurs tâches liées à la perception (PyZbar), l'analyse (OpenCV), la reconnaissance (Contour Detection et Pixel Density Analysis), le traitement (Rotation géométrique), la prise de décision (Sélection de la cellule ayant la densité de pixels la plus élevée) et l'interprétation des informations (Calcul automatique de la note).

Chapitre III: Conception du système Anonymos OS

- **Base de données SQLITE :**

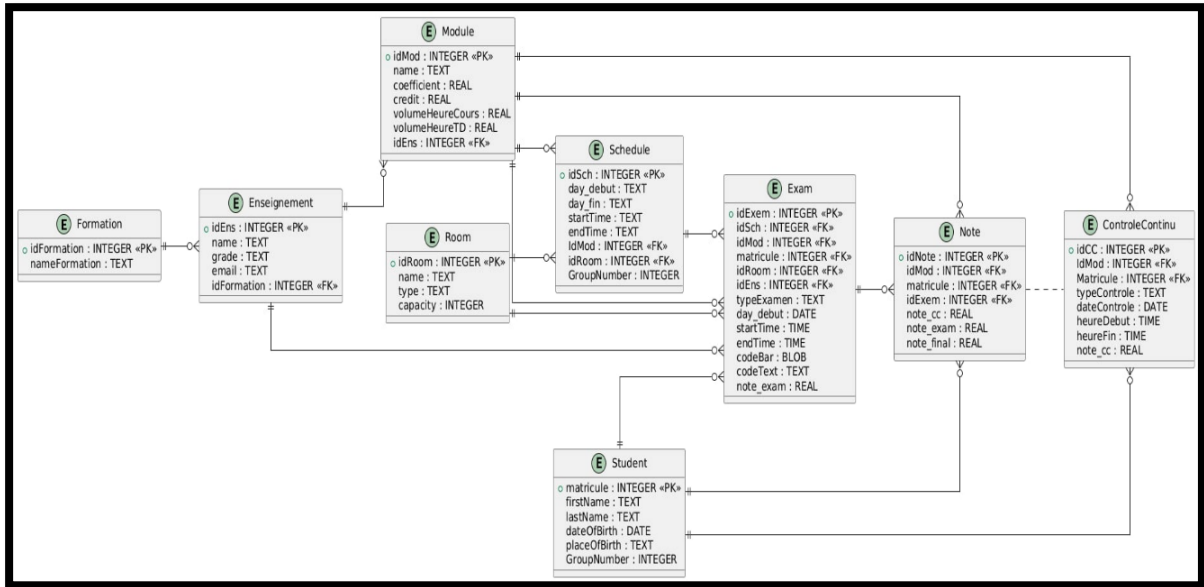


Figure III-1.UML DATABASE

La base de données constitue le dispositif de stockage central du système. Elle contient l'ensemble des informations nécessaires, notamment :

- Les étudiants.
- Les modules.
- Les examens.
- Les codes-barres.
- Les notes, etc.

Une fois la note calculée, le système met automatiquement à jour la base de données en associant cette note à l'étudiant concerné au moyen d'une requête SQL de type UPDATE.

- **Interface d'affichage :**

Afin de centraliser l'ensemble des fonctionnalités du système, une interface utilisateur interactive a été développée à l'aide de la bibliothèque Streamlit de Python. Cette interface constitue le point d'accès principal pour les utilisateurs (enseignants ou personnel administratif), leur permettant d'effectuer toutes les opérations nécessaires à partir d'un environnement unique. Les principaux besoins du système sont de gérer le processus d'évaluation par :

- La génération des fiches PDF contenant les codes-barres ;

Chapitre III: Conception du système Anonymos OS

- Le lancement du processus de décodage des copies capturées ;
- L'extraction automatique des notes ;
- L'exécution de la délibération automatisée ;
- La consultation des résultats obtenus.

Cette interface vise à simplifier l'utilisation du système et à réduire les interventions techniques liées au code ou à la gestion directe de la base de données. L'utilisateur peut ainsi gérer l'ensemble du processus de manière autonome à travers une interface unique.

Cette centralisation rend le système plus ergonomique, plus facile à exploiter au sein de l'établissement et mieux adapté à une utilisation en conditions réelles.

Après le traitement des copies, l'extraction des notes et la finalisation de la délibération, le système procède automatiquement à l'envoi des résultats par courrier électronique à l'enseignant responsable de l'examen. La réception directe des résultats par courrier électronique évite toute intervention intermédiaire dans la transmission des notes des étudiants.

Cette étape de transmission offre divers avantages :

- La rapidité de transmission des résultats.
- La réduction des tâches administratives liées à la communication des notes.
- L'automatisation complète du processus d'évaluation.
- La meilleure traçabilité des résultats transmis.

III.2.2 Interactions entre les différents modules :

Le bon fonctionnement du système repose sur une interaction continue entre les différents modules, chacun ayant son rôle à jouer dans la chaîne de traitement.

Dans un premier temps, après avoir corrigé les copies, le module ESP32-S3-CAM prend une image de la feuille d'examen. Ce cliché contient le code-barres et la note mise par l'enseignant.

Cette image est ensuite transmise au serveur intermédiaire via une requête HTTP. Celui-ci joue le rôle de point d'entrée du système en réceptionnant les données avant de les acheminer vers le module de traitement.

Chapitre III: Conception du système Anonymos OS

Le module de traitement développé en Python analyse ensuite l'image reçue. Il procède d'abord à l'extraction du code-barres afin d'identifier l'étudiant concerné, puis applique les différentes techniques de traitement d'images permettant de détecter et d'interpréter automatiquement la note.

Une fois ces informations extraites, le système accède à la base de données SQLite afin de localiser l'enregistrement correspondant à l'étudiant et d'y enregistrer la note obtenue.

Enfin, l'interface utilisateur permet de consulter en temps réel les données mises à jour et de les exploiter dans le cadre du processus de délibération.

Cette interaction entre les différents modules garantit l'automatisation complète du processus de traitement des copies, tout en réduisant les erreurs associées à la saisie manuelle des notes. Elle assure également une traçabilité intégrale des opérations, chaque étape étant associée à un identifiant unique représenté sous forme de code-barres.

III.3 Conception Matérielle :

III.3.1 Schema de L'ESP32-S3-CAM :

Le système proposé repose sur le module ESP32-S3-CAM, équipé d'un capteur d'image OV3660 et d'une connectivité Wi-Fi intégrée. Cette configuration lui permet de capturer des images et de les transmettre directement vers un serveur distant sans nécessiter de matériel supplémentaire. Il constitue ainsi une solution embarquée compacte, autonome et économique.

Dans l'architecture retenue, le rôle de l'ESP32-S3-CAM a volontairement été limité afin de simplifier son fonctionnement et d'optimiser ses performances. Ses principales tâches consistent à capturer l'image de la copie d'examen contenant le code-barres et la note manuscrite, puis à transmettre cette image au serveur à travers le réseau Wi-Fi. Aucun traitement d'image n'est réalisé localement sur le module embarqué. Ce choix architectural permet d'éviter la surcharge du microcontrôleur en déléguant les opérations de traitement complexes à une plateforme logicielle plus performante, exécutée sur une machine centrale.

Le fonctionnement du module repose sur une séquence d'opérations simple et efficace : après l'initialisation de la caméra, une image est capturée à la demande, encodée au format JPEG, puis transmise au serveur au moyen d'une requête HTTP. Bien que ces opérations soient relativement

Chapitre III: Conception du système Anonymos OS

simples, cette architecture présente plusieurs avantages, notamment une grande rapidité d'exécution, une bonne fiabilité et une intégration aisée au sein du système global.

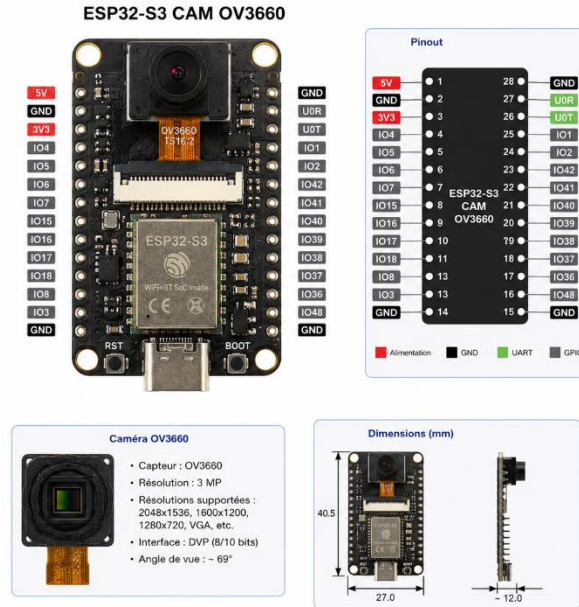


Figure III-2 ESP32-S3-CAM OV 3660

III.3.2 Alimentation :

L'alimentation du module ESP32-S3-CAM constitue un élément essentiel pour garantir le bon fonctionnement et la stabilité du système. En effet, l'utilisation simultanée de la caméra et de la connectivité Wi-Fi entraîne une consommation énergétique relativement importante, notamment lors de la capture et de la transmission des images.

Le module est généralement alimenté sous une tension de 5V. Dans notre système, cette alimentation peut être fournie soit par un adaptateur externe, soit par un convertisseur USB-série de type FTDI, également utilisé lors des phases de programmation.

Il convient de souligner que l'ESP32-S3-CAM est particulièrement sensible aux variations de tension. Une alimentation instable ou insuffisante peut provoquer divers dysfonctionnements tels que des redémarrages intempestifs, des erreurs lors de la capture des images ou encore des pertes de connexion au réseau Wi-Fi. Il est donc recommandé d'utiliser une source d'alimentation stable et correctement régulée afin de garantir un fonctionnement fiable et continu du système. Cette

Chapitre III: Conception du système Anonymos OS

exigence est particulièrement importante dans un contexte d'exploitation réelle, où le système doit assurer un service permanent et sans interruption.

III.3.3 Interface camera :

La camera intégrée dans le module ESP32-S3-CAM occupe une place centrale dans le fonctionnement du système, car elle assure la numérisation des copies d'examen.

Le capteur utilisé est de type OV3660. Il offre une résolution suffisante pour capturer avec précision les éléments essentiels présents sur la copie, notamment le code-barres d'identification et la grille de notation contenant la note manuscrite. La qualité des images acquises constitue un facteur déterminant pour assurer l'efficacité des étapes ultérieures de traitement et de reconnaissance.

Pour ce faire, un certain nombre de paramètres peuvent être ajustés notamment la résolution, la luminosité ou le contraste. Le réglage adéquat de ces paramètres améliore la lisibilité des informations contenues dans l'image et favorise ainsi la fiabilité des traitements réalisés par le module de reconnaissance.

Dans l'architecture proposée, la caméra est utilisée selon un mode de fonctionnement simple et ciblé. Le système ne repose pas sur un flux vidéo continu ; il procède plutôt à la capture d'images fixes à des instants précis. Chaque image capturée est immédiatement transmise au serveur afin d'être analysée par le module de traitement développé en Python.

Cette approche permet de réduire la charge de calcul du module embarqué tout en garantissant une qualité d'image suffisante pour l'extraction fiable des informations contenues dans les copies d'examen.

III.4 Conception logicielle :

III.4.1 Architecture du frimwar embarqué :

Le firmware embarqué correspond à l'ensemble des instructions exécutées directement sur le module ESP32-S3-CAM, lui permettant d'assurer l'interface entre les composants matériels (caméra et connectivité Wi-Fi) et le système logiciel global. Dans le cadre de notre système, une approche minimaliste a été adoptée pour la conception du firmware, en raison des ressources limitées du module ESP32-S3-CAM, notamment en termes de mémoire et de puissance de calcul.

Chapitre III: Conception du système Anonymos OS

Ainsi, seules les fonctions indispensables, à savoir l'acquisition et la transmission des images, sont exécutées sur le module embarqué, tandis que l'ensemble des traitements complexes est délégué au système central.

- **Initialisation du système :**

Lors du démarrage, le firmware procède à l'initialisation de l'ensemble des composants matériels :

- Configuration des broches de la caméra ;
- Initialisation du capteur OV3660 ;
- Configuration de la mémoire tampon pour le stockage temporaire des images ;
- Initialisation de la connexion Wi-Fi sur la base des paramètres réseau (SSID et mot de passe).

Une fois connecté au réseau, il est en mesure de communiquer avec le serveur distant.

- **Configuration de la caméra :**

La configuration de la caméra constitue une étape essentielle, car elle influence directement la qualité des images acquises. Les principaux paramètres configurés sont:

- La résolution (choisie dans un compromis qui garantit à la fois la qualité et la taille du fichier).
- Le format d'encodage (JPEG pour réduire la taille des données).
- Certains réglages internes comme la luminosité et le contraste.

Une résolution trop élevée augmenterait considérablement le temps de transmission, tandis qu'une résolution insuffisante pourrait compromettre la lisibilité du code-barres et des grilles de notation.

- **L'acquisition de l'image :**

Le module déclenche la capture à la demande en prenant une photographie de la copie d'examen contenant simultanément le code-barres d'identification et la grille de notation renseignée par l'enseignant. L'image obtenue est ensuite stockée temporairement dans la mémoire interne du module.

- **L'encodage et la préparation :**

Chapitre III: Conception du système Anonymos OS

L'image capturée est compressée et encodée au format JPEG afin d'accélérer sa transmission. Cette compression permet une réduction significative de la taille des données tout en préservant une qualité suffisante pour les traitements ultérieurs. Les données sont ensuite encapsulées dans une requête HTTP, généralement sous la forme d'un flux binaire.

- **Transmission vers le serveur :**

Le module transmet l'image au serveur à travers une requête HTTP de type POST adressée à l'API Flask (par exemple : /api/scan). La requête contient l'image capturée ainsi que d'éventuelles métadonnées telles que l'horodatage ou l'identifiant du dispositif. Après réception, le serveur prend en charge l'ensemble des opérations de traitement.

- **Choix d'architecture :**

L'absence de traitement d'image au niveau de l'ESP32-S3-CAM s'explique par plusieurs facteurs :

- Une contrainte matérielle
- La complexité des algorithmes de traitement d'image
- Un choix d'architecture « logique » qui centralise le traitement du côté serveur.

L'ESP32-S3-CAM est donc réduit à une sorte de module d'acquisition « intelligent », garantissant la sortie du monde physique (copies d'examen) et l'environnement logiciel de traitement.

III.4.2 Architecture logicielle coté pc :

L'architecture logicielle côté PC constitue le cœur du système proposé. C'est à ce niveau que sont réalisées les principales opérations, notamment la génération des identifiants anonymes, le traitement des images des copies d'examen, l'extraction automatique des notes ainsi que la gestion et la visualisation des données.

Le langage Python a été retenu en raison de sa simplicité, de sa flexibilité et de la richesse de son écosystème logiciel, particulièrement adapté au traitement d'images, à la manipulation de données et au développement d'interfaces utilisateur. Afin d'assurer une meilleure organisation du système et de faciliter sa maintenance, l'architecture logicielle a été structurée en plusieurs modules indépendants mais interconnectés, chacun remplissant une fonction spécifique au sein de la chaîne de traitement.

Chapitre III: Conception du système Anonymos OS

- **Module de génération des codes-barres sécurisés :**

Ce module intervient durant la phase préparatoire aux examens. Son objectif est de générer des identifiants anonymes sous forme de codes-barres afin de dissocier l'identité de l'étudiant de sa copie d'examen. Le processus de génération repose sur une approche sécurisée et déterministe implémentée à l'aide des bibliothèques Python hashlib, hmac et base64. Dans un premier temps, les informations nécessaires sont extraites de la base de données SQLite.

Ces informations comprennent notamment :

- L'identifiant de l'examen
- Le module
- Le numéro d'inscription de l'étudiant
- La salle d'examen.
- l'enseignant.

Ces données sont ensuite concaténées sous la forme d'une chaîne structurée constituant le message d'entrée. Une signature cryptographique est générée à l'aide de l'algorithme HMAC associé à la fonction de hachage SHA-256 et à une clé secrète prédéfinie. Cette méthode garantit l'unicité des codes produits, leur caractère imprévisible ainsi qu'une forte résistance aux tentatives de falsification. Le résultat obtenu est ensuite encodé en Base32 puis tronqué afin de produire une chaîne compacte de 17 caractères, adaptée à une représentation sous forme de code-barres. La bibliothèque python-barcode est ensuite utilisée pour générer les images au format Code128. Ces images sont stockées dans la base de données sous forme de données binaires (BLOB), ce qui permet leur réutilisation lors de la génération des documents PDF.

- **Module de génération des documents PDF :**

Ce module propose une production automatisée des fiches d'examen à partir des données enregistrées dans la base.

Développé à l'aide de la bibliothèque ReportLab, il permet de générer des documents structurés regroupant les étudiants selon les modules, les enseignants et les salles d'examen.

Pour chaque étudiant, le document généré contient :

- Le nom et le prénom.

Chapitre III: Conception du système Anonymos OS

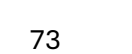
- Le matricule.
- Signature.
- Plusieurs copies du code-barres associé.

La présence de plusieurs codes-barres en ligne répond à des problématiques pratiques d'usage durant l'examen (pour le collage en présence du code-barre sur la copie, la fiche de présence, etc.).

- Le module gère également :
- La pagination automatique ;
- La mise en forme des tableaux ;
- L'alignement des éléments ;
- L'adaptation des tailles d'image.

Les documents générés seront ensuite imprimés et distribués aux étudiants.

EXAMEN : Traitement du Signal **ENSEIGNEMENT** : Dr Ahmed Benali
SALLE : Amphi A **DATE** : 2026-05-06 à 08:00 - 10:00

Etudiant	Matricule	Signe	Signe	Barcode	Barcode	Barcode
AMARA NAFISSA	212137566204					
BARACILE ABIR	202137747905					
BELBOUCHIHA Rachid	212138264701					
BELGACHEM Sid Ahmed	212137594419					
BELMAROUF Abdelhalim	212138510719					
Ben ALLOU ABDELILAH	212137423814					
BEN BOUBEKEUR EL habib	202138332511					
BENARIBI Houria	212137299512					
BENBAKHETI Ibrahim	212138250509					
BENHIBA Selma	212137696014					
BENMANSOUR Younnes	212137212702					
BENSLIMAN Morad	212137447413					
BERKANE Rahma	212138363811					
BOUALAME Ramzy	212137548609					
BOUAZZA Fatima	212137439315					

Chapitre III: Conception du système Anonymos OS

- **Module de traitement d'image et de reconnaissance des notes :**

C'est le cœur du système. Sa tâche est simple – copier le fonctionnement de l'humain lorsqu'il lit la copie de l'examen, mais complètement mécanisé. Comment? À travers les algorithmes utilisés dans l'intelligence artificielle et l'ordinateur-vision. Ces technologies aident l'application à analyser les photos des copies, reconnaître les étudiants, obtenir leurs informations et attribuer leurs notes. Tout cela se fait par plusieurs étapes comme suit :

a) Acquisition et validation intelligente de l'image

Cela débute avec la réception de l'image transmise par le module ESP32-S3-CAM. L'image est ensuite importée dans le programme par la bibliothèque OpenCV et passée directement au processus de vérification automatisée de la qualité de l'image. Toute image floue, faiblement cadrée ou sans valeur d'analyse est écartée du processus dès la première étape.

b) Détection intelligente du code-barres et identification de l'étudiant

À ce stade, PyZbar, une bibliothèque Python, effectue l'analyse automatisée et la reconnaissance des informations contenues dans le code-barres gravé sur la copie. Les informations stockées dans le code-barres portent l'identifiant anonyme de l'étudiant afin d'effectuer sa recherche dans la base de données.

Un autre problème technique qui se pose dans le processus de traitement de la copie est résolu ici également. C'est celui des copies non horizontalement orientées. Une fonction de correction géométrique effectue alors la rotation des copies pour faciliter leur analyse.

c) Prétraitement intelligent de l'image

Toute information de notation étant extraite, l'image est préparée par une série de traitements qui lui permettent une meilleure visualisation avant le processus de lecture. Cela passe par une conversion de l'image en niveaux de gris afin de faciliter le processus, une application de filtre gaussien pour minimiser la présence de bruit, le repérage des contours pour la mise en évidence des caractéristiques et enfin des opérations morphologiques pour optimiser la lecture.

d) Détection intelligente des régions de notation

Chapitre III: Conception du système Anonymos OS

La plateforme se met ensuite à analyser l'architecture générale de la feuille afin de repérer automatiquement les diagrammes utilisés pour faire l'évaluation. Cela implique l'utilisation de la méthode de repérage de contours et de formes précédemment employée afin d'éliminer les parties superflues de l'image et classer celles qui restent en fonction de leur taille, forme et emplacement. La conclusion de ce processus est qu'il devient possible au système de reconnaître correctement le diagramme représentant la valeur totale de la note ainsi que celui représentant sa partie décimale.

e) Reconnaissance des cases cochées et segmentation intelligente

Chaque tableau identifié est par la suite analysé cellule après cellule où chaque cellule représente une notation possible. Afin d'éviter toute lecture incorrecte causée par la ligne de division ou une certaine imperfection de la feuille de notes, le programme examine uniquement la partie centrale de chaque cellule. Ensuite, le programme calcule la densité de pixels de chaque cellule, ce qui signifie le taux d'occupation visuelle, et, comparant toutes les cellules entre elles, identifie celle qui a été marquée par l'enseignant.

Sur la feuille se trouve également une zone « Réservé à l'administration » qui permet de numériquement attribuer la copie à l'étudiant et au module correspondant.

f) Analyse intelligente et calcul automatique de la note

Après avoir interprété les cases cochées, le système va déterminer deux informations, soit la partie entière et la partie décimale de la note qui seront combinées pour obtenir la note définitive. L'affectation de cette note sera effectuée de manière automatique grâce à l'identification de l'élève via le numéro dans le code-barre lu au début de ce processus.

