

MINISTRY OF HIGHER EDUCATION AND SCIENTIFIC RESEARCH ABDELHAMID IBN

BADIS UNIVERSITY - MOSTAGANEM

FACULTY OF EXACT SCIENCES AND COMPUTER SCIENCE



Department of Mathematics and Computer Science

Sector: Computer science

END OF STUDY DISSERTATION

For Obtaining a Master's Degree in Computer Science

Option: Artificial Intelligence for Internet of Things (IA4IoT)

by:

BELMOUMENE NOHA

THEME:

Smart Application for personal food catering service

Project supported under decree 1275

In front of the jury composed of:

Mr SEHABA	PROFESSOR	University of Mostaganem	President
Mr ZAHMANI	MCA	University of Mostaganem	Examiner
Mrs BENHAMED	MCB	University of Mostaganem	Supervisor

Academic Year 2024-2025

Abstract

This thesis presents the integration of Artificial Intelligence (AI) into catering services through the NB-Delights application. By combining advanced methods such as Natural Language Processing (NLP) with BERT and collaborative filtering (COSINE SIMILARITY) for sentiment analysis, the application offers personalized recommendations, intuitive search capabilities, and optimized ordering. The results show significant improvements in user experience, recommendation relevance, and operational efficiency. The work also highlights the technical architecture of NB-Delights, its implementation, and the challenges faced, while emphasizing AI's potential to revolutionize food services.

Keywords: Catering, AI, BERT, NLP, COSINE SIMILARITY, Recommendation.

Résumé

Ce mémoire présente l'intégration de l'intelligence artificielle (IA) dans le service du catering à travers l'application NB-Delights. En combinant des méthodes avancées telles que le traitement du langage naturel (NLP) avec (BERT), le filtrage collaboratif (COSINE SIMILARITY) pour l'analyse de sentiments, l'application offre des recommandations personnalisées, des capacités de recherche intuitives et une optimisation des commandes. Les résultats montrent des améliorations significatives de l'expérience utilisateur, de la pertinence des recommandations et de l'efficacité opérationnelle. Le travail met également en lumière l'architecture technique de NB-Delights, sa mise en œuvre ainsi que les défis rencontrés, tout en soulignant le potentiel de l'IA à révolutionner les services alimentaires.

Mots-clés: Catering, IA, BERT, NLP, COSINE SIMILARITY, Recommendation.

ملخص

تقدم هذه الأطروحة دمج الذكاء الاصطناعي في خدمات تقديم الطعام من خلال تطبيق

NLP. من خلال الجمع بين الأساليب المتقدمة مثل معالجة اللغة الطبيعية (NB-Delights)

(لتحليل المشاعر، يقدم التطبيق COSINE SIMILARITY والتصنيف التعاونية (BERT مع توصيات مخصصة وإمكانات بحث بديهية وترتيباً محسناً. وتظهر النتائج تحسنات كبيرة في تجربة المستخدم وملاءمة التوصيات والكفاءة التشغيلية. ويسلط العمل الضوء أيضاً على البنية التقنية لـ وتنفيذها والتحديات التي تواجهها، مع التأكيد على إمكانات الذكاء الاصطناعي NB-Delights لإحداث ثورة في خدمات الأغذية.

كلمات مفتاحية

، معالجة اللغة الطبيعية، تشابه جيب التمام، BERT خدمات الطعام، الذكاء الاصطناعي، أنظمة التوصية.

Dedication

First and foremost, I dedicate this work to **myself** to the person who fought through five long years of exhaustion, doubt, and countless struggles. Especially this past year, when everything seemed overwhelming and the weight unbearable. To the nights filled with tears, the days when giving up felt easier, and the moments of loneliness thank you for standing firm, for summoning strength I didn't even know I had, and for never letting go of my dreams. This journey was hard, but I am prouder than ever of the resilience, courage, and heart that carried me through. I honor you. I thank you. This success belongs to you.

To my beloved **Mami and Dad**, whose love is the light that guided me through the darkest times. Your endless patience, your sacrifices, and your unwavering belief in me gave me hope when I lost it, and strength when I was weak. You are my safe place, my foundation, and my greatest blessing. And **to my dear sisters and brothers**, Thank you for your love and support. It means the world to me.

To my closest **friends**, who laughed with me until my heart was light and cried with me when I needed it most thank you for filling this journey with joy, comfort, and unforgettable memories. You have been my family away from home, and I carry each of you in my heart.

Gratitude

First and foremost, I would like to thank **God Almighty** for granting me the strength, patience, and perseverance to complete this work.

I would like to express my heartfelt gratitude to my supervisor, **Mme Benhamed Siham**, for her invaluable advice, continuous support, and thoughtful guidance throughout this project. Her expertise, encouragement, and dedication were essential to the successful completion of this work. And A special thank you as well to **Mr. Khelifa** for his kind assistance and helpful insights.

I also extend my sincere thanks to everyone whose presence, support, and kind advice gave me the courage and motivation to carry out this project. Your contributions have been truly meaningful.

Finally, I would like to express my deep appreciation to the entire teaching staff of **ABDELHAMID IBN BADIS University of Mostaganem**. Your commitment to education, your support, and your efforts throughout this academic year have played a vital role in my personal and professional growth. Thank you for creating an environment of learning, innovation, and excellence.

List of figures

Figure N°	Figure Title	Page
Figure 1	AI model	19
Figure 2	Apriori flowchart	23
Figure 3	FP-Growth flowchart	24
Figure 4	DBSCAN flowchart	27
Figure 5	Deep Q-Learning flowchart	29
Figure 6	Policy Gradient flowchart	30
Figure 7	Flow diagram of the proposed methodology	37
Figure 8	Transformer model architecture diagram	41
Figure 9	Training architecture diagram of BERT model	45
Figure 10	Application of LLM in e-commerce platform recommendation system	49
Figure 11	general architecture of NB-delights	57
Figure 12	NB-delights Class diagram	59
Figure 13	display Sequence diagram	60
Figure 14	search Sequence diagram	61
Figure 15	subscription Sequence diagram	62
Figure 16	ordering Sequence diagram	63
Figure 17	update Sequence diagram	63
Figure 18	delivery Sequence diagram	64
Figure 19	semantic search diagram	65

Figure 20	smart filter diagram	65
Figure 21	classification and clustering diagram	66
Figure 22	smart selector diagram	66
Figure 23	delivery planification diagram	67
Figure 24	firebase diagram	68
Figure 25	provider use case diagram	69
Figure 26	notify client use case diagram	70
Figure 27	notify driver use case diagram	70
Figure 28	admin dashboard of NB-Delights	76
Figure 29	product managing of NB-Delights	78
Figure 30	add new product	79
Figure 31	update product	79
Figure 32	orders of NB-Delights	80
Figure 33	modify orders of NB-Delights	80
Figure 34	customers of NB-Delights	81
Figure 35	delivery staff of NB-Delights	82
Figure 36	delivery calendar by day of NB-Delights	83
Figure 37	delivery calendar by week of NB-Delights	83
Figure 38	delivery calendar by month of NB-Delights	84
Figure 39	notifications of NB-Delights	84
Figure 40	splash screen of NB-Delights	85
Figure 41	home screen of NB-Delights	85
Figure 42	simple search	86

Figure 43	Smart search	86
Figure 44	filtering screen of NB-Delights	87
Figure 45	pack generate screen of NB-Delights	87
Figure 46	product details screen of NB-Delights	88
Figure 47	about NB-Delights screen	89
Figure 48	profile screen	90
Figure 49	sign up screen of NB-Delights	91
Figure 50	login screen of NB-Delights	91
Figure 51	Confirmation Phone number screen	92
Figure 52	Update password screen	92
Figure 53	Password recovery screen	92
Figure 54	profile screen	93
Figure 55	order history screen	93
Figure 56	edit profile screen	93
Figure 57	favourite screen of NB-Delights	93
Figure 58	cart screen of NB-Delights	95
Figure 59	customizing screen of NB-Delights	95

List of tables

Table N°	Table title	Page
Table 1	The difference between Deep Q-learning & Policy Gradient methods	31
Table 2	The features of FM & FFM	32
Table 3	Use case of methods	34