Dernier point : Grâce au code identifiant fourni par le déchiffrement de l'information codée des barres, le module sert automatiquement à attribuer la note obtenue à l'étudiant et stocke les résultats dans une base de données.

Par conséquent, ce module d'intelligence artificielle exerce de manière simultanée les fonctions de perception visuelle, d'analyse, de reconnaissance, de correction géométrique, d'interprétation des données et de décision, permettant ainsi de procéder totalement à l'automatisation de la tâche de lecture des copies d'examen.

Chapitre III: Conception du système Anonymos OS

- **Module de gestion de la base de données :**

La base de données SQLite a pour but le stockage et la gestion des données.

Le système interroge la base à l'aide de requêtes SQL, essentiellement :

- SELECT pour la récupération des données
- UPDATE pour l'enregistrement des notes

La mise à jour est automatique, en associant la note calculée au code-barres décodé.

- **Module d'interface utilisateur :**

Afin d'assurer la gestion et le pilotage de l'ensemble du système, une interface graphique a été développée. Cette interface constitue le point central d'interaction entre l'utilisateur et les différents modules logiciels intégrés à la solution. Son rôle ne se limite pas à l'affichage des

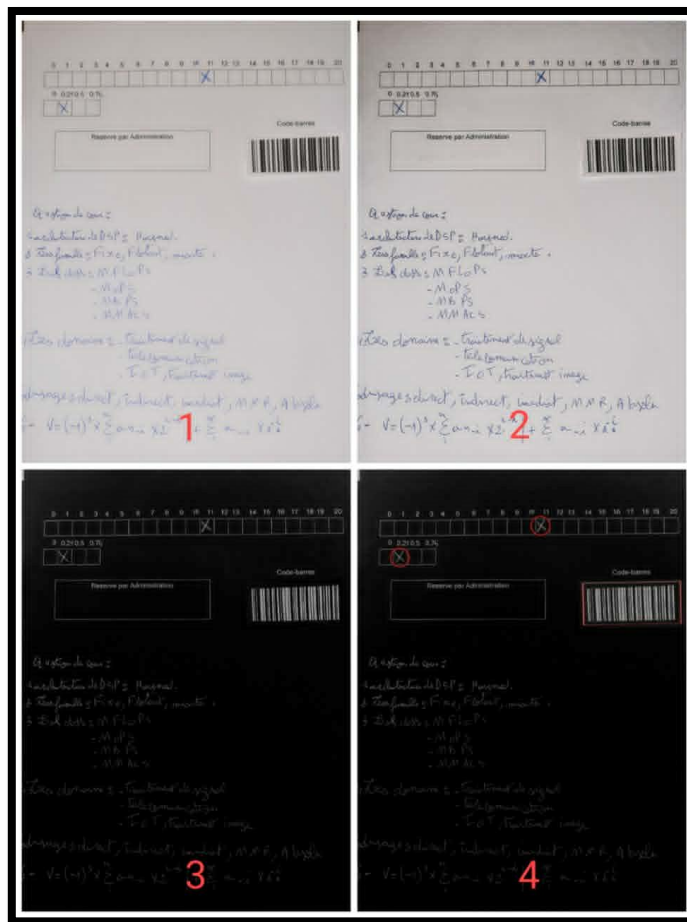


Figure III-4 Chaîne de traitement appliquée aux images des copies d'examen pour la détection des zones de notation et l'identification automatique des notes.

Chapitre III: Conception du système Anonymos OS

données ; elle permet également de contrôler les différents traitements, de suivre leur exécution et de consulter les résultats générés par le système.

L'interface a été développée à l'aide du framework Streamlit, une technologie permettant de concevoir rapidement des applications web interactives en Python. Ce choix a été motivé par sa simplicité de développement, sa rapidité de déploiement ainsi que sa capacité à intégrer facilement des tableaux de bord interactifs capables de communiquer efficacement avec les différents composants de l'architecture logicielle.

L'objectif principal était de fournir une interface ergonomique, intuitive et facile à utiliser, y compris pour des utilisateurs ne disposant pas de compétences techniques avancées.

L'accès à la plateforme débute par une procédure d'authentification sécurisée, constituant la première couche de sécurité du système. Avant d'accéder au tableau de bord principal, l'utilisateur doit renseigner son User ID ainsi que sa clé d'accès. Cette étape permet de garantir que seules les personnes autorisées peuvent accéder à la plateforme et aux données qu'elle contient.

Cette procédure permettra :

- De limiter l'accès au système aux seuls utilisateurs autorisés ;
- De protéger les informations relatives aux étudiants ;
- De garantir la confidentialité des données traitées ;
- D'assurer la sécurité des opérations de délibération et de publication des résultats.

Une fois l'authentification validée, l'utilisateur accède à l'interface principale de gestion, à partir de laquelle il peut piloter l'ensemble des fonctionnalités du système.

Ainsi, le mécanisme d'authentification constitue la première ligne de défense de l'architecture logicielle, contribuant à la sécurisation des données sensibles et à la protection des processus critiques liés à la gestion des examens.

Chapitre III: Conception du système Anonymos OS

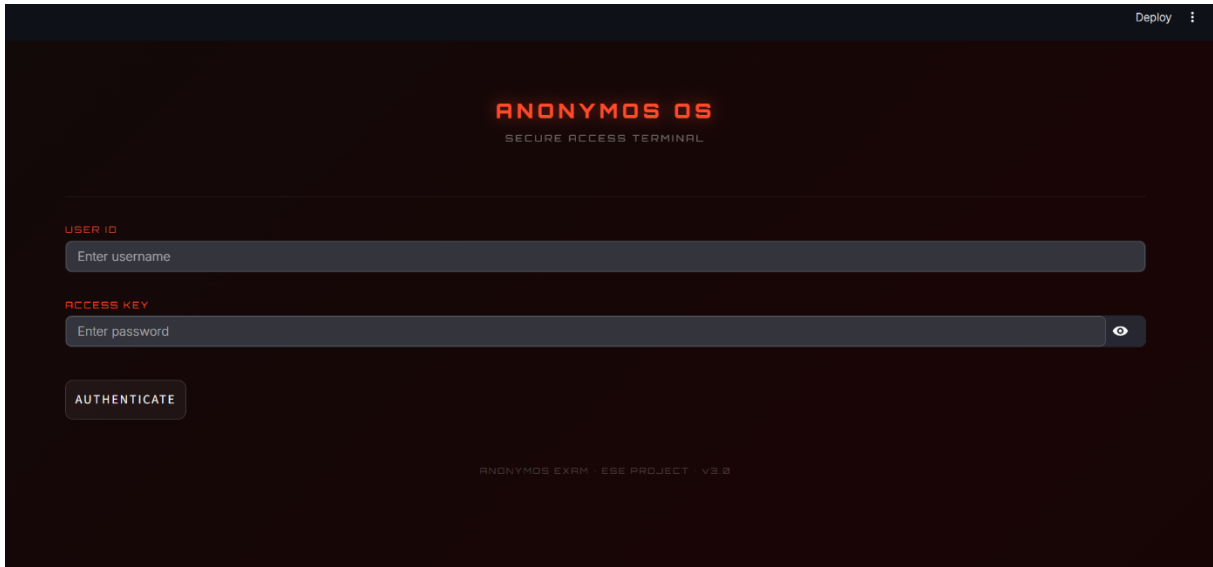


Figure III-5: Interface de sécurité du système.

Après l'authentification, l'utilisateur accède à l'interface de gestion du système. Cette interface a été conçue en tant que tableau de bord central qui aide à piloter toutes les opérations depuis un seul lieu.

Le Dashboard est composé de deux parties essentielles : une barre latérale de navigation à gauche et une zone de contrôle à droite. Le premier élément constitue le menu général de l'interface qui est constant tout au long de la navigation. Celle-ci comprend le logo de la plateforme, son état général, l'accès aux divers modules, le bouton de déconnexion et les informations générales concernant le projet. La barre latérale est toujours visible pour permettre une navigation facile entre les différentes pages de l'application.

La zone centrale est le noyau de l'application qui comprend tous les modules liés aux étapes du traitement du système. Chaque module contient des boutons de lancement, des commandes ainsi qu'une console de terminal qui permet de voir les messages de réponse du système. L'avantage de cette interface réside dans sa structure qui est très bien organisée.

Le premier module concerne la génération des codes-barres. Ce module permet de générer automatiquement les codes-barres relatifs aux copies ou identifiants concernés. Dès que l'utilisateur choisit l'option Barcode, une requête est immédiatement envoyée au backend Python. L'application recueille ensuite les informations nécessaires, qui sont celles relatives aux

Chapitre III: Conception du système Anonymos OS

identifiants concernés, avant de déclencher automatiquement le processus de génération du code-barres. Les codes-barres sont générés dynamiquement et de façon temporaires avant d'être utilisés pour le reste du processus. Dans ce processus, le terminal intégré dans ce module affiche à temps réel la situation du processus : initiation du processus, extraction des informations, génération des codes-barres, validation finale et message de fin de processus.

Le deuxième module sert à la création des documents PDF. Cette section se situe habituellement après la génération des codes-barres. Si l'utilisateur appuie sur la commande PDF, alors le processus d'autorisation de la production du document débute automatiquement. Tous les éléments de données, tels que ceux recueillis de la base de données et les codes-barres déjà générés, seront intégrés dans un PDF généré dynamiquement par le processus. Pendant la procédure, toutes les étapes effectuées seront affichées via l'écran terminal du PDF, qui comprend le lancement de l'opération, l'importation des données, la génération du document, son enregistrement et enfin la validation.

Le module Decode fait figure de module important pour la totalité de l'application car il consiste à interpréter et analyser les données collectées ou obtenues. Un terminal sert dans le cas présent à fournir plusieurs informations liées au processus de traitement de l'information. Cela inclut les identifiants, les valeurs obtenues et même les informations issues de l'analyse des données entrées dans le système.

Le module de Délibération constitue une des phases les plus importantes du système, car il garantit l'automatisation de l'association entre chaque copie corrigée, son évaluation respective et l'étudiant concerné. Suite au module de Décode, qui fait l'extraction de données à partir de codes-barres inscrits sur les feuilles d'évaluation, les numéros de code sont utilisés pour identifier précisément chaque étudiant. Le système en extrait donc les informations décryptées pour pouvoir établir une relation précise entre code-barres, document analysé et étudiant.

Une fois que le code-barres de chaque étudiant a été identifié, l'évaluation calculée est automatiquement associée à l'étudiant correspondant. Ce processus permet d'enregistrer dans la base de données l'association entre le nom de l'étudiant et le score obtenu par la copie respective, ce qui permet d'avoir la note finale de chaque étudiant basée sur sa propre copie, sans aucune intervention humaine supplémentaire.

Chapitre III: Conception du système Anonymos OS

Il s'agit donc de ce module de délibération qui joue le rôle d'un point de validation final et de consolidation des informations obtenues. Ce dernier permet d'affirmer qu'une note est bien attribuée à la bonne personne lors de son inscription dans le système. Lorsque la délibération est réalisée, le traitement des résultats est effectif puisqu'ils sont ensuite accessibles à l'ensemble de l'application pour leur consultation via l'interface utilisateur, leur génération sous format PDF et leur enregistrement dans la base de données. Enfin, un module d'envoi de ces derniers se fait automatiquement auprès des étudiants.

Le dernier module est responsable de la transmission des notes/résultats produits par l'application. Il permet le transfert automatique des données après traitement. L'utilisateur lance le processus de transmission, consulte les fichiers PDF générés et suit l'état de la transmission via la fenêtre du terminal. Des messages indiquent immédiatement si l'opération a réussi.

Dernier point, un élément très important dans cette interface est celui de l'intégration des terminaux de retour d'exécution. Différent de l'interface traditionnelle constituée uniquement de boutons, l'interface que nous proposons permet un retour d'information constant sur la situation interne du système. L'action de chaque utilisateur donne directement lieu à une visualisation dans le terminal qui lui est associé. Cette approche offre une vue très précise sur l'état des actions en cours, la conformité de leurs exécutions, ainsi qu'un moyen efficace de détecter les anomalies éventuelles.

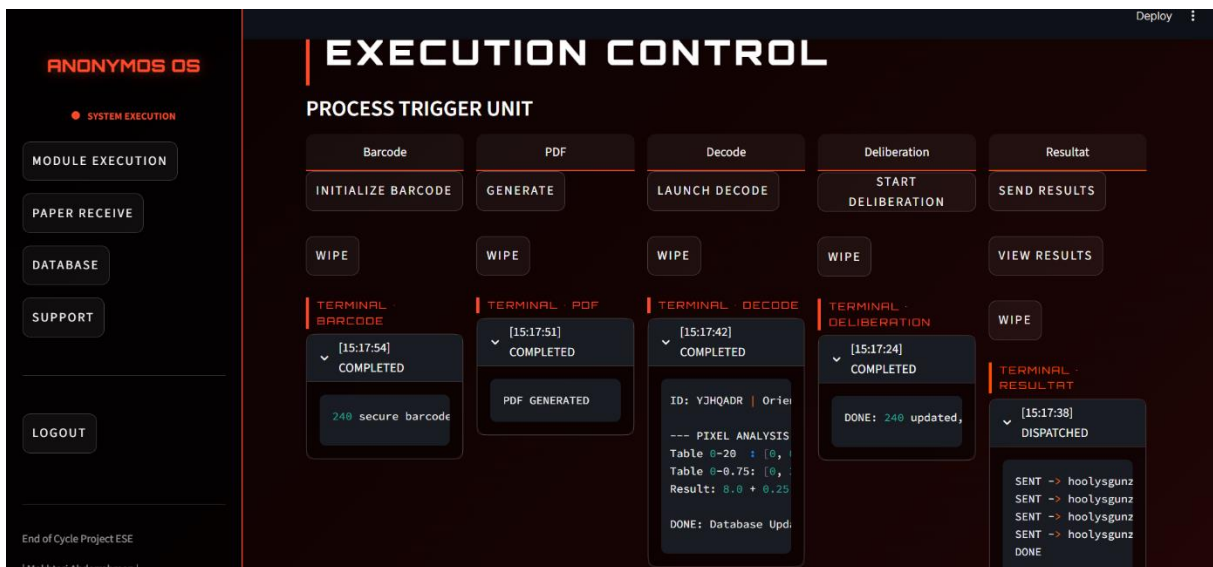


Figure III-6: Interface d'exécution et de contrôle du système.

Chapitre III: Conception du système Anonymos OS

L'interface intègre également un module de supervision du serveur, qui est destiné à surveiller les échanges entre le client et le serveur dans la perspective du traitement des informations et enregistrer les images de feuilles. Le module de surveillance offre des informations concernant l'arrivée des requêtes du client et la réception des fichiers d'images de feuilles. A chaque image transférée (l'image de la feuille) est automatiquement détectée par le serveur et il montrera une série de messages indiquant qu'il a reçu ce fichier, le stocke temporairement et commence à le traiter.

Parmi les principaux avantages de cette supervision figure le gain en termes de traçabilité des tâches. En effet, l'utilisateur sait exactement à quel moment l'image a été reçue, traitée et gérée par le système, améliorant ainsi la transparence de son fonctionnement. En outre, une telle surveillance contribue également à optimiser la stabilité du système en raison de sa capacité à repérer les dysfonctionnements réseau et les erreurs de gestion des fichiers.

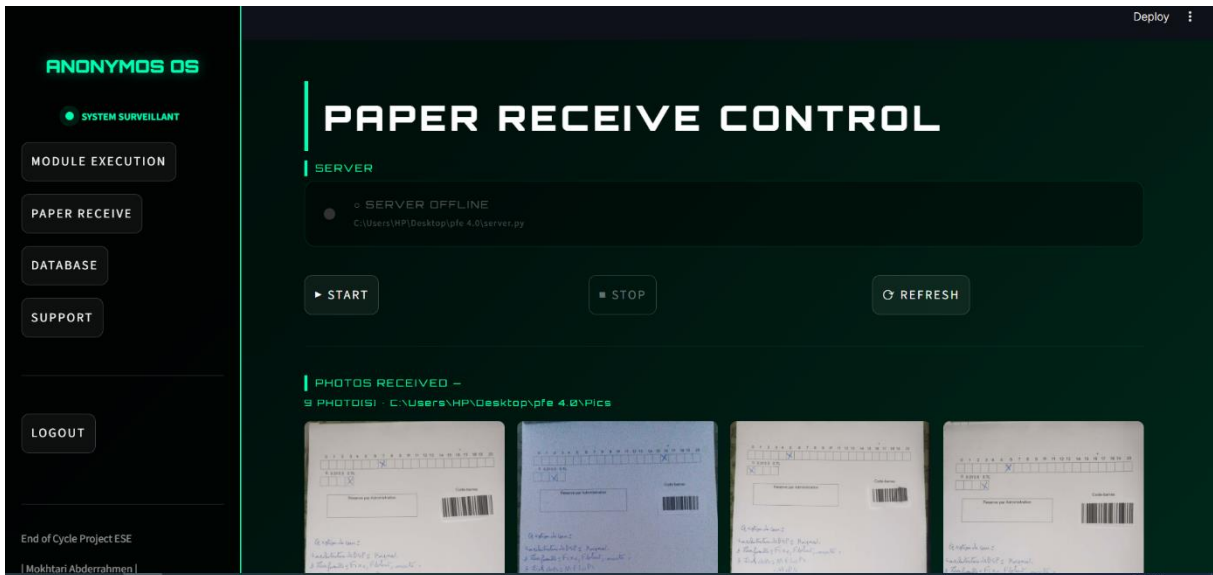


Figure III-7: Interface de réception des copies d'examen.

Chapitre III: Conception du système Anonymos OS

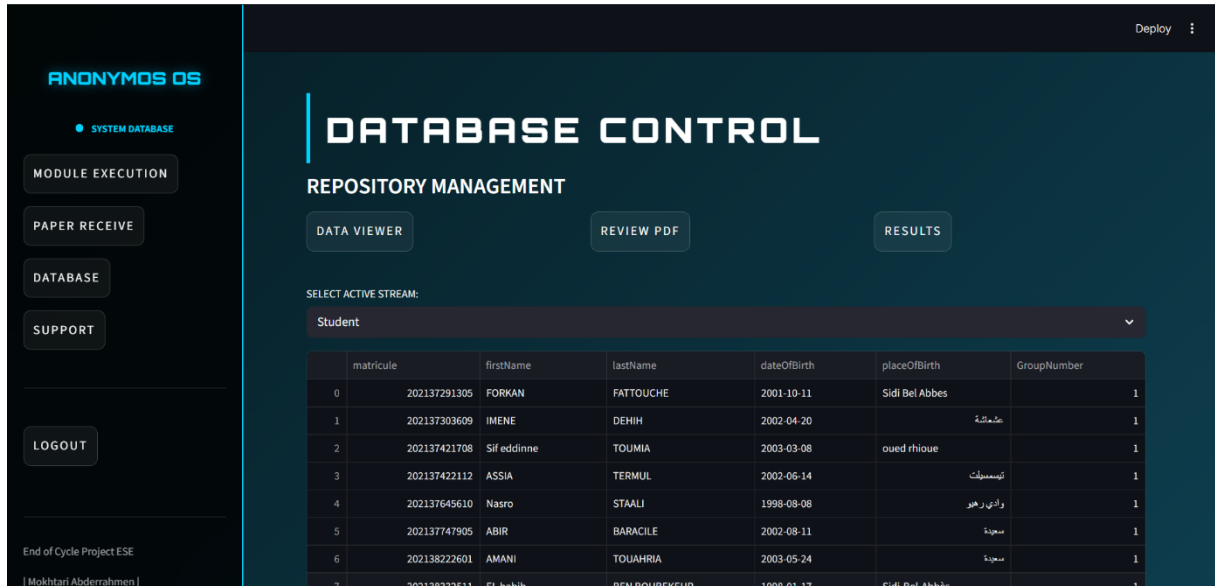


Figure III-8: Interface de consultation des données.

Un autre élément fondamental de la présente application est le DATABASE qui se trouve sur le côté gauche dans une barre spéciale. En ce qui concerne l'interface, cette dernière est le centre de contrôle et de gestion des tables de la base de données.

L'administrateur dispose également d'un accès direct aux différentes tables de la base de données à travers cette interface. Cette fonctionnalité lui permet de consulter les tables disponibles, d'explorer leur contenu et d'examiner les enregistrements qui y sont stockés afin de vérifier l'intégrité et la cohérence des données. Les informations accessibles concernent notamment les données des étudiants, les identifiants anonymes, les codes-barres générés, les notes extraites ainsi que les résultats finaux. Cette interface offre ainsi un moyen centralisé de supervision et de gestion des données nécessaires au fonctionnement de l'application. Grâce à cette fonctionnalité, l'administrateur peut effectuer les opérations de consultation et de contrôle directement depuis la plateforme, sans recourir à des outils externes d'administration de bases de données tels que phpMyAdmin ni à l'exécution manuelle de requêtes SQL. Cette approche améliore l'ergonomie du système, facilite les opérations de maintenance et réduit les risques d'erreurs liées aux manipulations directes de la base de données.