Liste of abbreviations

Abbreviations	Expression Complete
FFT	Fast Fourier Transform
FSEI	Faculté des Sciences Exactes et de l'Informatique
AI	Artificial Intelligence
NLP	Natural Language Processing
LSTM	Long Short-Term Memory
BERT	Bidirectional Encoder Representations from Transformers
MDP	Markov Decision Process
RL	Reinforcement Learning
GNN	Graph Neural Network
CTR	Click-Through Rate
FM	Factorization Machines
FFM	Field-aware Factorization Machines
ARIMA	Auto-Regressive Integrated Moving Average
MLM	Masked Language Model
NPP	Next Purchase Prediction
K-NN	k-Nearest Neighbors
DBSCAN	Density-Based Spatial Clustering
BST	Behavior Sequence Transformer
WDL	Wide & Deep Learning
DIN	Deep Interest Network

MLP	Multi-Layer Perceptron
-----	------------------------

Table of content

General Introduction	13
Chapter 1 Impact of AI Methods	17
1.1 Introduction.....	17
1.2 Artificial intelligence (AI).....	17
1.2.1 Definition of Artificial Intelligence (AI).....	17
1.2.2 History of Artificial Intelligence	18
1.2.3 Basic Principle of an Intelligent System.....	18
1.3 AI Methods.....	20
1.3.1 Extraction of Information from Human Speech (Analysis)	20
1.3.1.1 Natural Language Processing (NLP).....	20
1.3.1.2 Association Rule Mining	22
1.3.1.3 Sentiment Analysis	24
1.3.2 Collaborative Filtering.....	25
1.3.3 Clustering Algorithms	26
1.3.4 Deep Learning Methods	28
1.3.4.1 Methods Based on Neural Networks	28
1.3.4.2 Reinforcement Learning	28
1.3.5 Recommendation Methods	31
1.3.5.1 Recommendation Based on Matrix (Factorization Machines).....	31
1.3.5.2 Recommendation Based on Graph	33
1.3.5.3 Recommendation Based on Pattern (Time-Series Forecasting)	33
1.6 Use Case.....	34
1.7 Conclusion.....	35
Chapter 2 AI-Based Recommendation System	36
2.1 Introduction.....	36
2.2 Reinforcement Learning Recommender System using Bi-Clustering and Markov Decision Process.....	36

2.3 Transformer Models for E-Commerce.....	40
2.4 ML-Based Product Classification (BERT + k-NN).....	44
2.5 LLM-Enhanced Recommendations.....	47
2.6 GNN for Football Formations.....	51
2.7 Conclusion.....	55
Chapter 3 CONCEPTION OF NB-DELIGHTS	56
3.1 Introduction.....	56
3.2 NB-Delights Model Architecture.....	57
3.3 NB-Delights Class Diagram	58
3.3.1 Client Mobile Application	60
3.3.1.1 Sequence Diagrams.....	60
Display Sequence Diagram.....	60
Search Sequence Diagram.....	61
Subscription Sequence Diagram.....	61
Ordering Sequence Diagram	62
Update Sequence Diagram.....	63
Delivery Sequence Diagram	63
3.3.2 Backend API	64
Semantic Search	64
Smart Filter	65
Classification and Clustering.....	65
Smart Selector	66
Delivery Planification	67
3.3.3 Data base.....	67
3.3.4 Use Case Diagram.....	68
Provider Web Dashboard Use Case Diagram	68
Notification System	69
3.4 Conclusion	71
Chapter 4 Implementation of Personal Food Catering Service (NB-Delights)	72

4.1 Introduction.....	72
4.2 Development Environment.....	72
4.2.1 Development Software	72
4.2.2 Modeling Software.....	73
4.2.3 Programming Languages	73
4.2.4 Frameworks and Tools	73
4.2.5 Libraries and Dependencies.....	74
4.2.6 AI Model Files Used in Backend.....	74
4.2.7 Hardware Requirements	75
4.3 Demonstration of Interfaces	75
4.3.1 Admin Dashboard of NB-Delights.....	75
4.3.2 Client Mobile Application NB-Delights.....	85
4.4 Conclusion	95
Conclusion Générale	96
Bibliographie.....	97
Annex BMC , Business Plan	101

General Introduction

The integration of Artificial Intelligence (AI) into our daily lives has revolutionized many sectors by offering intelligent and personalized solutions. Thanks to advanced methods such as Natural Language Processing (NLP), Machine Learning, and Neural Networks, AI enhances efficiency, accuracy, and user experience. From personalized recommendations to virtual assistants, AI simplifies complex tasks and adapts to individual needs, becoming an essential tool in modern life.

In the field of catering and food services, AI brings major transformation by enabling customized services. It analyzes food preferences, dietary restrictions, and seasonal trends to suggest menus tailored to each event. Whether it's a wedding, a birthday, or a corporate meeting, AI optimizes meal planning, reduces waste, and improves customer satisfaction through intelligent and dynamic recommendations.

The NB-Delights application aims to modernize the catering sector by offering an intelligent and personalized platform. It seeks to simplify dish ordering, anticipate customer needs, and optimize logistics using AI. By combining advanced methods such as NLP, collaborative filtering, and sentiment analysis, the application provides a unique experience that blends convenience with innovation.

To achieve this goal, several AI techniques have been integrated into NB-Delights:

- **BERT and NLP** for understanding user queries and enabling semantic search.
- **Collaborative filtering and cosine similarity** for personalized recommendations.
- **Sentiment analysis** to evaluate customer feedback and improve suggestions.
- **Neural networks (LSTM)** to predict order trends.
- **Association rules (FP-Growth)** to suggest popular dish combinations.

Each of these methods plays a key role:

- BERT and NLP enable intuitive and accurate search by understanding users' natural language.
- Collaborative filtering enhances the relevance of recommendations based on similar customers' preferences.

- Sentiment analysis ensures continuous adaptation to feedback, boosting customer satisfaction.
- LSTM networks anticipate seasonal demand, optimizing stock management.
- FP-Growth suggests complete menus, increasing average basket size and simplifying user choices.

This project is structured into four chapters:

1. **Impact of AI Methods:** Presentation of key concepts and techniques used.
2. **AI-Based Recommendation System:** Analysis of modern approaches like Transformers and GNNs.
3. **Conception of NB-Delights:** Description of the architecture and UML diagrams.
4. **Implementation:** Technical details, interfaces, and AI integration.

CHAPTER 01

IMPACT of AI METHODS

1.1 Introduction

Recommendation systems are a prominent application of Artificial Intelligence (AI) that have become an essential part of our daily lives. They are designed to analyze user preferences, behavior, and historical data to provide personalized suggestions [1]. These systems are widely used in various domains, including e-commerce, entertainment, education, healthcare, and event planning, to enhance user experience and decision-making [2].

The key goal of a recommendation system is to filter large volumes of data and present the most relevant information to users, thereby reducing the cognitive effort required to make choices. There are several types of recommendation systems, including collaborative filtering, content-based filtering, and hybrid approaches. [3] Each of these methods leverages AI techniques like machine learning, natural language processing, and data mining to deliver tailored suggestions [3].

By incorporating AI, these systems enhance the efficiency and accuracy of recommendations, reduce manual efforts, and ensure a seamless user experience [4]. As highlighted in a report by Gartner (2022) [5], AI-driven recommendation systems are projected to be a key differentiator for businesses seeking to offer personalized and innovative solutions.

1.2 Artificial intelligence (AI)

1.2.1 Definition of Artificial intelligence (AI)

AI is the concept deployed to refer to the modelling of human intelligence processes by machines. It refers to the capability of computers to undertake activities that are rational and logical in nature such as understanding, perception, identification, and decision making [6].

1.2.2 History of Artificial Intelligence

AI can be traced back to the initial formulation in the field of Artificial Intelligence which was established in a workshop at Dartmouth College in 1956. The pioneers of AI include Alan Turing, Marvin Minsky, and John McCarthy for their contributions such as Turing test, neural networks, among other key breakthroughs in the initial development of self-learning programs [7], [8]. However, the expectations that were set regarding AI advancements were very high and then the field of AI went through a period of absolute regression also known as an “AI winter” in the seventies and eighties [9].