Chapitre III: Conception du système Anonymos OS

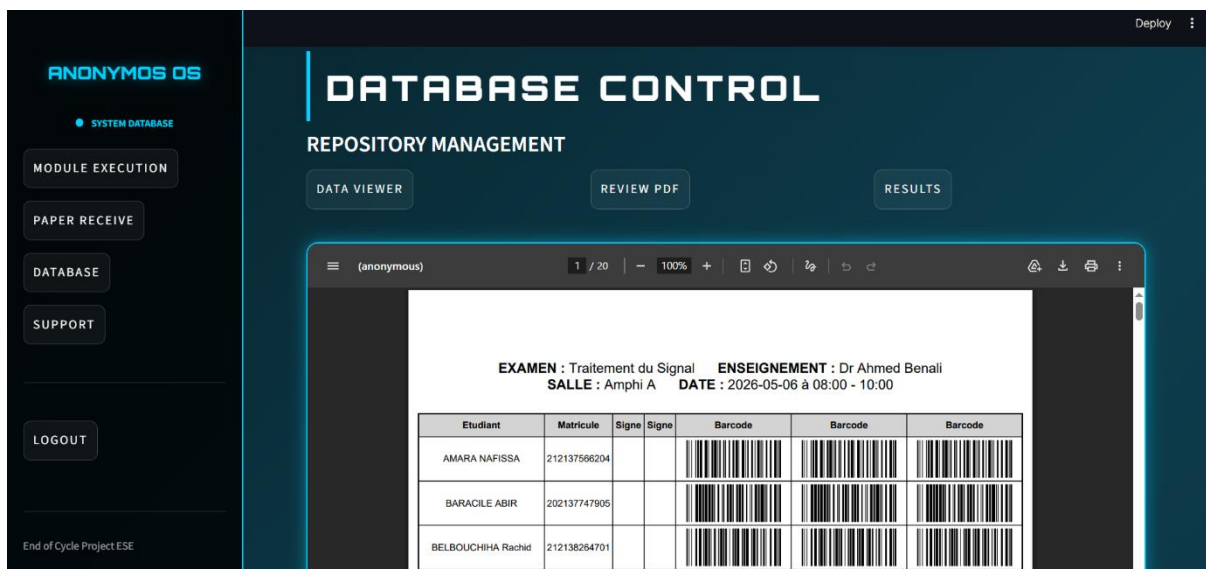


Figure III-9: Consultation des fichiers PDF dans la base de données

En outre, l'interface développée offre des capacités de pilotage et de supervision en temps réel de l'ensemble du système. À tout moment, l'utilisateur peut déclencher une opération spécifique depuis l'interface, ce qui entraîne immédiatement l'exécution du traitement correspondant au niveau du serveur.

Par exemple, la sélection du module « Barcode » déclenche automatiquement le processus de génération des codes-barres au sein du backend. De la même manière, l'activation du module « PDF » lance la création des documents PDF, qui sont ensuite mis à disposition de l'utilisateur sous une forme prête à être consultée ou téléchargée.

Cette interaction directe entre l'interface utilisateur et les modules de traitement contribue à assurer le fonctionnement en temps réel de l'application. Chaque action exécutée par l'utilisateur génère instantanément un retour visuel permettant de suivre l'état d'avancement des opérations en cours.

Ainsi, l'utilisateur peut superviser en permanence les différentes étapes du traitement, notamment la génération des documents, le décodage des informations extraites des copies, le processus de délibération ainsi que la transmission des résultats. Cette visibilité continue améliore la transparence du système, facilite le suivi des opérations et permet de détecter rapidement toute anomalie éventuelle.

Chapitre III: Conception du système Anonymos OS

Parmi les divers modules intégrés dans l'application, il existe un module qui se consacre uniquement au calcul automatique des scores académiques. L'objectif de ce module est de minimiser les interférences manuelles tout en limitant les erreurs dans les calculs des scores. Dans ce contexte, l'enseignant aura seulement à inscrire les scores CC via l'interface de l'application, alors que celle-ci effectuera les autres calculs requis en fonction de la règle éducative. Cela comprend l'application automatique des coefficients, les pondérations entre les divers aspects de l'évaluation, et le calcul de la moyenne finale de chaque élève.

Il est important de mentionner que tous les fichiers PDF générés par le système sont automatiquement archivés sur l'interface, spécifiquement sur le repository management. Il s'agit d'un moyen de veiller à ce que tous les fichiers générés par le système soient centralisés pour une facilité d'accès ultérieur.

The screenshot displays the ANONYMOS OS interface. On the left is a sidebar with navigation links: SYSTEM DATABASE, MODULE EXECUTION, PAPER RECEIVE, DATABASE, SUPPORT, and LOGOUT. The main area is titled 'DATABASE CONTROL' and 'REPOSITORY MANAGEMENT'. It includes buttons for 'DATA VIEWER', 'REVIEW PDF', and 'RESULTS'. A section labeled '4 DOCUMENT(S) FOUND' shows a 'SELECT DOCUMENT' dropdown menu with the following options: 'Microprocesseurs_et_Microcontrôleurs.pdf', 'Nothing selected', 'Microprocesseurs_et_Microcontrôleurs.pdf', 'Traitement_du_Signal.pdf', 'Électronique_Analogique.pdf', and 'Électronique_de_Puissance.pdf'. Below this, a table displays student data and scores.

212137594419	ANBARA	HOUSSEN	1.00	0.00	0.40
202137747905	BARACILE	ABIR	1.75	0.00	0.70
212138264701	BELBOUCHIHA	Rachid	13.00	0.00	5.20
212137594419	BELGACHEM	Sid Ahmed	17.75	0.00	7.10

Figure III-10: Table de contrôle des résultats dans la base de données

Chapitre III: Conception du système Anonymos OS

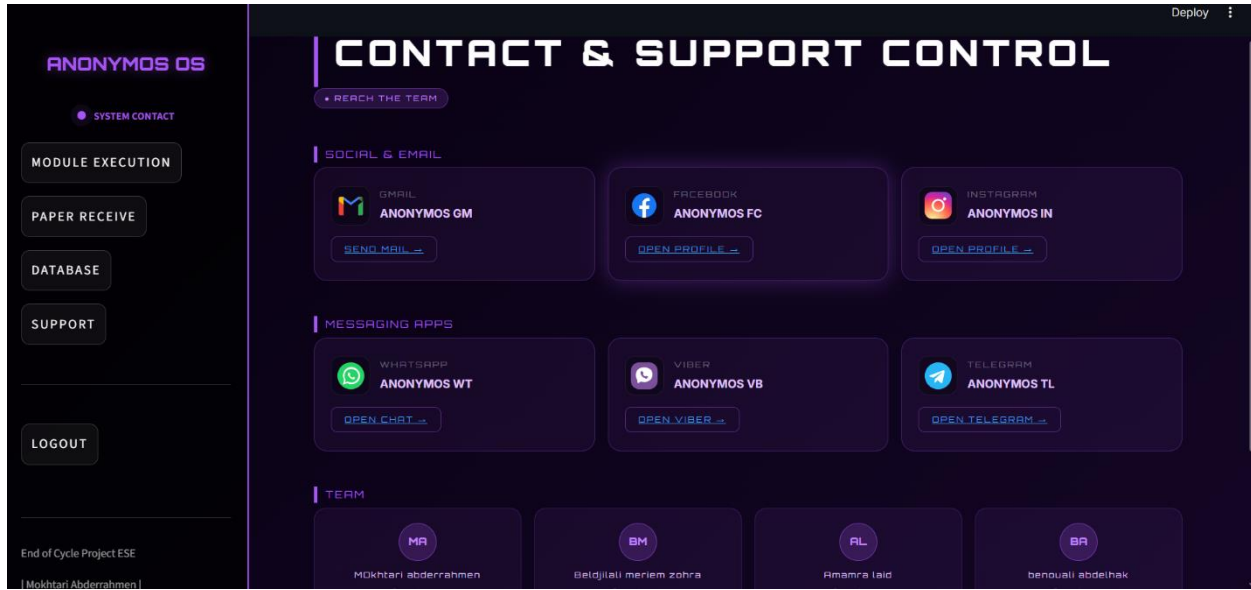


Figure III-11: Interface de support et de contrôle du système.

Enfin, nous n'oublions pas l'interface de support et de communication pour contacter l'équipe de développement en cas de problème ou de demande de renseignements.

La mise en place de cette interface constitue un élément central dans le fonctionnement global du système développé. Elle ne se limite pas à une simple couche de liaison entre les différentes fonctionnalités, mais représente une véritable plateforme de gestion intégrée regroupant l'ensemble des opérations du système au sein d'un environnement unique de travail. Grâce à cette approche, il n'est plus nécessaire de recourir à des outils externes pour piloter les différentes tâches, l'ensemble des fonctionnalités étant accessible depuis une seule interface.

Cette centralisation présente plusieurs avantages majeurs. Tout d'abord, elle permet une exécution en temps réel des opérations, puisque toute action déclenchée par l'utilisateur via l'interface entraîne une réponse immédiate du système. Ainsi, la génération des codes-barres, la création des fichiers PDF, l'interprétation des données, le lancement du processus de délibération ainsi que l'envoi des résultats sont exécutés à la demande, de manière instantanée.

Deuxièmement, cette architecture garantit une traçabilité complète des opérations. En effet, l'ensemble des étapes du traitement est suivi, enregistré et consultable, depuis la génération initiale des données jusqu'à la production finale des résultats et leur transmission automatique par courrier électronique. Cette visibilité globale permet non seulement de s'assurer du bon

Chapitre III: Conception du système Anonymos OS

déroulement des processus, mais également de faciliter la détection d'éventuelles erreurs ainsi que les opérations de maintenance et de vérification.

Troisièmement, cette approche améliore significativement l'ergonomie du système en regroupant dans une seule interface toutes les fonctionnalités liées à la gestion, à la visualisation et au traitement des résultats. L'utilisateur évolue ainsi dans un environnement unifié, simple et intuitif, lui permettant de piloter l'ensemble du système sans difficulté technique.

Cette interface dépasse largement le cadre d'un simple tableau de bord ou d'un outil d'exécution de commandes. Elle constitue un environnement intelligent de traitement, d'analyse, de gestion des données et de diffusion automatisée des informations. Elle permet à l'utilisateur de gérer l'intégralité du processus à partir d'une seule plateforme, tout en bénéficiant d'un retour d'information en temps réel sur l'ensemble des opérations effectuées.

Il s'agit donc d'un environnement interactif intelligent, capable de prendre en charge de manière complète et automatisée le cycle de production et de diffusion des informations, depuis la phase d'initialisation jusqu'à la génération finale et la distribution des résultats. Sa modularité, sa simplicité d'utilisation, son fonctionnement en temps réel ainsi que son haut niveau d'intégration en font une solution parfaitement adaptée aux exigences du présent projet.

- **API et orchestration du système :**

Grâce à l'API développée sous Flask, la communication intermodulée est permise. En effet, elle permet :

- De recevoir les images envoyées par l'ESP32 ;
- De déclencher le traitement ;
- De contrôler les échanges entre les différents composants.

Nous avons choisi de développer une architecture client-serveur permettant une bonne séparation de responsabilité.

Chapitre III: Conception du système Anonymos OS

III.5 Modélisation UML :

- Diagramme de cas d'utilisation :

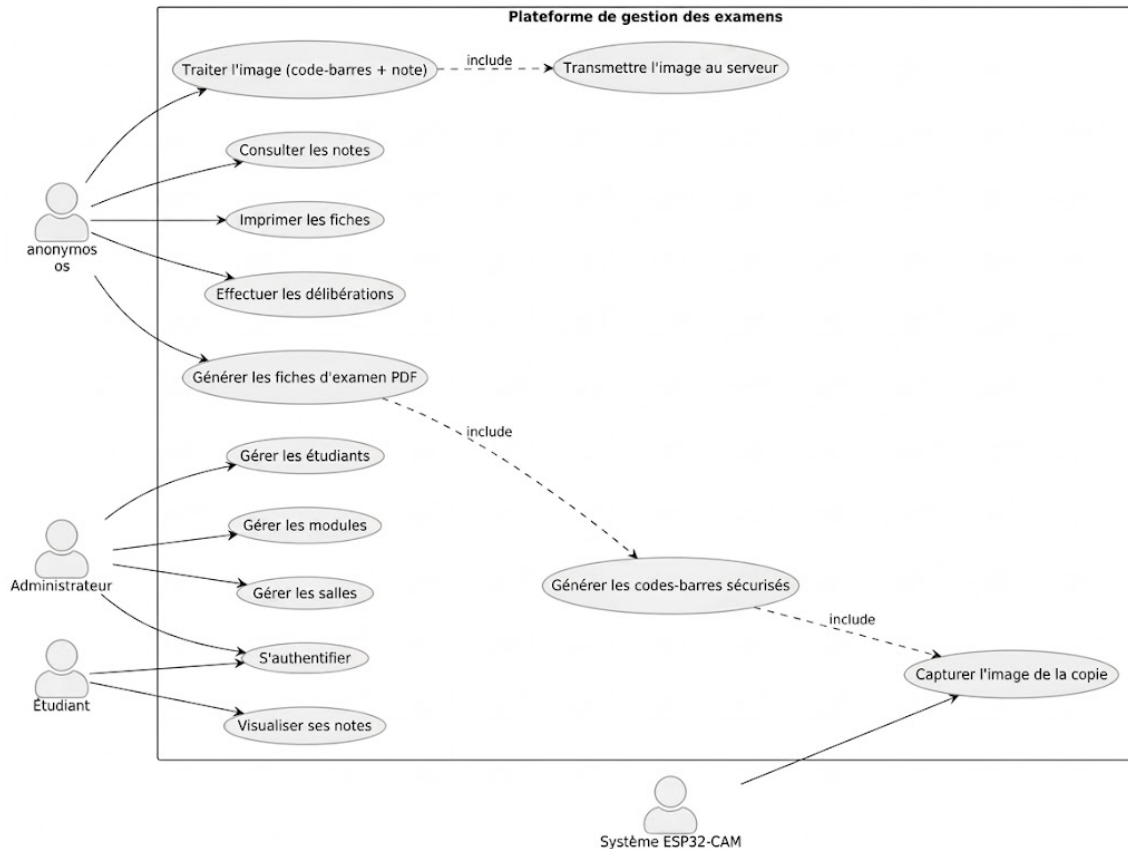


Figure III-12 : Diagramme de Cas d'Utilisation du système

Cette figure illustre les différents cas d'utilisation réalisés par les acteurs du système au sein de la plateforme de gestion de l'examen développée dans le cadre de ce travail. Elle souligne les principales caractéristiques de cette application et comment chacun de ces acteurs intervient dans le processus global de gestion, de traitement et de diffusion des résultats.

Le système repose sur quatre acteurs principaux : l'administrateur, l'étudiant, l'interface anonymos os et l'ESP32-S3-CAM, outre le serveur Flask qui sert de passerelle pour la communication entre les modules. À chaque acteur correspond une série de cas d'utilisations propres à sa participation dans cette plateforme.

L'interface anonymos os gère les opérations centrales comme le traitement des images (code-barres et notes), la consultation des notes, l'impression des fiches, la réalisation des délibérations et la génération des fiches d'examen PDF. Cette dernière fonctionnalité comprend la génération des codes-barres sécurisés, qui à leur tour déclenchent la capture de l'image de la copie assurée par le Système ESP32-CAM.

Chapitre III: Conception du système Anonymos OS

L'Administrateur est quant à lui en charge de la gestion des étudiants, des modules, des salles et de l'authentification.

L'Étudiant dispose d'un accès limité lui permettant de s'authentifier et de visualiser ses notes.

Enfin, la transmission des images vers le serveur est intégrée comme une étape du traitement des images, assurant ainsi une communication fluide entre les composants matériels et logiciels de la plateforme.

Chapitre III: Conception du système Anonymos OS

- Diagramme d'activités :

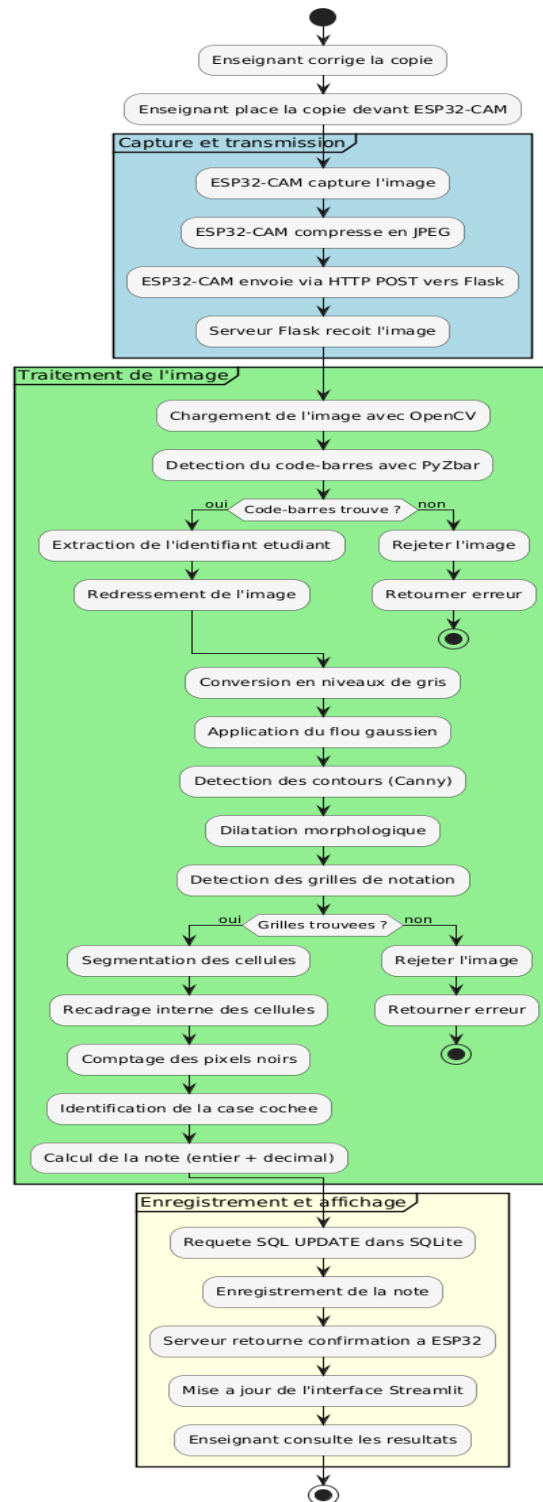


Figure III-13: Diagramme d'activités du système.

Chapitre III: Conception du système Anonymos OS

Le diagramme d'activité présenté ci-dessus illustre le processus de traitement automatisé des copies d'examen dans le cadre du système développé. Contrairement au diagramme de cas d'utilisation, qui met l'accent sur les interactions entre les acteurs et le système, le diagramme d'activité se concentre principalement sur la séquence chronologique des opérations exécutées, depuis la correction de la copie jusqu'à l'affichage de la note finale.

Dans ce processus, la première étape consiste en la correction manuelle de la copie par l'enseignant. Une fois la note inscrite sur la copie, celle-ci est présentée au module ESP32-S3-CAM, qui capture une image de la feuille d'examen contenant le score attribué. Les images obtenues sont ensuite compressées au format JPEG afin de réduire leur taille avant leur transmission, puis envoyées vers le serveur Flask via une requête HTTP de type POST.

Dès réception de l'image, celle-ci est transmise au module de traitement développé en Python. Le processus débute par le chargement de l'image à l'aide de la bibliothèque OpenCV, suivi de la détection du code-barres via le package PyZbar. Cette étape est essentielle, car le code-barres permet d'identifier de manière unique l'étudiant associé à la copie.

Le diagramme met en évidence un premier point de décision : si le code-barres ne peut pas être détecté, le système signale une erreur liée à l'image reçue. Dans le cas contraire, l'étudiant est correctement identifié et l'image est réorientée si nécessaire afin de garantir la fiabilité des traitements ultérieurs.

Après cette phase d'identification, l'image subit plusieurs opérations de prétraitement visant à améliorer sa qualité. Elle est d'abord convertie en niveaux de gris afin de réduire la complexité des données. Ensuite, un flou gaussien est appliqué pour réduire le bruit visuel. Le traitement se poursuit par une détection des contours à l'aide de l'algorithme de Canny, suivie d'une opération de dilatation morphologique permettant de renforcer les structures détectées.

À l'issue de ce prétraitement, le système procède à la localisation des grilles de notation présentes sur la copie. Un second point de décision intervient à ce niveau : si la détection des grilles échoue, le document est considéré comme non valide et un message d'erreur est généré. En revanche, si la détection est réussie, le traitement se poursuit par la segmentation des grilles en cellules correspondant aux différentes valeurs possibles de la note.

Chapitre III: Conception du système Anonymos OS

Afin d'améliorer la précision de l'analyse, une étape de recadrage interne est appliquée à chaque cellule, ne conservant que la zone centrale afin de réduire les interférences dues aux bordures de la grille.

L'étape suivante consiste à analyser le contenu de chaque cellule. L'algorithme calcule le nombre de pixels noirs présents dans chaque zone et considère comme case sélectionnée celle présentant la densité la plus élevée. Cette méthode permet d'identifier séparément les valeurs correspondant à la partie entière et à la partie décimale de la note. Les deux valeurs sont ensuite combinées pour obtenir la note finale de l'étudiant.

La dernière étape du processus concerne l'enregistrement et la restitution des résultats. La note calculée est associée au code-barres décodé, permettant ainsi d'identifier l'étudiant concerné. Une requête SQL de type UPDATE est ensuite exécutée afin de mettre à jour la base de données SQLite.

Après validation de l'opération, le serveur renvoie une notification de traitement, et les résultats sont immédiatement affichés via l'interface Streamlit. L'enseignant peut alors consulter les notes obtenues et poursuivre les opérations d'évaluation.

Cependant, on remarque facilement que ce diagramme d'activité met clairement en lumière le processus qui est accompli par notre programme, depuis la numérisation du papier jusqu'à l'enregistrement de la note obtenue. On observe également les procédures qui servent à la détection des erreurs qui peuvent apparaître pendant ce processus (par exemple le manque de code-barres ou de la grille de notation).