However, this has come back again in the 1990s with the new developments in areas such as the neural networks, machine translation and identification of images. The second one is IBM’s Deep Blue which crushed world chess champion Garry Kasparov in the year 1997 [10]. Further advancements in AI occurred in the 2000s, major successful endeavours being speech as a form of interaction and processing, facial recognition, self-driving cars, and in question-answering tools like IBM Watson [9], [8].

AI experienced unprecedented growth in the 2010s due to the rise of computing capacity, the emergence of big data, and marvellous algorithms like deep learning. The current decade has been labelled as the AI or Artificial Intelligence era. Many problems have been tamed with machine learning techniques and AI systems are evolving to be more intelligent and encroaching more into our daily experiences.

1.2.3 Basic principle of an intelligent system

Basic principle of an intelligent system, such as an AI model is an iterative process that commonly includes the following sections (see figure1) [11]:

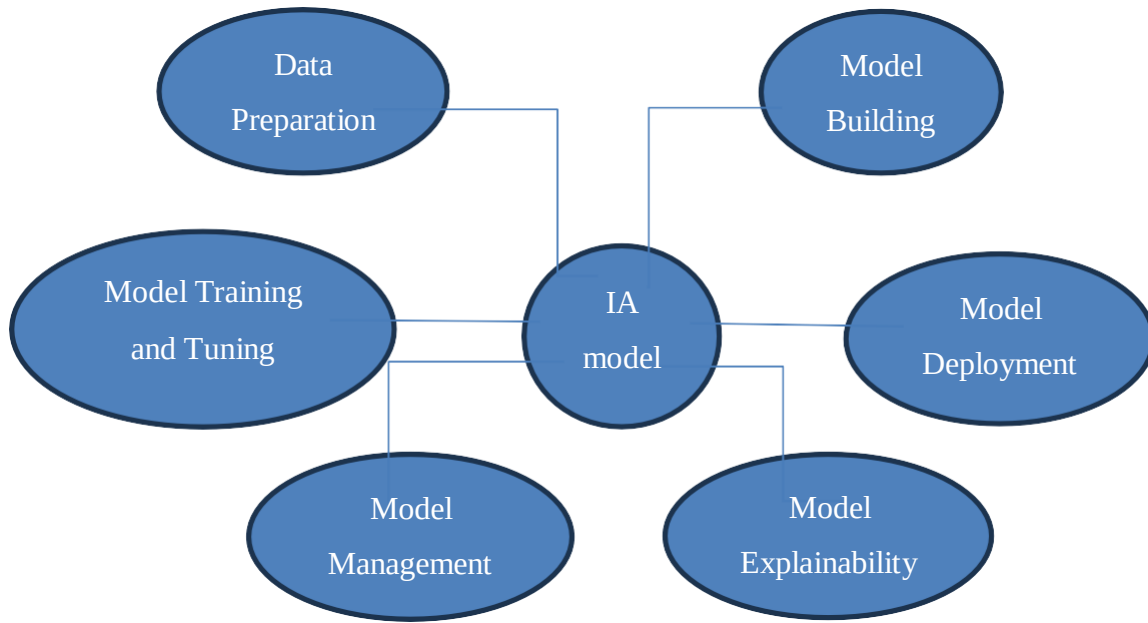


Figure 1: AI model [11]

1.2.3.1 Data Preparation

This initial step involves getting data ready for model training it includes:

- **Data Cleaning:** Removing noise, handling missing values, and ensuring data quality.
- **Feature Engineering:** Creating relevant features or transforming existing ones.
- **Data Transformations:** Normalising, scaling, or encoding features appropriately.
- **Splitting Data:** Dividing your dataset into training, validation, and test sets.

1.2.3.2 Building Models

When we talk about constructing machine learning (ML) or deep learning (DL) models, we refer to the process of building predictive or analytical systems that can learn from data. This process typically involves choosing appropriate frameworks [12], defining the model architecture [13], training with data [14], and evaluating performance [15].