Chapitre III: Conception du système Anonymos OS

- Diagramme de séquences :

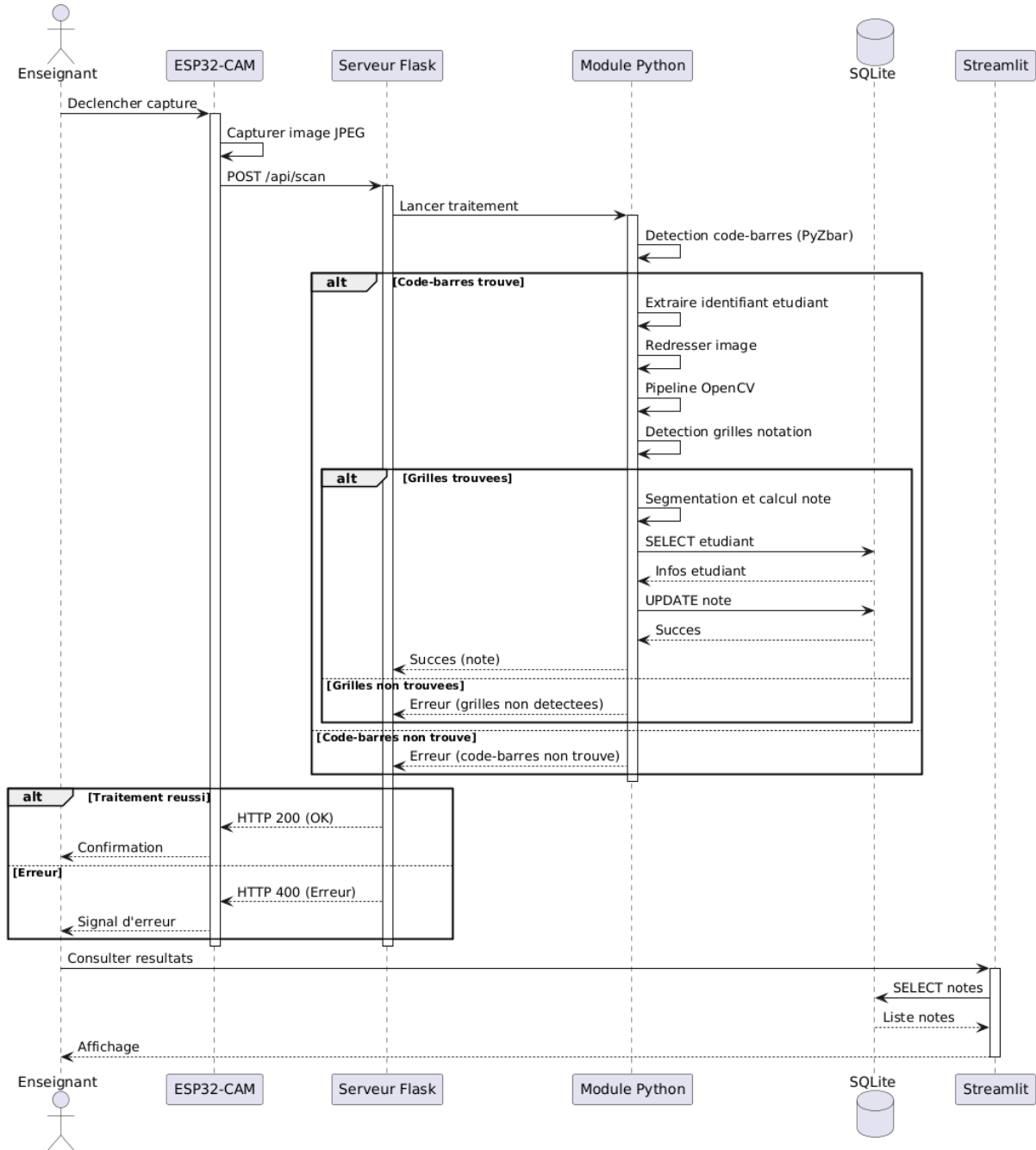


Figure III-14: Diagramme de séquence du système

Lors de l'analyse du diagramme de séquence présenté ci-dessus, on comprend le déroulement complet du processus d'anonymisation et de délibération basé sur la capture des copies d'examen. Ce diagramme permet de mettre en évidence l'interdépendance entre les différents composants du système, lesquels collaborent de manière coordonnée afin d'assurer l'exécution automatique et ordonnée du processus.

Chapitre III: Conception du système Anonymos OS

Le processus est initialisé par une action manuelle de l'enseignant, qui déclenche la capture des copies via l'application. Cette action est ensuite prise en charge par le module ESP32-S3-CAM, qui réalise la prise de vue de la copie d'examen avant de transmettre l'image au serveur Flask.

Le serveur joue alors un rôle central dans la coordination des opérations, en redirigeant les données vers le module de traitement développé en Python. Ce dernier assure l'analyse de l'image, l'identification du code-barres associé à l'étudiant ainsi que l'extraction automatique de la note.

À chaque étape du processus, le système intègre des mécanismes de gestion des erreurs afin d'assurer sa robustesse. Ainsi, en cas d'échec de lecture du code-barres ou de non-détection de la grille de notation, un message d'erreur explicite est renvoyé à l'enseignant, lui permettant d'intervenir sans compromettre les données déjà traitées. À l'inverse, lorsque le traitement est effectué avec succès, un message de confirmation est généré et la note est automatiquement enregistrée dans la base de données.

Une fois l'ensemble des copies traitées, l'enseignant peut consulter l'ensemble des résultats directement via l'interface de l'application, ce qui permet de finaliser le processus de délibération sans intervention manuelle sur l'identité des étudiants.

Ce diagramme de séquence illustre clairement le fonctionnement dynamique du système ainsi que la coordination entre ses différents modules, depuis le déclenchement de la capture jusqu'à la validation et la consultation des résultats, tout en garantissant l'automatisation et la sécurité du processus.

III.6 Communication entre l'ESP32 et le PC:

La communication entre le module embarqué ESP32-S3-CAM et le système de traitement côté PC constitue une étape essentielle dans le fonctionnement global de la plateforme. Elle permet d'assurer le transfert des images capturées vers le serveur de traitement afin d'extraire automatiquement les informations contenues sur les copies d'examen, notamment les codes-barres et les notes attribuées par l'enseignant.

Chapitre III: Conception du système Anonymos OS

Dans ce projet, la communication a été conçue de manière à être simple, rapide et adaptée aux contraintes d'un système embarqué temps réel. L'ESP32-S3-CAM agit principalement comme un module d'acquisition d'images, tandis que le traitement intensif des données est réalisé sur le PC à travers des scripts Python et une API Flask.

III.6.1 Protocole de communication :

Afin d'assurer l'échange des données entre l'ESP32-S3-CAM et le serveur de traitement, le protocole HTTP a été utilisé. Ce choix s'explique par sa simplicité d'implémentation, sa compatibilité avec les applications web ainsi que sa facilité d'intégration avec les bibliothèques Python utilisées dans le projet.

Le module ESP32-S3-CAM se connecte au réseau Wi-Fi local puis communique avec le serveur Flask hébergé sur le PC. Une fois la copie d'examen photographiée, l'image capturée est envoyée au serveur à travers une requête HTTP de type POST vers une route spécifique de l'API REST, généralement définie sous la forme :

`/api/scan`

Le serveur Flask joue alors le rôle d'intermédiaire entre le module embarqué et le système de traitement d'image développé en Python. Dès réception de l'image, le serveur déclenche automatiquement les différentes opérations nécessaires à l'analyse de la copie.

Le fonctionnement global de la communication peut être résumé selon les étapes suivantes :

1. Initialisation de la connexion Wi-Fi sur l'ESP32-S3-CAM ;
2. Capture de l'image de la copie d'examen ;
3. Compression et préparation de l'image ;
4. Envoi de l'image vers le serveur Flask via HTTP POST ;
5. Réception et sauvegarde temporaire de l'image côté serveur ;
6. Lancement du module Python de traitement ;
7. Décodage du code-barres et reconnaissance de la note ;
8. Mise à jour de la base de données SQLite ;
9. Retour d'une réponse de confirmation à l'ESP32-S3-CAM ;
10. Affichage des résultats dans l'application web Streamlit.

Chapitre III: Conception du système Anonymos OS

Cette architecture permet de séparer les tâches matérielles des traitements lourds. L'ESP32-S3-CAM se limite ainsi à l'acquisition des images, tandis que les opérations complexes de vision par ordinateur sont exécutées sur le PC, disposant de ressources de calcul plus importantes.

Par ailleurs, l'utilisation du protocole HTTP facilite l'évolution future du système, notamment l'intégration d'un serveur distant ou d'un hébergement cloud.

III.6.2 Format des données échangées

L'ESP32-S3-CAM et le serveur FLASK échangent principalement des données constituées d'images qui représentent les copies d'examen. Ces images comprennent à la fois :

- Le code-barres anonyme de l'étudiant ;
- La zone de notation ;
- Les cases correspondant aux notes ;
- Les marques manuscrites effectuées par l'enseignant.

L'image est en général transmise au format JPEG afin de réduire la taille des données envoyées tout en conservant une qualité suffisante pour le traitement d'image. Lors de l'envoi au serveur, l'ESP32-S3-CAM utilise une requête qui contient :

- L'image capturée.
- Les informations d'identification du périphérique.
- Le type de contenu (Content-Type).

Le système réalise ensuite plusieurs étapes de traitement :

- Conversion de l'image en niveaux de gris.
- Application du filtre Gaussian Blur pour réduire le bruit.
- Détection des contours avec l'algorithme Canny.
- Segmentation des zones de notation.
- Analyse de densité des pixels.
- Détection de la case cochée.
- Extraction de la note finale.

Une fois l'image reçue, le serveur FLASK transmet les données au module Python de traitement d'image.

Chapitre III: Conception du système Anonymos OS

Ce module fait appel à plusieurs bibliothèques spécialisées qui extraient automatiquement les informations utiles :

- OpenCV pour le prétraitement et la détection des contours ;
- NumPy pour les opérations matricielles ;
- PyZbar pour le décodage des codes-barres ;
- SQLite3 pour l'enregistrement des résultats.

Une fois les données extraites, le programme génère automatiquement une requête SQL de type UPDATE afin d'inscrire la note dans la base de données SQLite. Une fois le traitement effectué, le serveur renvoie une réponse indiquant si l'opération a réussi ou échoué. Cette réponse peut comprendre :

- L'identifiant de l'étudiant.
- La note détectée.
- L'état du traitement.
- Un message de confirmation.

Enfin, les résultats enregistrés dans la base SQLite sont affichés automatiquement dans l'interface web Streamlit pour permettre à l'enseignant de consulter les notes, vérifier les résultats et effectuer les opérations de délibération.

Chapitre III: Conception du système Anonymos OS

III.7 Conclusion :

Cette étape a permis de décrire avec précision la conception de la plateforme de gestion et d'anonymisation des examens développée dans le cadre de ce projet. L'analyse de l'architecture du système, a mis en évidence sa modularité ainsi que les interactions entre les différents composants matériels et logiciels qui le constituent. Au cours de cette phase, les différents diagrammes élaborés ont servi à formaliser l'ensemble du processus de réalisation de la solution proposée, allant de la génération sécurisée des codes-barres jusqu'à l'attribution des évaluations, en passant par la récupération des copies, le traitement d'image, le décodage des réponses, le processus de délibération automatisée ainsi que la publication des résultats.

Une attention particulière a été accordée à la conception de l'interface utilisateur, qui constitue le point central de contrôle du système. Celle-ci regroupe l'ensemble des fonctionnalités de pilotage, de supervision, de calcul, de consultation et de transmission des résultats, offrant ainsi une gestion unifiée et centralisée de l'application. C'est donc grâce à cette étape de conception que nous avons pu mettre en place une architecture cohérente, robuste et adaptée aux contraintes fonctionnelles du projet. Elle constitue une base solide pour le développement et l'intégration des différents modules du système.

Le prochain chapitre sera consacré à l'implémentation expérimentale de la solution ainsi qu'à la présentation des différents tests réalisés pour s'assurer du bon fonctionnement de ce dernier et de ses performances respectives en rapport avec des critères définis.

Chapitre IV: Protocol test ANONYMOS OS

Chapitre IV:Protocol test ANONYMOS OS

IV.1 Introduction

Après avoir exposé avec précision les aspects théoriques, l'architecture et la mise en œuvre matérielle et logicielle de notre solution ANONYMOS OS, ce qui se fait dans ce chapitre, il est temps de faire une pause pour évoquer la partie la plus importante du projet qui est la phase de tests. Le test fait partie intégrante de notre plateforme et constitue son point culminant parce qu'il va permettre d'assurer l'intégrité de notre système par rapport aux exigences initialement définies dans le cahier des charges.

Le but essentiel du présent test est de confirmer la faisabilité, la solidité et la pertinence de notre plateforme automatisée de gestion et de création de documents d'examen pour les universités. Dans le cadre de notre projet qui est un peu hybride en termes de technologies utilisées, la validation devra nécessairement englober deux aspects technologiques importants :

- Le premier axe concerne le logiciel le logiciel et la base de données : C'est la vérification de la fluidité de l'interaction de l'interface réalisée à l'aide de la bibliothèque Streamlit, la cohérence des instructions dans la base de données SQLite et aussi la fluidité de la création automatisée des documents d'examen au format PDF, avec les données des étudiants et les codes barres.
- Le second axe se rapporte au matériel et au réseau : C'est la validation de la chaîne de traitement d'acquisition qui permettra au serveur de la caméra ESP32-S3-CAM de transférer efficacement les flux image vers l'application centrale.

Afin de construire cette démarche de validation, nous avons élaboré une utilisation complète, en suggérant la situation réelle d'exploitation dans une université. En ce qui concerne les prochaines étapes, nous allons illustrer, dans un ordre chronologique et approfondi, toutes les opérations importantes de ce système. En effet, chacune de ces opérations est soutenue par des écrans de l'interface, des photos de l'appareil, et surtout une étude technique de ce que nous avons obtenu..

IV.2 Phase de sécurisation, d'authentification et d'accès global à l'application ANONYMOS OS

Le premier volet de l'aspect opérationnel de notre protocole de test vise à valider le mécanisme de sécurité et de restriction d'accès. Le ANONYMOS OS, il fait référence au tout du système matériel et logiciel, conçu pour automatiser et sécuriser les documents d'examen universitaire.

Chapitre IV: Protocol test ANONYMOS OS

En raison de l'information sensible qui sera manipulée (dans ce cas : informations de confidentialité, résultats des étudiants, codes-barres générés), la présence d'un mur de sécurité avant toute application s'imposait à titre de critère indispensable.

En effet, comme on peut le voir dans l'illustration de la figure ci-dessous, l'initialisation du système force l'utilisateur à naviguer jusqu'à une page d'accueil personnalisée, jouant également le rôle d'une interface d'accès sécurisé (Secure Access Terminal). Ici, toutes les fonctionnalités de la plate-forme sont verrouillées et invisibles. L'interface, minimaliste et sécurisée, contraint l'utilisateur (typiquement, un administrateur ou enseignant du service d'examen) à remplir deux champs obligatoires :

USER ID : L'identifiant texte de l'opérateur, défini par le rôle de l'utilisateur au sein du système (saisi sous l'identifiant "Admin" dans le présent cas).

Access Key : La clé d'accès, code alphanumérique secret.

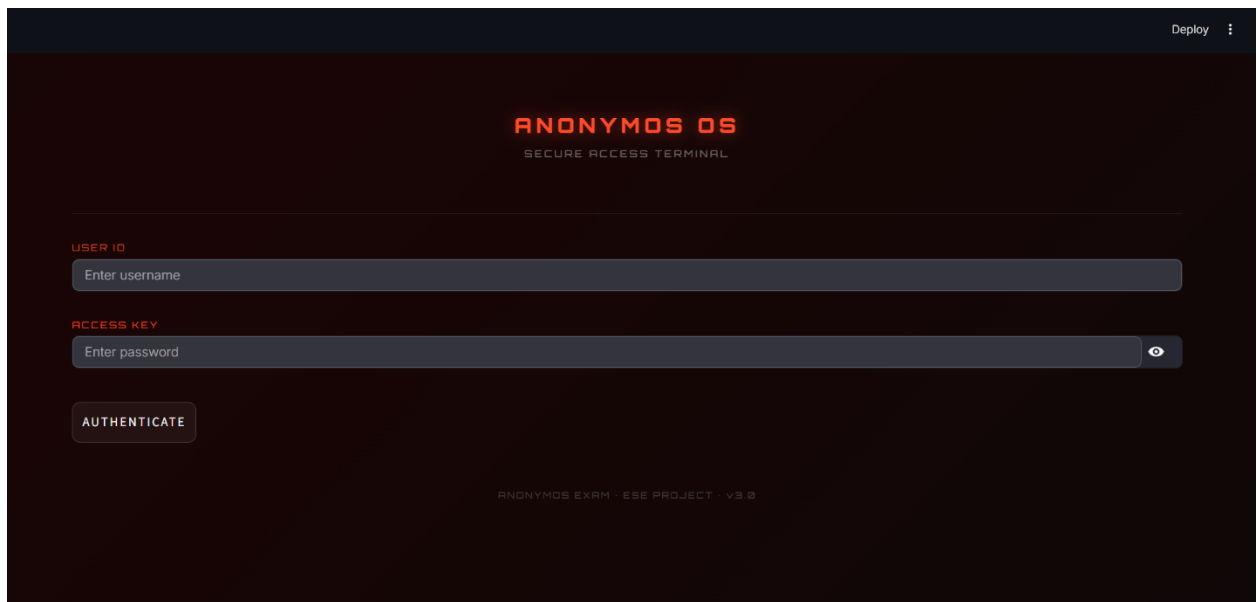


Figure IV-1 Interface d'authentification des utilisateurs

IV.3 Procédure d'Exécution du Module de Génération de Codes-barres D'anonymisation (Barcode)

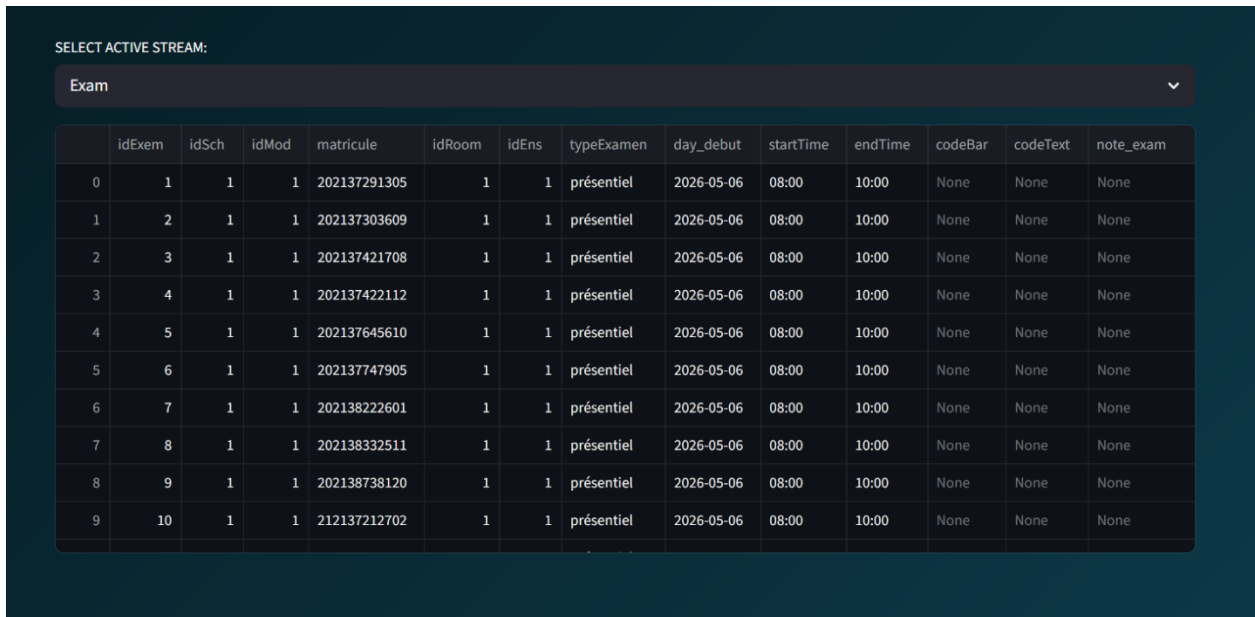
L'opération principale après l'authentification dans notre application ANONYMOS OS est le paramétrage de l'anonymisation des fiches examen. L'anonymisation s'effectue par le biais de la création d'une suite aléatoire de chaîne de caractère et leur conversion sous forme de codes-barres

Chapitre IV:Protocol test ANONYMOS OS

graphiques. Pour tester le bon fonctionnement de ce module, nous appliquons une procédure de test en avant/après exécution.

IV.3.1 Phase d'avant-exécution : État initial du système

L'examen de la table « Exam » par notre responsable, comme illustré dans la figure 4.2 ci-dessous, montre que les colonnes essentielles à la sécurité et à l'anonymat de l'examen, à savoir « codeBar » et « codeText », sont vides et contiennent la valeur NULL. De même, la colonne « note_exam » reste vide à ce stade.



SELECT ACTIVE STREAM:

Exam

	idExem	idSch	idMod	matricule	idRoom	idEns	typeExamen	day_debut	startTime	endTime	codeBar	codeText	note_exam
0	1	1	1	202137291305	1	1	présentiel	2026-05-06	08:00	10:00	None	None	None
1	2	1	1	202137303609	1	1	présentiel	2026-05-06	08:00	10:00	None	None	None
2	3	1	1	202137421708	1	1	présentiel	2026-05-06	08:00	10:00	None	None	None
3	4	1	1	202137422112	1	1	présentiel	2026-05-06	08:00	10:00	None	None	None
4	5	1	1	202137645610	1	1	présentiel	2026-05-06	08:00	10:00	None	None	None
5	6	1	1	202137747905	1	1	présentiel	2026-05-06	08:00	10:00	None	None	None
6	7	1	1	202138222601	1	1	présentiel	2026-05-06	08:00	10:00	None	None	None
7	8	1	1	202138332511	1	1	présentiel	2026-05-06	08:00	10:00	None	None	None
8	9	1	1	202138738120	1	1	présentiel	2026-05-06	08:00	10:00	None	None	None
9	10	1	1	212137212702	1	1	présentiel	2026-05-06	08:00	10:00	None	None	None

Figure IV-2 : Table des examens de la base de données

Du côté utilisateur, le sous-module Barcode du Control Panel Streamlit, qui est visible dans la figure 4.3, reste dans son état initial. La zone d'information du TERMINAL · BARCODE affiche l'étiquette système de message en attente NO OUTPUT YET, démontrant qu'aucune procédure n'a été lancée.

L'opérateur lance ensuite la procédure en cliquant sur le bouton INITIALIZE BARCODE, ce qui déclenche immédiatement les scripts de génération de codes-barres.

Chapitre IV: Protocol test ANONYMOS OS

Comme mis en évidence sur la Figure 4.4 , le terminal dédié change de couleur et d'état. Il affiche un horodatage précis accompagné de la mention de succès COMPLETED. Le retour textuel du script indique l'exécution d'une fonction sécurisée (240 secure barcode), confirmant que le traitement algorithmique s'est déroulé sans aucune erreur d'exécution.

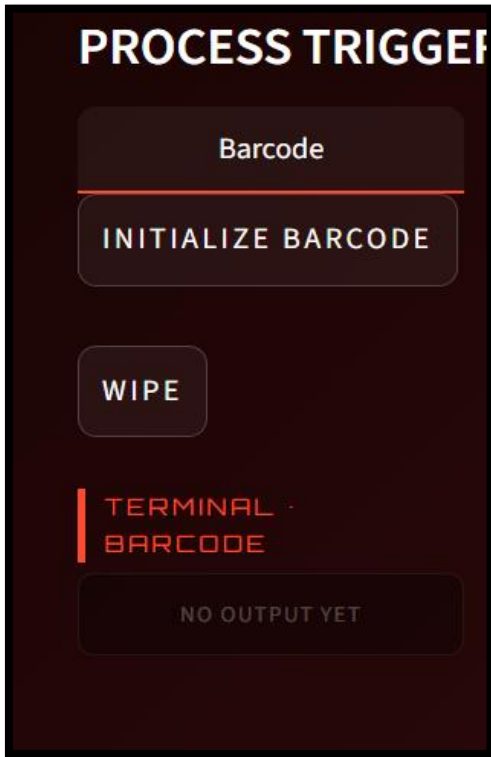


Figure IV-3 : Interface avant l'initialisation des codes-barres

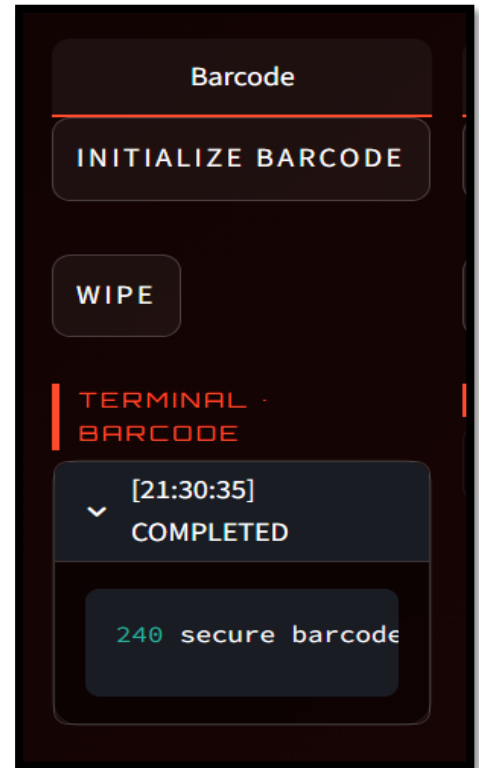
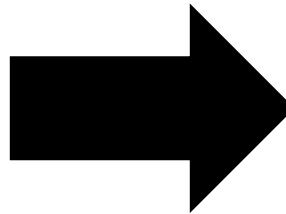


Figure IV-4 : Interface après l'initialisation des codes-barres

Pour prouver que nos données sont persistantes, nous mettons à jour la table Exam, conformément à ce qui est montré dans la figure 4.5. On s'aperçoit qu'il y a eu un changement instantané et totalement correct :

- La colonne codeBar est remplie avec des objets binaires du type BLOB, c'est-à-dire les images des codes-barres qui sont déjà prêtes à être insérées sur les feuilles fiches PDF d'examen.
- Le champ codeText est maintenant rempli avec des identifiants anonymes textuels, lesquels ont été générés de manière aléatoire.

Chapitre IV:Protocol test ANONYMOS OS

SELECT ACTIVE STREAM:

Exam

	idEns	typeExamen	day_debut	startTime	endTime	codeBar	codeText	note_exam
0	1	présentiel	2026-05-06	08:00	10:00	137,80,78,71,13,10,26,10,0,0,0,13,73,72,68,82,0,0,1,68,0,0,0,200,8,2,0,0,0,31,141,15,13	DS4OHWP	None
1	1	présentiel	2026-05-06	08:00	10:00	137,80,78,71,13,10,26,10,0,0,0,13,73,72,68,82,0,0,1,68,0,0,0,200,8,2,0,0,0,31,141,15,13	ZL14B5R	None
2	1	présentiel	2026-05-06	08:00	10:00	137,80,78,71,13,10,26,10,0,0,0,13,73,72,68,82,0,0,1,68,0,0,0,200,8,2,0,0,0,31,141,15,13	62HUH45	None
3	1	présentiel	2026-05-06	08:00	10:00	137,80,78,71,13,10,26,10,0,0,0,13,73,72,68,82,0,0,1,68,0,0,0,200,8,2,0,0,0,31,141,15,13	KVTTJVT	None
4	1	présentiel	2026-05-06	08:00	10:00	137,80,78,71,13,10,26,10,0,0,0,13,73,72,68,82,0,0,1,68,0,0,0,200,8,2,0,0,0,31,141,15,13	IHBDJV6	None
5	1	présentiel	2026-05-06	08:00	10:00	137,80,78,71,13,10,26,10,0,0,0,13,73,72,68,82,0,0,1,68,0,0,0,200,8,2,0,0,0,31,141,15,13	PPFLLIQ	None
6	1	présentiel	2026-05-06	08:00	10:00	137,80,78,71,13,10,26,10,0,0,0,13,73,72,68,82,0,0,1,68,0,0,0,200,8,2,0,0,0,31,141,15,13	TBBU7VP	None
7	1	présentiel	2026-05-06	08:00	10:00	137,80,78,71,13,10,26,10,0,0,0,13,73,72,68,82,0,0,1,68,0,0,0,200,8,2,0,0,0,31,141,15,13	DDK4W2T	None
8	1	présentiel	2026-05-06	08:00	10:00	137,80,78,71,13,10,26,10,0,0,0,13,73,72,68,82,0,0,1,68,0,0,0,200,8,2,0,0,0,31,141,15,13	WESU3PF	None
9	1	présentiel	2026-05-06	08:00	10:00	137,80,78,71,13,10,26,10,0,0,0,13,73,72,68,82,0,0,1,68,0,0,0,200,8,2,0,0,0,31,141,15,13	3BU32FO	None

Figure IV-5 : Table des examens après la mise à jour des données

IV.4 Protocole d'exécution du module de génération des fiches PDF

Après la réussite de l'implantation des codes-barres anonymes au sein de la base de données, le processus se poursuit par la vérification de la fiabilité du module d'édition des documents. Ceci étant fait, l'automatisation procède à l'extraction des informations sur les étudiants, en leur associant leur code-barre respectif pour créer leur fiche d'émargement/examen.

Il s'agit du protocole qui commence par la page du dépôt de prévisualisation, illustrée dans la Figure 4.6 . Avant de faire le processus, un message d'avertissement explicite sous la forme de bandeau est affiché en disant que le LE SYSTÈME AFFICHE LE MESSAGE TARGET NOT GENERATED YET.. Cela signifie que le processus de generation du fichier est toujours à venir

Chapitre IV:Protocol test ANONYMOS OS

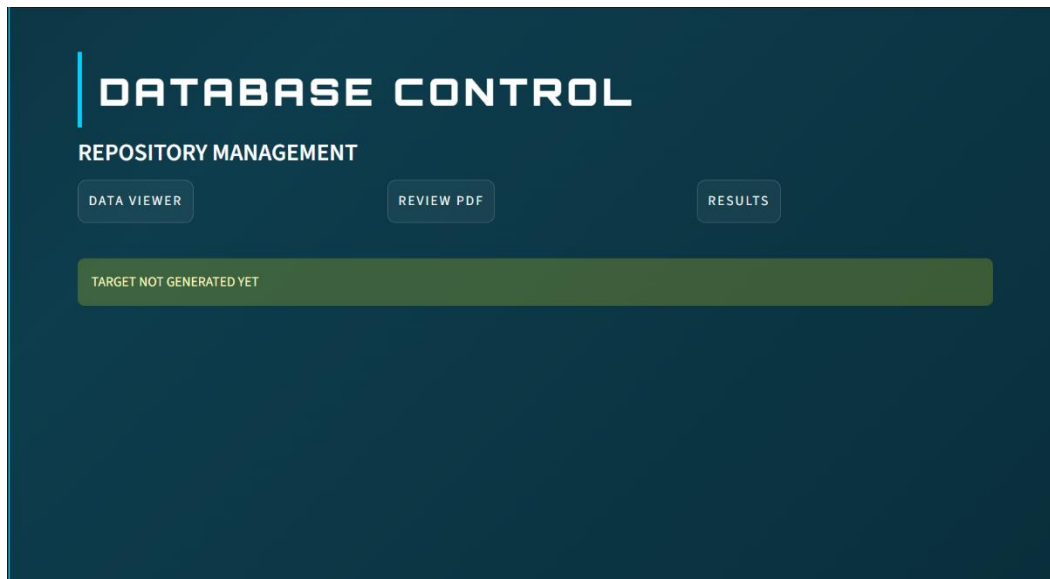


Figure IV-6 : Aperçu PDF avant la génération des documents

Il faut noter que l'opérateur passe alors à la console centrale d'exécution, comme montré sur la Figure 4.7. Toutefois, à cet instant-là, le sous-module PDF de l'unité logicielle est bien dans son état initial. L'outil dédié `TERMINAL · PDF` est à l'état de repos complet avec le message naturel `NO OUTPUT YET`, signifiant qu'aucun processus de compilation n'est lancé.

Le processus d'exécution du code est lancé par l'opérateur avec la commande `GENERATE`. Selon la présentation de la Figure 4.8, le script exécuté en arrière-plan est exécuté correctement et finit son processus avec un résultat positif. La console Streamlit passe immédiatement à l'état `COMPLETED` avec l'heure exacte et affiche `PDF GENERATED`.



Figure IV-8 Interface avant la génération du fichier PDF

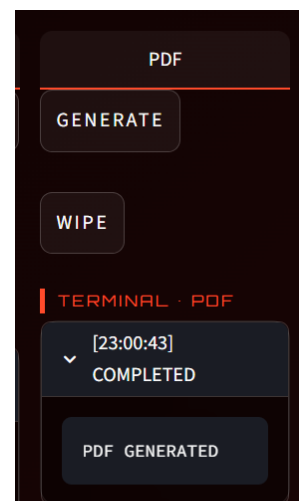
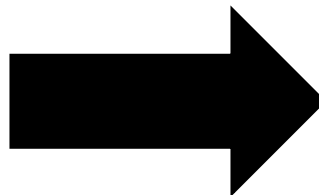


Figure IV-7 Interface après la génération du fichier PDF.

Chapitre IV:Protocol test ANONYMOS OS

La validation finale s'opère en retournant sur la page de gestion du hub de données, comme montré sur la Figure 4.9. Le bandeau d'avertissement précédent a disparu pour laisser place au module de visualisation active. L'activation du bouton REVIEW PDF affiche dynamiquement au sein de l'interface le document PDF correctement généré.

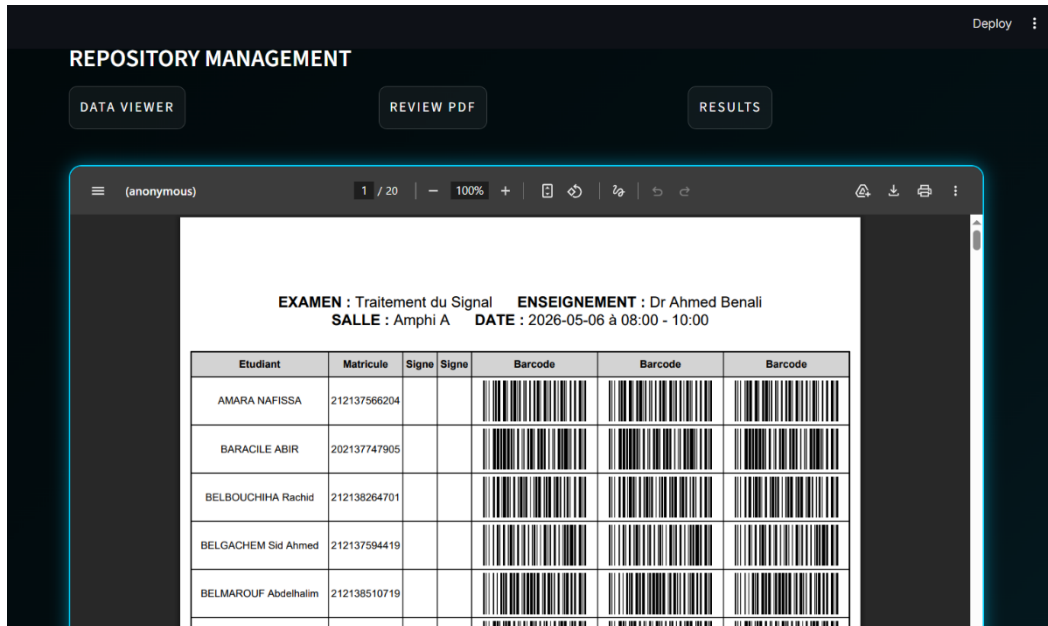


Figure IV-9 Aperçu PDF après la génération du fichier PDF.

IV.5 Protocole d'activation du serveur de réception d'images (ESP32-S3-CAM)

Ce module portant le nom de PAPER RECEIVE CONTROL **est** assure la gestion d'un serveur local qui est capable de capturer et de conserver en temps réel les images envoyées par la caméra ESP32-S3-CAM intégrée au terminal mobile du surveillant. Cette procédure est mise à l'épreuve par une séquence tripartite bien structurée de vérification.

La procédure se termine par la fenêtre de contrôle du serveur en sa configuration initiale telle que représentée à la Figure 4.10. Ainsi, sans aucune action de la part de l'utilisateur, le bandeau supérieur montre , accompagné de la phrase SERVER OFFLINE. De même, le bandeau inférieur

Chapitre IV:Protocol test ANONYMOS OS

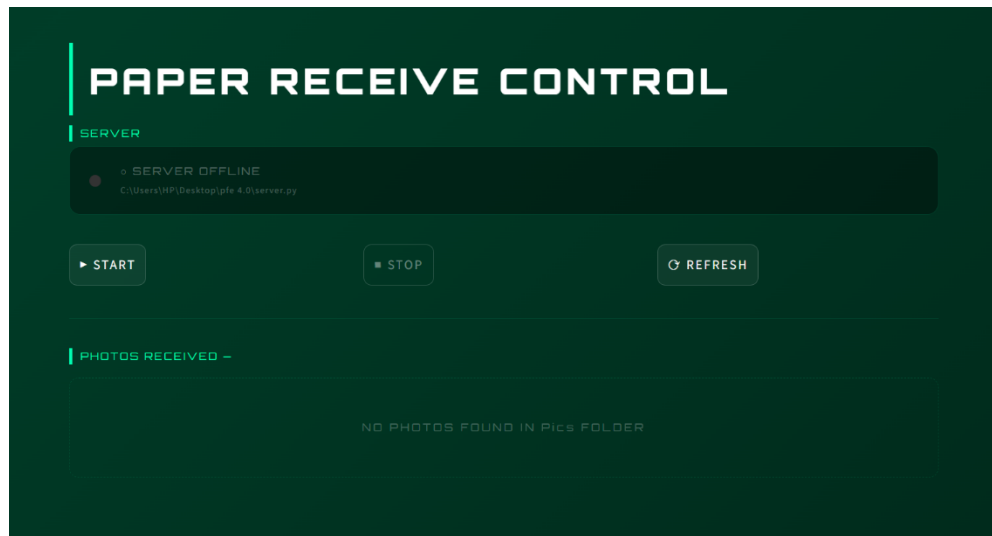


Figure IV-10 contrôleur de réception des copies avant le démarrage du traitement.

réservé à la gestion des photos stockées (PHOTOS RECEIVED) signale une absence d'informations sous la forme suivante : NO PHOTOS FOUND IN PICS FOLDER.

L'opérateur sélectionne alors la commande START pour déclencher le script réseau. Comme illustré dans la Figure 4.11 suivante, la réponse de l'interface utilisateur de Streamlit est immédiate : le message du statut SERVER ACTIVE — WAITING FOR ESP32 IMAGES s'affiche. La fenêtre d'accueil indique par son comportement visuel que l'état de mise en attente est effectivement actif : SERVEUR ACTIF – EN ATTENTE DE PHOTOS..... Le processus est maintenant prêt à capter les demandes HTTP/POST ou sockets de l'ESP32-S3-CAM.

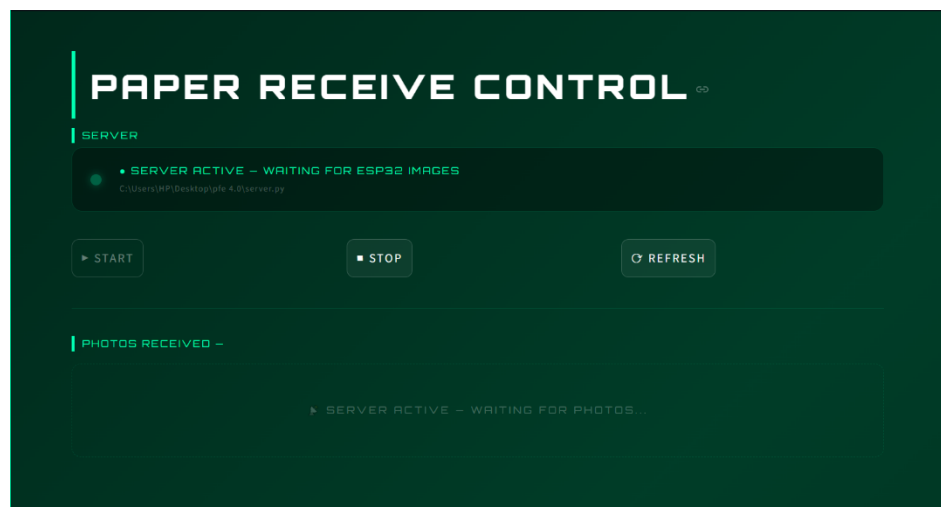


Figure IV-11 contrôleur de réception des copies après le démarrage du processus

Chapitre IV: Protocol test ANONYMOS OS

lorsque le surveillant numérise les fiches d'examen sur le terrain en utilisant le clic sur le bouton bostoire, alors les données binaires des images seront transférées au serveur en utilisant l'interface Wi-Fi de l'ESP32-S3-CAM. Le renouvellement de l'interface en cliquant sur le bouton REFRESH montre efficacement le résultat obtenu selon ce qui se passe comme illustré dans la Figure 4.12.

Il est visible qu'une série de captures d'écran (selon le fait dit que «17 PHOTO(S)») ont été obtenues. C'est sûr que le lieu de stockage des documents enregistrés est validé sur la mémoire locale (\...\pfe 4.0\Pics). En outre, on remarque que les données reçues sont présentées avec précision dans une collection d'images incluant visiblement les extraits des copies contenant des notations de pointage et un code-barres anonymisé.

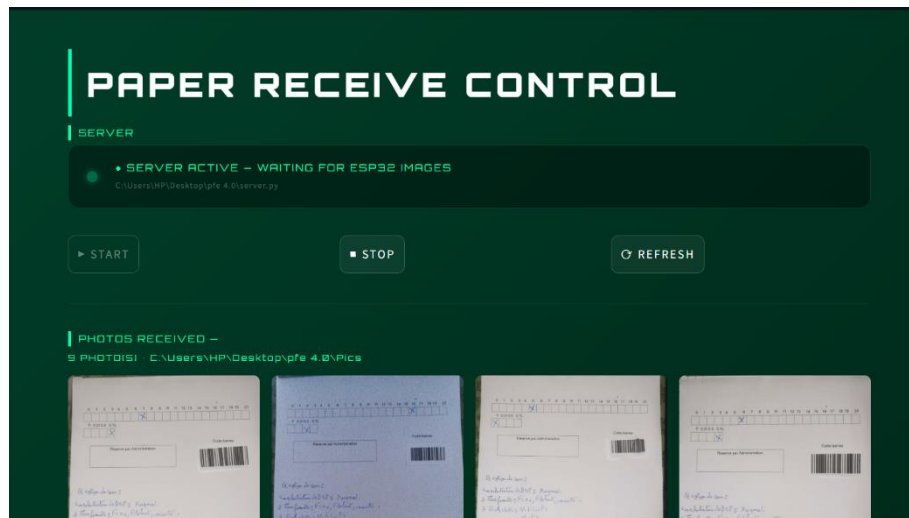


Figure IV-12 Interface de consultation des copies reçues.

La figure 4.13 ci-dessous donne l'exemple d'une image qui est transférée par Wi-Fi et CAM OV3660. Cette image est alors transférée aux autres parties du processus afin de compléter le traitement d'image dans la page d'exclusion de décodage.

Chapitre IV: Protocol test ANONYMOS OS

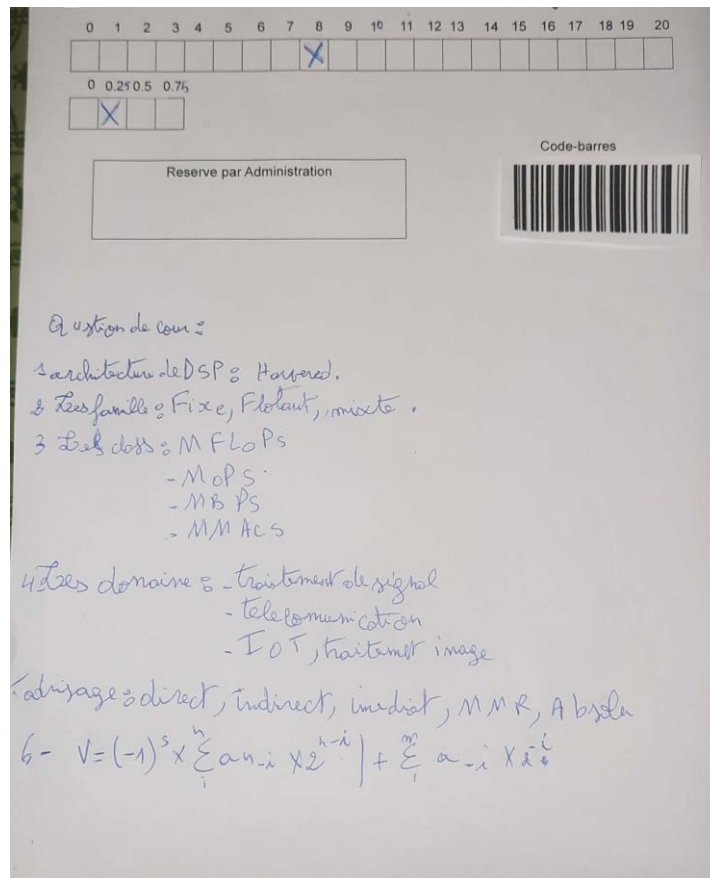


Figure IV-13 Image d'une copie d'examen à traiter

Il est important de souligner que le module développé ne se limite pas à la capture des images nécessaires à l'analyse des pixels et au décodage des codes-barres. Il assure également une fonction d'archivage numérique complète des copies d'examen. En effet, chaque image numérisée est automatiquement enregistrée dans un dossier sécurisé local (Figure 4.14), garantissant ainsi la conservation des documents originaux sous forme numérique.

Le décodage du code-barres présent sur chaque feuille d'examen permet d'identifier automatiquement les informations associées à l'évaluation, notamment l'identité de l'étudiant, la date de l'examen, le lieu de déroulement de l'épreuve ainsi que l'enseignant responsable du module. Ces informations sont directement inscrites sur la feuille d'examen scannée.

Après la numérisation complète de l'ensemble des copies, toutes les images enregistrées sont automatiquement regroupées dans un fichier PDF unique. Ce document constitue une version

Chapitre IV: Protocol test ANONYMOS OS

numérique fidèle des feuilles d'examen et fait partie intégrante des archives pédagogiques de l'établissement.

À l'issue du processus de correction et de calcul des notes, le système génère automatiquement une liste récapitulative des résultats des étudiants. Les copies numérisées ainsi que la liste des notes sont ensuite transmises par courrier électronique via le protocole SMTP à l'enseignant responsable du module. Cette procédure permet à l'enseignant de consulter rapidement les résultats, de vérifier les copies originales en cas de besoin et de disposer d'un archivage numérique sécurisé et durable de l'ensemble des évaluations.

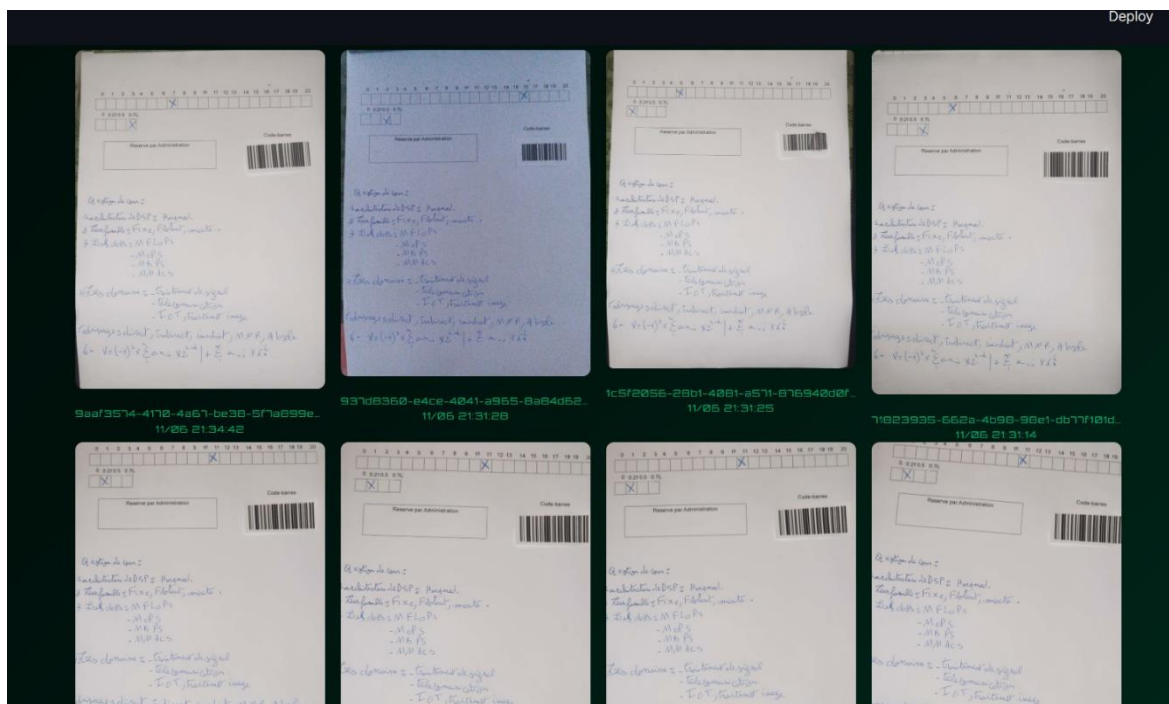


Figure IV-14 Interface de consultation des copies archivées

IV.6 Protocole d'exécution du module de traitement d'image, de décodage et d'extraction des notes automatique :

Après avoir reçu les images des fiches d'examen et les avoir stockées localement, l'application ANONYMOS OS passe à la phase qui lui est la plus cruciale, soit l'extraction automatique de ces images en tant qu'extraction automatique des notes par l'ordinateur et la levée de l'anonymat.

Chapitre IV: Protocol test ANONYMOS OS

Il faut préciser que avant de se lancer dans la lecture, il faut s'assurer que l'unité de traitement informatique est disponible. Cette unité est présentée dans la Figure 4.14. La sous-unité Decode est au repos et son composant TERMINAL · DECODE donne comme signe la phrase système NO OUTPUT YET. Ceci montre que les buffers d'entrée ont été initialisés et que ce sont des buffers capables d'intercepter le flux d'images dans le répertoire temporaire. L'opérateur appuie ensuite le bouton LAUNCH DECODE et le terminal Streamlit se transforme aussitôt et passe en mode COMPLETED (Figure 4.15).

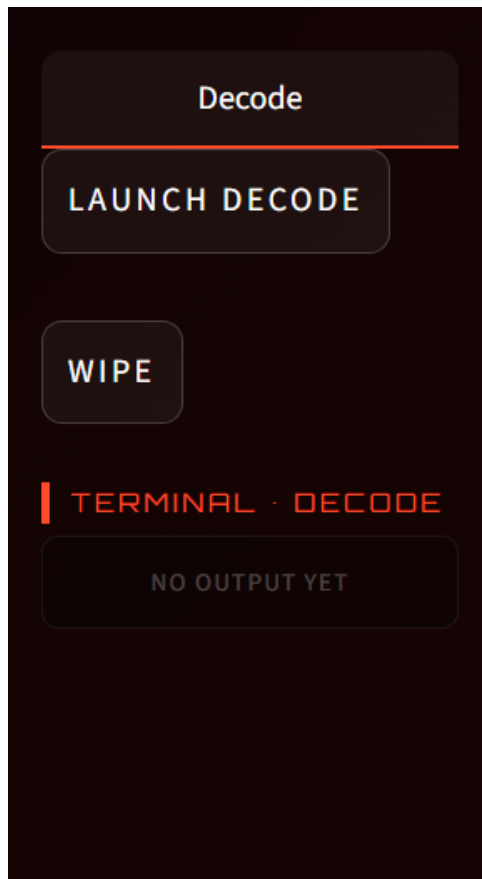


Figure IV-16 Interface avant le lancement du décodage.

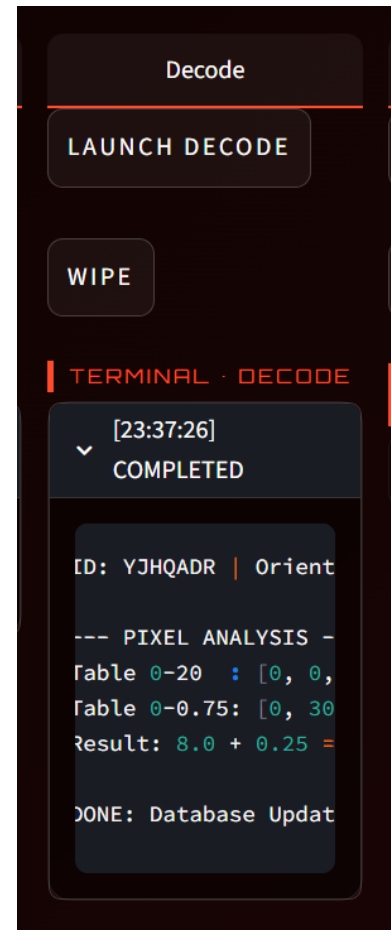
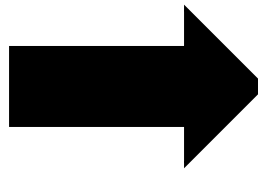


Figure IV-15 Interface après le décodage des copies d'examen.

- Lecture du Code-Barres et Orientation : Tout d'abord, la transformation de perspective permet de décrypter le code-barres contenant l'identifiant unique de la copie qui est dans notre cas YJHQADR. De plus, la variable orientation est aussi vérifiée afin de rectifier l'image dans le cas où l'orientation est inclinée.

Chapitre IV:Protocol test ANONYMOS OS

- Pixel Analysis : Dans un deuxième temps, le code utilise la fonction de segmentation par seuillage adaptatif afin de sélectionner géométriquement les deux grilles de notation présentes dans la capture. La preuve par le résultat affiché sur le terminal

Terminal :

```
YJHQADR | Orientation UP  
  
--- PIXEL ANALYSIS ---  
  
0-20 : [0, 0, 0, 0, 0, 0, 0, 0, 348, 12, 10, 7, 63, 11, 5, 55, 74, 0, 5, 9, 7]  
  
0-0.75 : [0, 302, 0, 0]  
  
:: 8.0 + 0.25 = 8.25  
  
Database Updated.
```

- **Ligne 0-20 (Note entière)** : L'algorithme mesure la densité de pixels noirs à l'intérieur de chaque une des 21 cases de la première grille. Les cases vides retournent tous des valeurs d'activation à zéro. La seule exception concerne la case de la note 8 où une forte densité de pixels noirs est visible, suite à la marque de croix à l'aide d'un stylo.
- **Ligne 0-0.75 (Note décimale)** : Un nouveau balayage est effectué à partir de la seconde grille de 4 cases. Dans ce cas, le système constate une forte densité de pixels noirs (dans la valeur d'activation de 302 pixels modifiés) située spécifiquement dans la deuxième case et correspondant à 0.25 décimal.

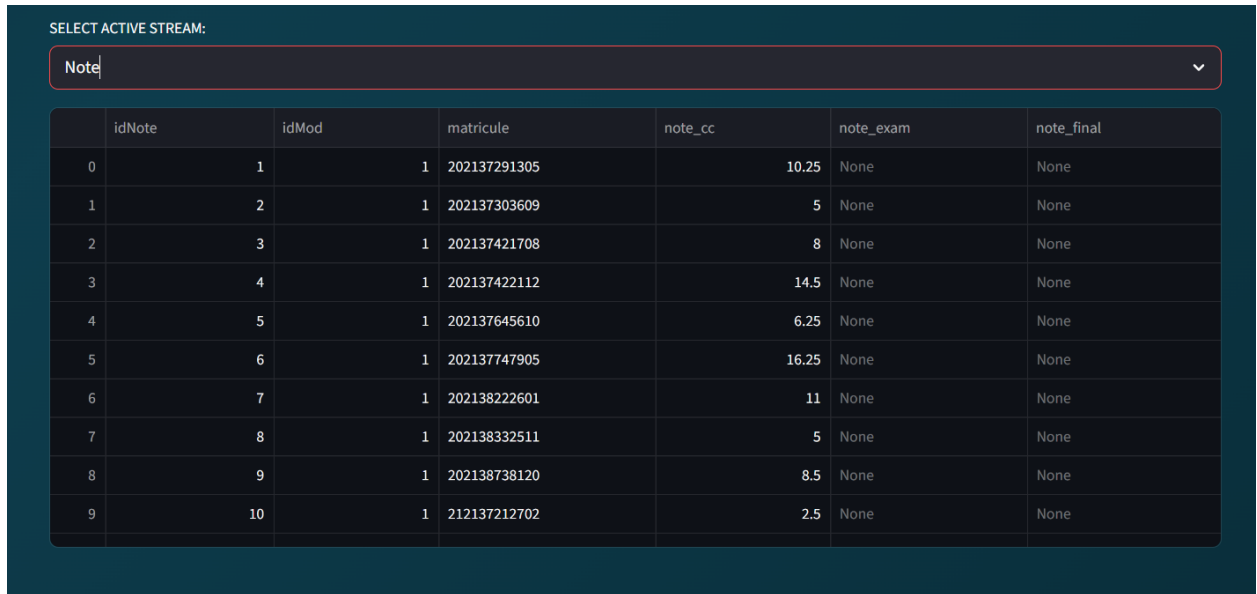
Les index activés sont traduits numériquement. Le terminal exécute l'équation finale : $8.0 + 0.25 = 8.25$. La note est ainsi extraite de manière purement mathématique.

Ensuite, l'algorithme recherche le code-barres correspondant au sein de la base de données, à savoir **JHQADR**, afin d'attribuer la note finale de **8,25**. Ce procédé, exécuté sur la grille de notation d'un **étudiant** pour une Module donnée, est répété de manière itérative pour l'ensemble des **étudiants** et pour toutes les Modules.

Chapitre IV:Protocol test ANONYMOS OS

IV.7 Protocole d'exécution du module de délibération et de levée d'anonymat

Tous les examens sont corrigés et les notes anonymisées sont saisies dans le système, ANONYMOS OS aborde sa dernière étape administrative : la délibération. Cette étape permet de fusionner les données des examens avec la base de données principale des étudiants afin de révéler leur identité et de calculer les moyennes pour finaliser les notes.



	idNote	idMod	matricule	note_cc	note_exam	note_final
0	1	1	202137291305	10.25	None	None
1	2	1	202137303609	5	None	None
2	3	1	202137421708	8	None	None
3	4	1	202137422112	14.5	None	None
4	5	1	202137645610	6.25	None	None
5	6	1	202137747905	16.25	None	None
6	7	1	202138222601	11	None	None
7	8	1	202138332511	5	None	None
8	9	1	202138738120	8.5	None	None
9	10	1	212137212702	2.5	None	None

Figure IV-17 Table des notes dans la base de données avant délibération.

Sur le tableau central d'administration Figure 4.17, l'unité logicielle Deliberation est en attente. Le composant TERMINAL · DELIBERATION indique le message standard NO OUTPUT YET, ce qui prouve qu'aucune requête de fusion de données n'est active.

Chapitre IV:Protocol test ANONYMOS OS

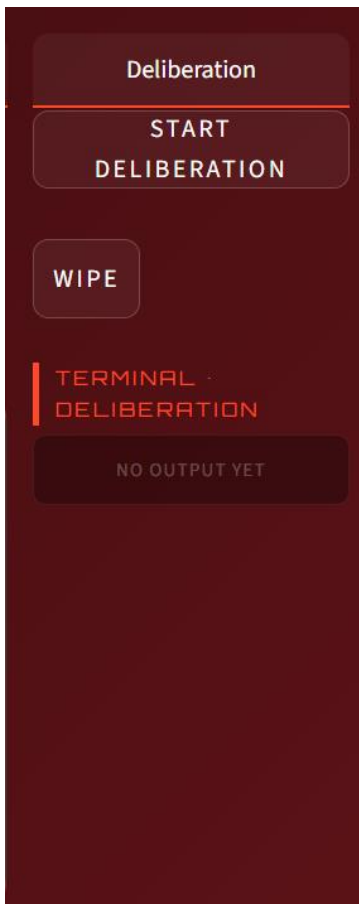


Figure IV-18 Interface avant le lancement de la délibération.

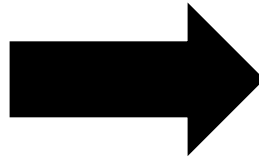


Figure IV-19 Interface après le lancement de la délibération.

L'opérateur déclenche le processus en appuyant sur le bouton START DELIBERATION. Le terminal Streamlit change de statut pour afficher la mention COMPLETED accompagnée de son horodatage avec le nombre de notes calculées comme l'illustre la Figure 4.18

La réussite de l'opération se vérifie immédiatement sur la page du Hub de données de la Figure 4.19 . En activant la DATA VIEWER, l'équipe pédagogique dispose d'un tableau récapitulatif complet et transparent.

La base de données est définitivement mise à jour : chaque ligne associe désormais de façon limpide le numéro de matricule, le nom et le prénom de l'étudiant à sa note d'examen extraite par la caméra (comme la note 8.25 calculée précédemment).

Comme nous l'avons mentionné dans les chapitres précédant, les points de contrôle continué ont été introduits manuel pour crée une simulation d'un cas réel. Ce procédé est simple, mais nécessaire à l'exécution de notre programme.

Chapitre IV:Protocol test ANONYMOS OS

SELECT ACTIVE STREAM:

Note

	idNote	idMod	matricule	note_cc	note_exam	note_final
15	16	1	212137316919	8.75	0	3.5
16	17	1	212137342705	6	0	2.4
17	18	1	212137393104	12	0	4.8
18	19	1	212137423406	5.25	0	2.1
19	20	1	212137423814	16.5	8.25	11.55
20	21	1	212137429224	11.5	11.25	11.35
21	22	1	212137439315	11.25	0	4.5
22	23	1	212137447413	2.25	0	0.9
23	24	1	212137450308	4.5	0	1.8
24	25	1	212137508713	10	0	4
25	26	1	212137539104	7.75	0	3.1

Figure IV-20 Table des notes dans la base de données après délibération

Remarque : le processus calcule tous les étudiants et toutes les Modules même si vous n’avez pas saisi de note d’examen du fait de l’absence d’un étudiant ou autres circonstances particulières.

IV.8 Protocole d’exécution du module d’exportation PDF par module et d’expédition automatisée (SEND RESULTS)

La conclusion logique et administrative de notre application ANONYMOS OS est la distribution automatisée et sécurisée des résultats académiques consolidés. Après avoir terminé la délibération globale, le système doit séparer les notes par module (Module), produire pour chaque module un procès-verbal officiel au format PDF, l’archiver localement, puis l’envoyer directement par courriel (via le protocole SMTP de Gmail) à l’enseignant responsable. Cette dernière phase opérationnelle suit un enchaînement technique, validé en quatre étapes.

Le protocole commence par la vérification de la console d'administration centrale montrée sur l'illustration 4.20 Le sous-module « SEND RESULTS » se trouve dans son état d’origine. Le terminal logiciel de surveillance (TERMINAL · ENVOYER DES NOTES) montre le message d'attente redondant AUCUNE SORTIE ENCORE, ce qui confirme qu'aucune file d'attente pour l'envoi d'e-mails ou la génération de rapports de secteur n'a encore été démarrée.

Chapitre IV: Protocol test ANONYMOS OS

L'opérateur appuie sur la commande INITIALIZE SEND RESULTS. Le terminal Streamlit passe immédiatement à l'état COMPLETED et affiche le log d'exécution des tâches en arrière-plan avec le gmail et le nom du module seccus qui a envoyé le message, visible sur l'illustration 4.21.

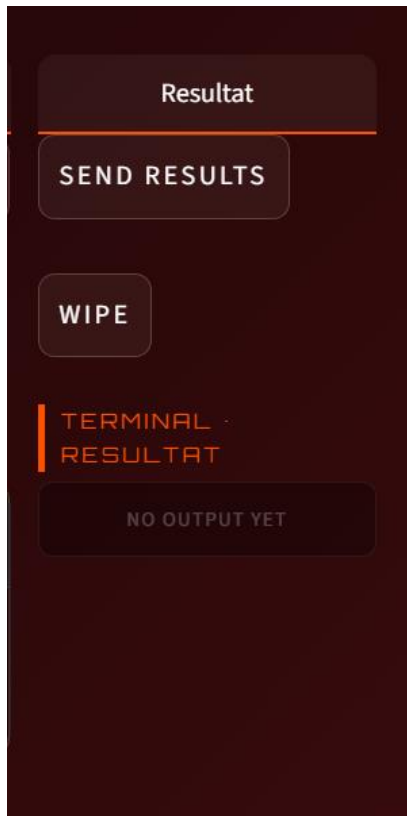


Figure IV-22 Interface avant l'envoi des résultats.

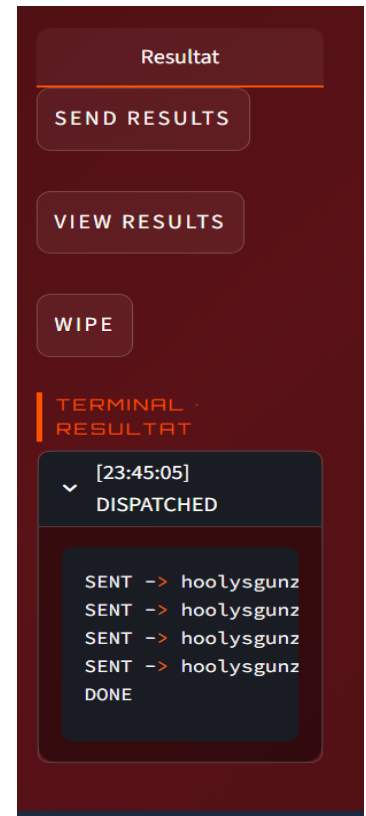
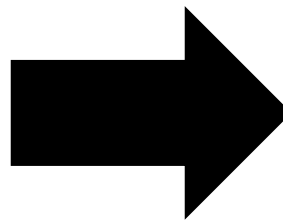


Figure IV-21 Interface après l'envoi des résultats.

Nous validons l'instruction de sauvegarde, nous inspectons l'arborescence des fichiers du projet. Comme le montre la Figure 4.22, l'algorithme a créé de manière autonome un sous-dossier de stockage dédié. À l'intérieur de ce dossier, on trouve le fichier cible (fichier PDF correspondant) dont le nom correspond exactement à la nomenclature des Modules avec son pdf généré.

Chapitre IV:Protocol test ANONYMOS OS

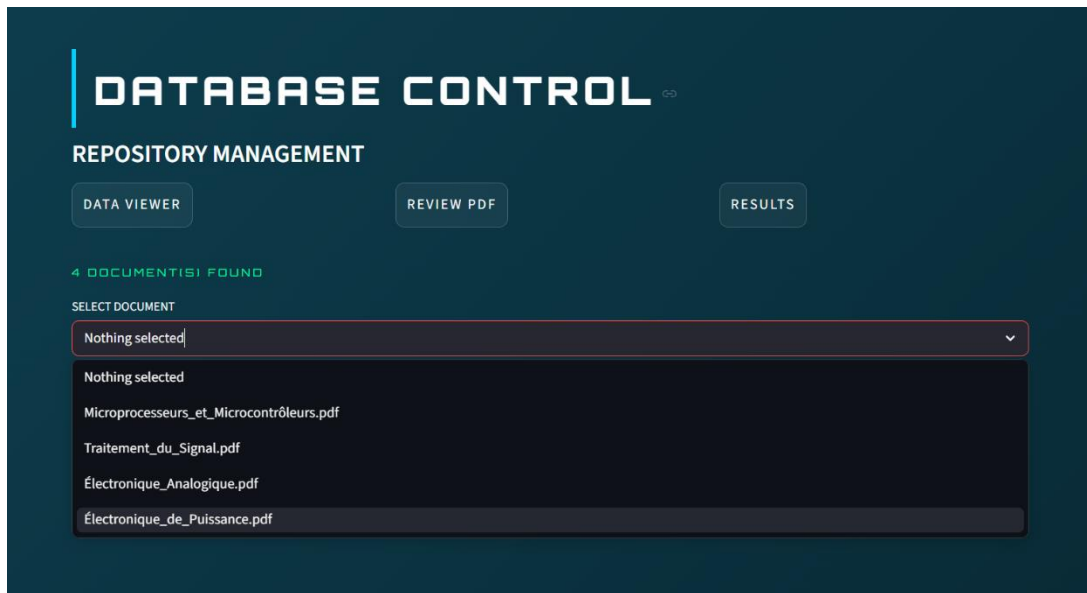


Figure IV-23 Liste des fichiers PDF des modules archivés.

Vous pouvez à tout moment vérifier la validité du fichier soumis en consultant tous les fichiers PDF soumis manuellement sur la page de données.

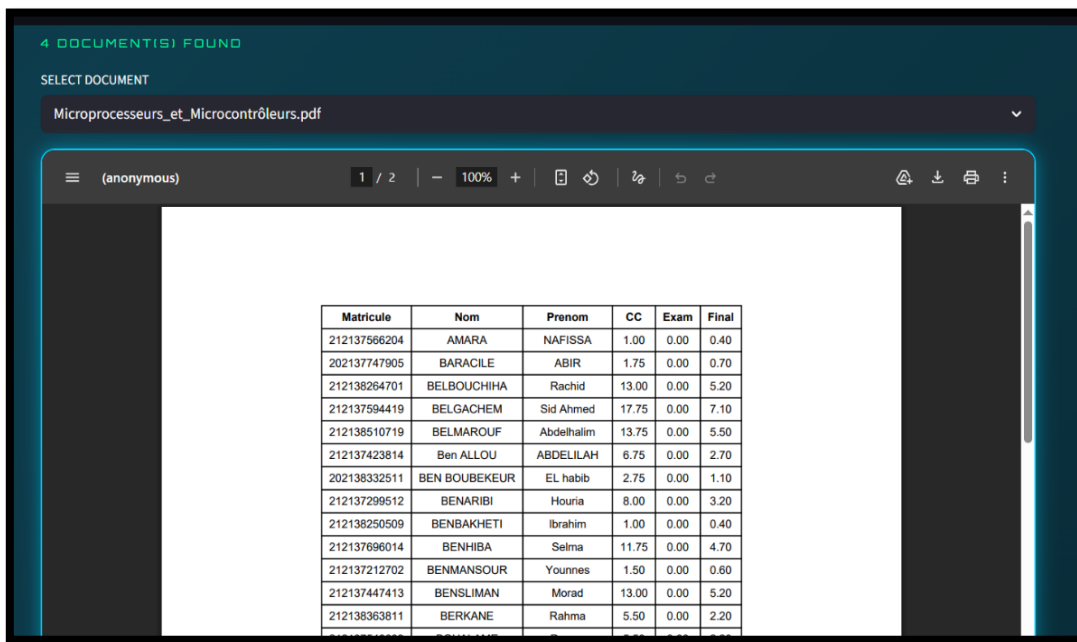


Figure IV-24 fichier PDF généré après sélection du module

En ouvrant le message, on observe comme le montre la Figure 4.24, un texte simple, clair et professionnel destiné à l'enseignant : « Bonjour, veuillez trouver ci-joint les notes d'examen pour le module Traitement du Signal. Bien à vous, L'administration ». La pièce .pdf y est correctement jointe.

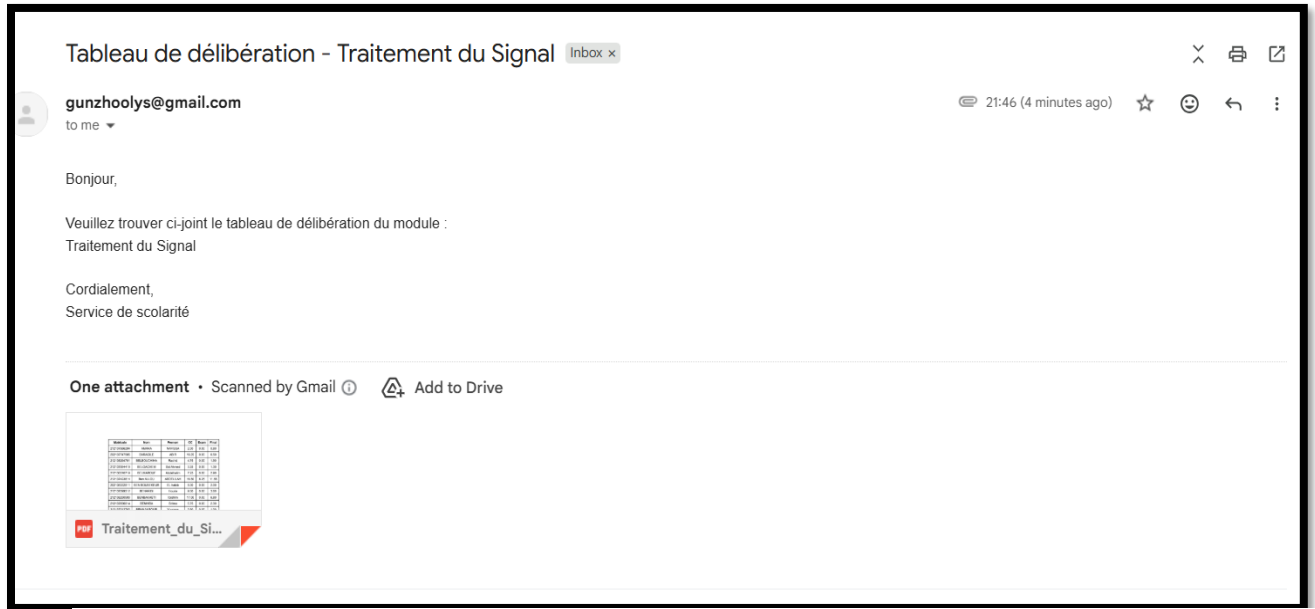


Figure IV-25 Courriel envoyé au professeur via Gmail

IV.9 Perspectives d'évolution et futurs développements d'ANONYMOS OS

Bien que l'application ANONYMOS OS réponde rigoureusement au cahier des charges initial en orchestrant avec succès tout le cycle allant de la sécurisation des accès à la distribution automatisée des notes, plusieurs axes d'amélioration technologique peuvent être projetés pour optimiser ses performances à grande échelle.

IV.9.1 Migration vers le Cloud et architecture relationnelle multi-utilisateurs

La première perspective de développement consiste à faire évoluer l'infrastructure logicielle d'une architecture locale basée sur SQLite vers une solution Cloud centralisée, animée par un système de gestion de base de données relationnelle plus robuste comme PostgreSQL. Ce passage au nuage permettrait de gérer des connexions simultanées provenant de multiples départements ou facultés universitaires de manière fluide. Par ailleurs, cette transition logicielle permettrait d'introduire un contrôle d'accès strict basé sur les rôles afin de compartimenter et de sécuriser les espaces de travail des administrateurs, des correcteurs et des agents de scolarité via une authentification forte à double facteur.

Chapitre IV:Protocol test ANONYMOS OS

IV.9.2 Optimisation du traitement d'image par réseaux de neurones convolutifs et numérisation industrielle

Le cœur algorithmique du système gagnerait en flexibilité, en vitesse et en robustesse en faisant évoluer la méthode d'acquisition matérielle. Au lieu d'une capture manuelle et unitaire via le module ESP32-S3-CAM, perspective est faite d'intégrer une machine de numérisation industrielle entièrement automatique équipée d'un chargeur de documents à alimentation continue. Cette station permettrait à l'opérateur de déposer l'intégralité du paquet de fiches d'examen en une seule fois. Le scanner autonome avalerait et numériserait à haute vitesse chaque copie sous un éclairage parfaitement contrôlé, éliminant ainsi définitivement les problèmes d'inclinaison, de flou de bougé ou de variations lumineuses ambiantes.

IV.9.3 Amélioration du terminal IoT et standardisation de l'acquisition visuelle

Sur le plan matériel, l'optimisation du terminal de capture basé sur l'ESP32-S3-CAM s'orienterait vers la conception d'un boîtier de numérisation dédié en impression 3D, équipé d'un système d'éclairage LED annulaire constant. Cette structure permettrait de standardiser l'acquisition visuelle sur le terrain en éliminant les variations de lumière ambiante des salles de correction. De plus, l'intégration des protocoles de sécurité Wi-Fi d'entreprise permettrait une intégration transparente et hautement sécurisée de ces modules IoT sur les réseaux sans fil institutionnels des universités.

IV.9.4 Intégration d'une extension mobile multiplateforme pour les usagers

Enfin, l'écosystème global d'ANONYMOS OS pourrait s'enrichir d'une application mobile native destinée à étendre les services de la plateforme. Cette extension offrirait aux enseignants la possibilité d'utiliser directement l'appareil photo de leur smartphone pour numériser et téléverser les grilles de notation en toute mobilité. En parallèle, les étudiants bénéficieraient d'un portail utilisateur personnalisé pour consulter instantanément leurs résultats académiques et télécharger leurs procès-verbaux de délibération dès l'activation du module d'envoi automatique par l'administration.

IV.10 Conclusion :

Le travail présenté dans ce mémoire s'inscrit dans le cadre de la modernisation, de la numérisation et de la sécurisation des processus administratifs au sein des établissements d'enseignement supérieur. À travers la conception et le développement de la plateforme intelligente **ANONYMOS OS**, nous avons proposé une solution logicielle innovante, automatisée et sécurisée à l'un des défis les plus délicats de la gestion pédagogique : la préservation absolue de l'anonymat des examens ainsi que la traçabilité des notes de la copie jusqu'au procès-verbal de délibération.

Au cours de ce projet, nous avons relevé le défi d'interagir plusieurs disciplines de l'ingénierie informatique et électronique au sein d'une même solution. La mise en œuvre - du système repose sur un flux de traitement cohérent et robuste. Celui-ci débute par la génération automatisée de documents d'examens comportant des identifiants anonymes sous forme de codes-barres. Par la suite, l'intégration des techniques de vision par ordinateur et du traitement d'image a permis de transformer une simple acquisition visuelle des grilles de notation en données numériques fiables, grâce à l'analyse de la densité des pixels pour en extraire la note arithmétique exacte sans tout en réduisant considérablement les interventions manuelles et les risques d'erreur, le système est capable d'extraire automatiquement les notes avec précision,

Par ailleurs, la force de la plateforme **ANONYMOS OS** réside dans la couche logicielle backend centralisée assurant la gestion et la sécurisation des informations au sein d'une base de données relationnelle. Cette architecture permet non seulement de garantir la traçabilité des informations tout au long du processus, mais également d'assurer la levée de l'anonymat lors des délibérations et la génération automatique des procès-verbaux au format PDF, accompagnée de leur transmission électronique aux enseignants responsables.

Cependant, la réalisation de ce système a également mis évidence plusieurs défis techniques importants, notamment lors de la phase d'acquisition d'images. L'intégration du module ESP32-S3-CAM s'est révélée plus complexe que prévu en raison de certaines contraintes matérielles et logicielles. Ce composant s'est révélé particulièrement complexe à configurer et à stabiliser par rapport à un microcontrôleur ESP32 standard. La gestion de son architecture de mémoire restreinte (PSRAM), les contraintes de bande passante pour le flux vidéo, ainsi que sa sensibilité aux perturbations thermiques et lumineuses ont rendu son intégration bien plus laborieuse que

Chapitre IV:Protocol test ANONYMOS OS

prévu pour un usage fluide et grand public. La gestion de la mémoire embarquée (PSRAM), les limitations liées à la bande passante nécessaire au transfert des images, ainsi que la sensibilité du module aux conditions d'éclairage et aux variations thermiques ont constitué des obstacles techniques importants. Malgré ces difficultés, les objectifs fixés ont été atteints et le système a démontré sa capacité à fonctionner de manière fiable dans un contexte réel d'utilisation.

Sur le plan personnel et professionnel, Ce projet de fin d'études a représenté une expérience particulièrement enrichissante. Il nous permis de consolider nos compétences dans plusieurs domaines, notamment la programmation en Python, en gestion de bases de données et en développement d'interfaces web dynamiques avec le framework Streamlit. De plus, l'intégration pratique de concepts liés à l'Internet des Objets (IoT) et au traitement d'images appliqués à des problématiques concrètes m'a donné une vision globale de l'ingénierie des logiciels interconnectés.

Enfin, ce projet constitue une base solide pour de futures évolutions.. Les perspectives d'évolution envisagées figurent la migration vers une architecture Cloud sécurisée multi-utilisateurs, l'intégration de scanners professionnels à alimentation automatique afin d'améliorer les performances de numérisation, ainsi que l'exploitation de techniques avancées d'intelligence artificielle, notamment des réseaux de neurones convolutifs, pour optimiser davantage la reconnaissance et l'analyse des données. À terme, **ANONYMOS OS** ambitionne de devenir un outil de référence pour la gestion des examens universitaires, en garantissant aux établissements d'enseignement supérieur davantage de transparence totale, un gain de temps considérable et de sécurité inviolable pour le traitement des évaluations académiques.

Références

[1] HAL Open Science, “Scientific Theses and Research Repository – Computer Science and Engineering,” [Online].

Available: <https://theses.hal.science/search>

[2] TechTarget, “Software Engineering Overview and Definitions,” TechTarget Learning Hub, [Online].

<https://www.techtarget.com/searchsoftwarequality/definition/software-engineering>

[3] GeeksforGeeks, « Advantages and Disadvantages of Embedded System », GeeksforGeeks, [En ligne]. Disponible sur : <https://www.geeksforgeeks.org/digital-logic/advantages-and-disadvantages-of-embedded-system/>.

[4] GenX TechY, « Embedded Systems: Application Areas », GenX TechY, [En ligne]. Disponible sur : <https://genxtechy.com/embedded-systems-application-areas/>.

[5] CONTEC Americas, « Solutions embarquées et industrielles », CONTEC, [En ligne]. Disponible sur : <https://us.contec.com/>.

[6] Espressif Systems, *ESP32-S3 Series Datasheet*, Espressif Systems, [En ligne]. Disponible sur : https://www.espressif.com/sites/default/files/documentation/esp32-s3_datasheet_en.pdf

[7] Zephyr Project, « Documentation officielle du système d'exploitation Zephyr », Zephyr Project, [En ligne]. Disponible sur : <https://docs.zephyrproject.org/>.

[8] Arduino, « Arduino Uno Rev3 — Technical Specifications », Arduino, [En ligne]. Disponible sur : <https://store.arduino.cc/products/arduino-uno-rev3>.

[9] OmniVision Technologies, *OV3660 Camera Sensor Datasheet*, OmniVision, [En ligne]. Disponible sur : <https://www.ovt.com/>.

[10] Espressif Systems, « ESP-IDF Programming Guide — ESP32-S3 Camera and Peripherals Documentation », Espressif Systems, [En ligne]. Disponible sur : <https://docs.espressif.com/projects/esp-idf/>.

[11] Encyclopaedia Universalis, « Code-barres », Encyclopaedia Universalis, [En ligne]. Disponible sur : <https://www.universalis.fr/>.

[12] Le Modèle Informatique, « Articles et ressources sur les systèmes informatiques », [En ligne]. Disponible sur : <https://lemodeleinformatique.fr/>.

[13] Oracle NetSuite, « Enterprise Resource Planning (ERP) Solutions », Oracle, [En ligne]. Disponible sur : <https://www.netsuite.com/>.

[14] Barcodes Inc., Barcodes Inc, “Barcode Technology Overview and Types of Barcodes,” [Online]. Available: <https://www.barcodesinc.com/articles/what-is-a-barcode.htm>

[15] DOI Foundation, « Digital Object Identifier System », DOI Foundation, [En ligne]. Disponible sur : <https://www.doi.org/>.

[16] Digital Commons, « Research Repository », University of Nebraska–Lincoln, [En ligne]. Disponible sur : <https://digitalcommons.unl.edu/>.

[17] Cognex, « Types de codes-barres », Cognex, [En ligne]. Disponible sur : <https://www.cognex.com/fr/tools-and-resources/resource-center/what-is-a-barcode/types-of-barcodes>.

[18] R. Singh et M. Sharma, « Barcode Symbology and Its Applications in Modern Libraries », *Library Philosophy and Practice*, 2007, référencé dans Dass & Singh, 2011.

[19] ISO/IEC 15420:2009, *Information technology — Automatic identification and data capture techniques — EAN/UPC bar code symbology specification*, Organisation internationale de normalisation, 2009. Disponible sur : <https://www.iso.org/standard/46143.html>.

[20] GS1 International, « The GS1 System of Standards », GS1, 2024. Disponible sur : <https://www.gs1.org>.

[21] ISO/IEC 15438:2015, *Information technology — Automatic identification and data capture techniques — PDF417 bar code symbology specification*, Organisation internationale de normalisation, 2015.

[22] M. Hara / Denso Wave Inc., « QR Code System », Denso Wave, 1994. Disponible sur : <https://www.qrcode.com/en/history/>.

[23] B. Sikka Bajaj, « Reliability on QR Codes and Reed-Solomon Codes », *arXiv*, arXiv:2407.17364, 2024. Disponible sur : <https://arxiv.org/abs/2407.17364>.

[24] P. Nagpal *et al.*, « Barcode Recognition Techniques: Review & Application », *International Journal of Innovative Research in Computer Science & Technology (IJIRCST)*, 2016. Disponible sur : <https://www.ijircst.org>.

[25] Techniques de l'Ingénieur, « Technologies d'identification et d'authentification pour un système de traçabilité », *Techniques de l'Ingénieur*, réf. TR640. Disponible sur : <https://www.techniques-ingenieur.fr/base-documentaire/genie-industriel-th6/systemes-d-informations-et-tic-pour-la-tracabilite-42117210/technologies-d-identification-et-d-authentification-pour-un-systeme-de-tracabilite-tr640/>.

[26] Techniques de l'Ingénieur, « Base documentaire scientifique et technique de référence », [En ligne]. Disponible sur : <https://www.techniques-ingenieur.fr/>.

[27] SPIE, « Ressources scientifiques en optique, photonique et imagerie », Society of Photo-Optical Instrumentation Engineers, [En ligne]. Disponible sur : <https://www.spie.org/>.

[28] E. R. Fossum, « CMOS Image Sensors: Electronic Camera-On-A-Chip », *IEEE Transactions on Electron Devices*, [En ligne]. Disponible sur : <https://ieeexplore.ieee.org/document/551910>. Consulté le : 15/05/2025.

[29] KEYENCE America, « Basics of Lighting Selection in Machine Vision Inspection », KEYENCE, [En ligne]. Disponible sur : <https://www.keyence.com/products/vision/vision-sys/resources>.

[30] Thèses.fr, « Algorithmes, architecture et éléments optiques pour l'acquisition embarquée d'images totalement focalisées », Thèse de doctorat, Institut Mines-Télécom Atlantique, 2017. Disponible sur : <https://theses.fr/2017IMTA0055>.

[31] e-con Systems, « What is Signal-to-Noise Ratio (SNR)? Why is SNR important in embedded cameras? », *Edge AI and Vision Alliance*, 2023. Disponible sur : <https://www.edge-ai-vision.com/2023/01>.

[32] ICO Optics, « Signal-to-Noise Ratio (SNR) in Low-Light Imaging: Key Principles and Enhancement Strategies », ICO Optics, 2025. Disponible sur : <https://www.ico-optics.org/signal-to-noise-ratio-snr-in-low-light-imaging/>.

[33] PMC/NCBI, « 1D Barcode Detection: Novel Benchmark Datasets and Comprehensive Comparison of Deep Convolutional Neural Network Approaches », *PMC*, 2022. Disponible sur : <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC9696533/>.

[34] ResearchGate, « Real-Time Barcode Detection and Classification using Deep Learning », ResearchGate, 2017. Disponible sur : <https://www.researchgate.net/publication/321065123>.

[35] Nature / Scientific Reports, « Deep Learning Framework for Barcode Localization and Decoding using Simulated UAV Imagery », *Scientific Reports*, 2025. Disponible sur : <https://www.nature.com/articles/s41598-025-29720-w>.

[36] IJAIDSML, « Deep Learning for Industrial Barcode Recognition at High Throughput », *IJAIDSML*, 2024. Disponible sur : <https://ijaidsm.org/index.php/ijaidsm/article/download/271/248>.

[37] GS1 International, « GS1 Global Standards », GS1, [En ligne]. Disponible sur : <https://www.gs1.org>.

[38] Ministère du Commerce, « Réglementation sur la codification des produits », Ministère du Commerce, Algérie, [En ligne].

[39] La Génération Active, « Codes-barres ou QR codes : quelle technologie d'identification choisir ? », *La Génération Active*, 2025. Disponible sur : <https://www.lagenerationactive.fr/codes-barres-qr-codes-technologie-identification/>.

[40] BARCODIS, « Ressources et solutions codes-barres », BARCODIS, [En ligne]. Disponible sur : <https://www.barcodis.com/>.

[42] Higher Education Quality Management: Evidence from Adrar University, AESS Web <https://archive.aessweb.com/index.php/5009/article/view/304>

[42] The application of total quality management in higher education in Algeria, *Revue d'Economie et de Management* <https://journals.univ-tlemcen.dz/REM/index.php/REM/fr/article/view/264>

[43] Level of Quality Education Services and the Effect on Students' Satisfaction: Saida University – Algeria, <https://journals.ust.edu/index.php/AJQAHE/article/view/1190>

[44] Pressman, Roger – *Software Engineering: A Practitioner's Approach*
https://www.tutorialspoint.com/software_engineering/index.htm

ANNEXE

Core de programme

- **Initialisation et Génération des Codes-Barres Anonymes**

```
message = f'{idExem}-{idSch}-{idMod}-{matricule}-{idRoom}-{idEns}'.encode()

digest = hmac.new(secret_key, message, hashlib.sha256).digest()

code_value = base64.b32encode(digest).decode('utf-8').rstrip("=")[:7]

barcode_img = CODE128(code_value, writer=ImageWriter())

cur.execute(

    "UPDATE Exam SET codeBar=?, codeText=? WHERE idExem=?",

    (sqlite3.Binary(img_data), code_value, idExem))
```

Ce programme permet une initialisation des codes-barres anonymes par génération d'un identifiant unique et sécurisé qui est obtenu à partir de l'information concernant l'étudiant et l'examen en utilisant l'algorithme cryptographique HMAC-SHA256. L'identifiant ainsi obtenu est transformé en un code-barres dans le format Code 128 et est sauvegardé dans la base de données au format texte et image.

- **La génération automatique des fichiers PDF**

```
for idMod, moduleName, ensName, roomName, day_debut, startTime, endTime in modules:

    cur.execute("""

        SELECT s.matricule, s.firstName, s.lastName, e.codeBar

        FROM Exam e

        JOIN Student s ON e.matricule = s.matricule

        WHERE e.idMod = ?

    """, (idMod,))

    rows = cur.fetchall()

    for matricule, firstName, lastName, img_blob in rows:

        buffer = BytesIO(img_blob)

        barcode_img = Image(buffer, width=110, height=40)

        table_data.append([

            f'{lastName} {firstName}',

            matricule,barcode_img])
```

Ce programme permet l'automatisation de la création des documents PDF. Il s'occupe de collecter toutes les informations contenues dans les modules, les étudiants ainsi que les codes-barres anonymes stockés dans la base de données. Il structure ensuite toutes ces informations par tableaux avec toutes les informations nécessaires pour passer l'examen.

- **Lecture Automatique des Copies et Attribution des Notes**

```
bc = self.get_barcode(img)

student_code = bc.data.decode('utf-8')

rects = self.find_tables(img)

binary_ink = self.make_ink_mask(img)

val_int, _ = self.analyze_cells(binary_ink, rects[0], self.integer_mapping)

val_dec, _ = self.analyze_cells(binary_ink, rects[1], self.decimal_mapping)

total_grade = val_int + val_dec

cursor.execute(

    "UPDATE exam SET note_exam = ? WHERE codeText = ?",

    (total_grade, student_code))

conn.commit()
```

Le logiciel s'occupe de la lecture ainsi que de la vérification des copies anonymes. Le logiciel commence par scanner le code-barre de manière à anonymiser l'identité de la copie et analyse ensuite les cases marquées par l'enseignant pour donner la note. Suite au calcul de la note, celle-ci est automatiquement inscrite dans la base de données sous l'étudiant concerné.

- **calcul Automatique des Notes Finales et Délibération**

```
note_final = round((note_cc * 0.4) + (note_exam * 0.6), 2)

cursor.execute("""

    UPDATE note

    SET note_cc = ?, note_exam = ?, note_final = ?

    WHERE idMod = ? AND matricule = ?

    """, (note_cc, note_exam, note_final, idMod, matricule))
```

Ce programme effectue de manière automatique le calcul des notes de chaque élève à partir des notes de contrôle continue et d'examen avec le coefficient défini par l'école. Lorsque le calcul

est fait, les scores sont automatiquement sauves dans la base de données, ce qui facilite le processus de délibération.

- **Génération Des Pdf et Envoi les aux enseignants**

```
students = cur.fetchall()

build_pdf(module_name, students, pdf_path)

send_email(

    teacher_email,

    pdf_path,

    module_name
```

Cet outil génère automatiquement les tableaux et les feuilles de délibération à partir des données stockées dans la base de données. Pour chaque module, un fichier PDF contenant les notes des étudiants est créé. Ce document est ensuite envoyé automatiquement par courrier électronique à l'enseignant responsable du module. Cette automatisation facilite le processus de délibération, améliore la fiabilité des résultats et réduit les tâches administratives.

- **Acquisition et Réception des Copies d'Examen Numérisées**

```
@app.route("/upload", methods=["POST"])

def upload():

    file = request.files["file"]

    source = request.form.get("source", "esp32")

    save_path = os.path.join(SAVE_DIR, filename)

    if source == "phone": img = Image.open(file.stream)

        img = ImageOps.exif_transpose(img)

        img.convert("RGB").save(save_path, "JPEG")

    else:

        file.save(save_path)

    return jsonify({"status": "success"})
```

Ce programme assure la réception et l'enregistrement des copies d'examen numérisées envoyées par un smartphone ou une caméra ESP32-CAM.

Business Model Canvas (modèle d'affaire)

نموذج العمل التجاري

فكرتي هي تصميم وتنفيذ منصة ذكية لإدارة الامتحانات: إخفاء الهوية، وإمكانية التتبع، والمداولات الآلية.

الشراكات الرئيسية	الأنشطة الرئيسية	القيمة المقترحة	العلاقات مع العملاء	شرائح العملاء
- الشركات ومجلات الجملة المتخصصة في توفير الميكروكترولر، لوحات التطوير، والقطع الإلكترونية. - وزارة المؤسسات الناشئة الشريك الاستراتيجي الأول الذي يوفر الدعم القانوني. - الشركات التي توفر خوادم التخزين المحلية وأجهزة طباعة الباركود عالية الدقة والسرعة - وزارة التعليم العالي والبحث العلمي والجامعات: كشريك وشريك تجريبي أول - مقدمو خدمات الاستضافة السحابية الآمنة	- تطوير وتحديث البرمجيات وخوارزميات الذكاء الاصطناعي - تصميم وإدارة قواعد البيانات الآمنة - ضمان الجودة، الحماية، والأمن السيبراني - الإنتاج الفعلي للنظام من خلال برمجة وتركيب محطات التصوير الرقمي، تهيئة الخوادم واختبار كفاءة وسرعة المنصة	- الشفافية المطلقة والمصادقية: يضمن النظام فصل الهوية للمرشح تماماً عن ورقة الإجابة من خلال تشفيرها، - المداولات واستخراج النتائج بالذكاء الاصطناعي: استخدام ذكاء اصطناعي لتقليل المتاعب الإدارية. وتقليل من أخطاء يدوية. - نظام تتبع رقمي متكامل إمكانية مراقبة وتتبع مسار ورقة الامتحان. - الأتمتة الكاملة (الآلية) في استخراج النتائج: يتم إلغاء التدخل البشري والتقليدي في رصد العلامات.	- الدعم الفني المباشر خلال فترات الامتحانات: - بناء علاقة تعاقدية تضمن للمؤسسات التعليمية صيانة دورية للنظام - تقديم خدمة مخصصة لكل جامعة أو مركز امتحان حسب الطلب - تنظيم ورشات تدريبية وتكوينية - بناء الثقة مع الإدارة والأساتذة من خلال ضمان نسخ احتياطي فوري	- العملاء الأساسيون - مؤسسات التعليم العالي والجامعات (الكليات، الأقسام، ومكاتب التمدرس) - الديوان الوطني لامتحانات والمسابقات (ONEC) - العملاء المحتملون - المدارس والمعاهد التكوينية الخاصة - المؤسسات والشركات الكبرى (قطاع التوظيف)
	- الموارد الرئيسية - وحدات المعالجة والتقاط الصور - الخوادم المحلية أو السحابية المستعملة لاستقبال وتخزين صور أوراق الامتحانات - الكفاءات التقنية المسؤولة عن تصميم قواعد البيانات، وإدارة النظام - خوارزميات الرؤية الحاسوبية (Computer Vision)		- قنوات - تنظيم زيارات رسمية لإدارات الجامعات، الكليات، ومسؤولي قطاع (B2B Sales) - موقع محترف مخصص لعرض الخصائص التقنية للمنصة - استغلال التواجد داخل الحاضنات الجامعية والمشاركة في المعارض الوطنية - تخصيص قنوات تواصل مباشرة (البريد الإلكتروني و هاتف)	

مصادر الإيرادات

- عوائد مالية متكررة ناتجة عن بيع تراخيص استخدام منصة للجامعات والكليات
- عقود صيانة دورية سنوية تضمن للمؤسسات الحصول على دعم فني مباشر وفوري خلال الامتحانات
- رسوم التخصيص والتطوير حسب الطلب بالنظر الي حجم البيانات المدروسة
- بيع وتهيئة محطات الاستقبال والتصوير الرقمي المعتمدة على الكاميرات والأنظمة المدمجة

هيكل التكاليف

- رواتب فريق العمل من مبرمجي، مهندسي الأنظمة، وخبراء قواعد البيانات المسؤولين عن بناء المنصة
- مصاريف حجز وإدارة الخوادم وقواعد البيانات السحابية
- تكلفة اقتناء الميكروكترولر، لوحات التطوير، والقطع الإلكترونية
- مصاريف التنقل للجامعات ومراكز الامتحانات والمعارض للتسويقية

Business Plan



à intégrer ↓
Durée d'amortissement 5 années

à intégrer ↓	à intégrer ↓	à intégrer ↓	à intégrer ↓	à intégrer ↓	à intégrer ↓	à intégrer ↓	à intégrer ↓
Désignation du matériel	Fonctionnalité du matériel	Prix unitaire	Année 01	Année 02	Année 03	Année 04	Année 05
N°01 ESP32-S3-CAM Kits	Exam sheet capture	4,000	200,000	200,000	160,000	0	0
N°02 LCD 2004A	Affiche Message	1,500	75,000	75,000	60,000	0	0
N°03 Network Equipment	Connectivity	150,000	150,000	220,000	220,000	0	0
N°04 Cloud/Hosting Setup	Deployment	100,000	100,000	140,000	140,000	0	0
N°06 Development Workstations	Software development	300,000	300,000	0	0	0	0
N°07 Central Server PC	Processing & OCR	250,000	250,000	0	0	0	0

Tableau 2 Investissement

		à intégrer ↓	à intégrer ↓	Calcul automatique	Calcul automatique	Mensuel à intégrer ↓	Calcul automatique	Calcul automatique	Calcul automatique	Calcul automatique
		Salaire Base Mensuel	Indemnité Mensuel	Retenue Sécurité	Retenue Sécurité	Prime Panier & Transport	Salaire imposable	IRG	Salaire Net Mensuel	Salaire chargé Annuel
N°01		0	0	0	0	0	0	0	0	0
N°02	Secrétaire	35,000	5000	3,150	10,400	5,000	41,850	2,070	39,780	689,640
N°04	délegue	45,000	8000	4,050	13,780	5,000	53,950	4,450	49,500	914,760
N°05	ingenieurs	90,000	15000	8,100	27,300	10,000	106,900	16,900	90,000	1,910,400
N°06	techniciens	50,000	8000	4,500	15,080	10,000	63,500	5,800	57,700	1,066,560
N°07		0	0	0	0	0	0	0	0	0

Tableau 3 Masse salariale

Chiffre d'affaires			FY22	FY23	FY24	FY25	FY26	FY27
			N-1 (historique)	Année 01	Année 02	Année 03	Année 04	Année 05
Produit/Service N°01	Désignation:	installation						
	Quantité:	165		18	25	32	40	50
	Prix unitaire:	170000		170000	170000	170000	170000	170000
	Sous-Total:		0	3060000	4250000	5440000	6800000	8500000
Produit/Service N°02	Désignation:	maintenance/mise a jour						
	Quantité:	247		33	50	40	54	70
	Prix unitaire:	2000		2000	2000	2000	2000	2000
	Sous-Total:		0	66000	100000	80000	108000	140000
Produit/Service N°03	Désignation:	formation						
	Quantité:	50		10	10	10	10	10
	Prix unitaire:	8000		8000	8000	8000	8000	8000
	Sous-Total:		0	80000	80000	80000	80000	80000
Produit/Service N°04	Désignation:	Licences annuelles						
	Quantité:	416		18	43	75	115	165
	Prix unitaire:	160000		160000	160000	160000	160000	160000
	Sous-Total:		0	2880000	6880000	12000000	18400000	26400000
Produit/Service N°05	Désignation:	Développement sur mesure						
	Quantité:	36		3	5	7	9	12
	Prix unitaire:	200000		200000	200000	200000	200000	200000
	Sous-Total:		0	600000	1000000	1400000	1800000	2400000
TOTAL:			0	6686000	12310000	19000000	27188000	37520000
Progression:				-	84%	54%	43%	38%

Tableau 4 Chiffre d'affaires

Grille tarifaire - Licences annuelles

Niveau	Nb d'étudiants	Forfait annuel (DZD)	Remise %
Niveau 1	1 - 499	160,000	0.0%
Niveau 2	500 - 999	280,000	12.5%
Niveau 3	1 000 - 1 999	450,000	29.7%
Niveau 4	2 000 - 3 999	700,000	45.3%
Niveau 5	4 000 et +	1,000,000	Variable

Tableau 5 Grille tarifaire Des Licences Annuelles

Charges externes

	à intégrer ↓ Historique (n-1) FY22	à intégrer ↓ Année 01 FY23	à intégrer ↓ Année 02 FY24	à intégrer ↓ Année 03 FY25	à intégrer ↓ Année 04 FY26	à intégrer ↓ Année 05 FY27
DZD						
Sous-traitance						
Electricité		30,000	30,000	36,000	36,000	42,000
Frais Marketing		100,000	150,000	125,000	175,000	200,000
Honoraires du Notaire						
Honoraires d'expert-comptable						
Honoraires Commissaire aux Comptes						
Frais du transit						
Frais télécom		120,000	120,000	144,000	144,000	168,000
Divers fournitures		80,000	80,000	96,000	96,000	112,000
Frais de formation						
R&D		100,000	140,000	180,000	220,000	260,000
Autre 1						
Autre 2						
Total	0	430,000	520,000	581,000	671,000	782,000

Tableau 6 Charges externes

	à intégrer ↓		Année 01 FY23	Année 02 FY24	Année 03 FY25	Année 04 FY26	Année 05 FY27
Clients							
DZD	DSO (délai de paiement clients)						
Chiffre d'affaires			6,686,000	12,310,000	19,000,000	27,188,000	37,520,000
Clients en jours de CA	0		0	0	0	0	0
Clients			-	-	-	-	-
Fournisseurs							
DZD	DSO (délai fournisseur)						
Consommation matières			99,000	137,500	176,000	220,000	275,000
Achats			430,000	520,000	581,000	671,000	782,000
Fournisseurs en jours d'achats	0		0	0	0	0	0
Fournisseurs			-	-	-	-	-
Stocks							
DZD							
Consommation matières			99,000	137,500	176,000	220,000	275,000
Stocks de MP en jours d'achats consommés	60		60	60	60	60	60
Stocks			16,500	22,917	29,333.33	36,666.67	45,833.33

Tableau 7 Besoin en Fonds de Roulement

	N-1 (historique)	Année 01	Année 02	Année 03	Année 04	Année 05
DZD	FY22	FY23	FY24	FY25	FY26	FY27
Chiffre d'affaires	-	6,686,000	12,310,000	19,000,000	27,188,000	37,520,000
Achats consommés	-	(99,000)	(137,500)	(176,000)	(220,000)	(275,000)
Marge brute	-	6,587,000	12,172,500	18,824,000	26,968,000	37,245,000
Charges externes	-	(430,000)	(520,000)	(581,000)	(671,000)	(782,000)
Salaires et charges sociales	-	(5,802,120)	(9,468,720)	(13,360,440)	(13,360,440)	(16,337,400)
EBITDA	-	354,880	2,183,780	4,882,560	12,936,560	20,125,600
Amortissements et provisions		(215,000)	(342,000)	(458,000)	(458,000)	(458,000)
Reprise sur provisions		-	-	-	-	-
EBIT		139,880	1,841,780	4,424,560	12,478,560	19,667,600
Charges financières						
Résultat avant impôts		139,880	1,841,780	4,424,560	12,478,560	19,667,600
Impôts sur les sociétés		-	-	-	-	-
Résultat net		139,880	1,841,780	4,424,560	12,478,560	19,667,600
<i>Evolution du chiffre d'affaires</i>		-	84%	54%	43%	38%
<i>Tx de marge brute</i>	-	99%	99%	99%	99%	99%
<i>Tx d'EBITDA</i>	-	5%	18%	26%	48%	54%

Tableau 9 compte de résultat

	Année 01	Année 02	Année 03	Année 04	Année 05
DZD	FY23	FY24	FY25	FY26	FY27
EBITDA	354,880	2,183,780	4,882,560	12,936,560	20,125,600
Variation de BFR	(16,500)	(6,417)	(6,417)	(7,333)	(9,167)
<i>BFR d'exploitation</i>	16,500	22,917	29,333	36,667	45,833
<i>BFR hors exploitation</i>					
BFR	16,500	22,917	29,333	36,667	45,833
IBS	-	-	-	-	-
Flux de trésorerie provenant de l'exploitation	338,380	2,177,363	4,876,143	12,929,227	20,116,433
CAPEX (Investissements)	(1,075,000)	(635,000)	(580,000)	-	-
Flux de trésorerie provenant de l'investissement	(1,075,000)	(635,000)	(580,000)	-	-
Free cash flow	(736,620)	1,542,363	4,296,143	12,929,227	20,116,433
Emprunt bancaire					
Apport en capital					
Charges financières	-	-	-	-	-
Dividendes					
Flux de trésorerie provenant du financement	-	-	-	-	-
Net Cash flow	(736,620)	1,542,363	4,296,143	12,929,227	20,116,433
Solde initial	-	(736,620)	805,743	5,101,887	18,031,113
Solde final	(736,620)	805,743	5,101,887	18,031,113	38,147,547

Tableau 8 Flux de Trésorerie

Détection des erreurs « ambigu » dans l'analyse des pixels

Après avoir identifié et lu le code-barres, le Ai recherche Les tableau. Pour ce faire, « Canny Edge Detection » est utilisé afin de détecter les contours tableau automatiquement. Ensuite, il analyse la densité des pixels ou les variations de l'intensité des niveaux de gris dans chaque

Exemple [0,23,300,22] X dans 3^{eme} cell

Si la densité de la case la plus remplie n'est pas suffisamment supérieure à celle de la deuxième case, c'est-à-dire lorsque la densité de cette dernière atteint ou dépasse 60 % de la première, le programme considère le résultat comme ambigu et non fiable. Dans ce cas, la note n'est pas validée.

Exemple [0,23,300,200] X dans 3^{eme} celle mais 4^{eme} Très proche

Ce mécanisme permet d'éviter les erreurs dues à un marquage peu clair de la part du correcteur, ainsi qu'aux éventuelles erreurs de détection automatique, telles qu'un mauvais repérage du tableau ou la présence de bruit dans l'image. Cela garantit une meilleure précision lors de l'extraction et de l'enregistrement des notes dans la base de données.