1.2.3.3 Model Training and Tuning

Hyperparameters optimization and model calibration. The steps of the model are based on:

- Training the model on the training data.
- Optimising hyperparameters (like learning rate, batch size, etc.).
- Fine-tuning the model to achieve better performance.

1.2.3.4 Model Deployment

Once the model has been successfully trained, it is deployed into a production environment. This phase ensures that the model is fully operational, accessible to users or systems, and capable of delivering real-time predictions.

1.2.3.5 Model Management

Continuously monitor the deployed model's performance Detection and address any issues that arise.

1.3 AI METHODS

1.3.1 Extraction of information from human speech(analysis)

1.3.1.1 Natural Language Processing (NLP)

NLP focuses on the understanding and generation of human language by machines, including machine translation, speech recognition and the creation of chatbots, Among the most powerful and widely used are BERT (Bidirectional Encoder Representations from Transformers) and Word2Vec [16]. These models transform text into numerical representations that capture meaning and context. BERT is a deep learning model based on transformers that understands language context bidirectionally, while Word2Vec is a shallow neural network that learns semantic relationships between words.

➤ **BERT (Bidirectional Encoder Representations from Transformers) [17]:**

BERT is a transformer-based model designed to extract context-rich features from text. It learns language representations in a bidirectional manner, capturing both left and right context simultaneously. The model training is divided into two phases:

1. Pre-training

1. Data Input: Collection of a large corpus of unannotated text.
2. Tokenization: Splitting the text into subwords (tokens) using a specific vocabulary.
3. Creating the Representation: Adding special tokens such as [CLS] (classification token) and [SEP] (separator token) to indicate the start and separation of sentences.
4. Word Masking and Pre-training:
This step includes two core tasks:
 - Masked Language Model (MLM): Randomly masks a percentage of tokens and trains the model to predict them using surrounding context, enabling bidirectional understanding.
 - Next Sentence Prediction (NSP): Trains the model to predict whether a given sentence logically follows another.
5. Optimization: Using an optimizer (e.g., Adam) to adjust the model's weights based on the training data.

2. Fine-tuning (Adaptation)

- Loading the pre-trained model
- Preparing the data
 - Building an annotated dataset for the specific task (e.g., classification, question answering).
 - Adjusting hyperparameters: setting learning rate, number of epochs, batch size.
- Training
 - Training the model on the specific dataset (e.g., text classification) with an appropriate loss function.
- Evaluation
 - Testing the model on a validation set to assess performance.
- Prediction
 - Using the fine-tuned model to make predictions on new data.

➤ **Word2Vec:**

This method converts words or phrases into dense vector representations by analyzing their usage patterns in large corpora. It captures the meaning of words based on their context, enabling semantic comparisons [18].

1.3.1.2 Association Rule Mining

fundamental technique in data mining that focuses on discovering interesting relationships, patterns, or associations between items in large datasets. It is widely used in market basket analysis, recommendation systems, and customer behavior modeling. For instance, it can reveal that customers who buy bread and butter are likely to also buy jam. The goal is to identify item combinations that occur frequently together and generate actionable rules based on these patterns. These rules are evaluated using key metrics like support, confidence, and lift to ensure their statistical significance and usefulness in decision-making.

To efficiently extract these associations, several algorithms have been developed. The two most well-known are:

➤ **Apriori:**

The process of Apriori algorithm is a process of finding frequent item sets from candidate sets. [19].

Steps of Apriori method :

1. **Initialize:**

- Set a minimum support threshold.
- Extract frequent 1-itemsets (items that meet support threshold).

2. **Generate Candidate Itemsets:**

- Use the **Apriori property** to generate (k+1)-itemsets from k-itemsets.

3. **Filter Itemsets:**

- Keep only those itemsets that meet the minimum support.

4. **Repeat:**

- Continue generating and filtering itemsets until no new frequent itemsets are found.

5. Generate Association Rules:

- For each frequent itemset, generate rules of the form $X \rightarrow Y$ that meet the minimum confidence threshold.

We propose in the following organigram the process of the generation rules:

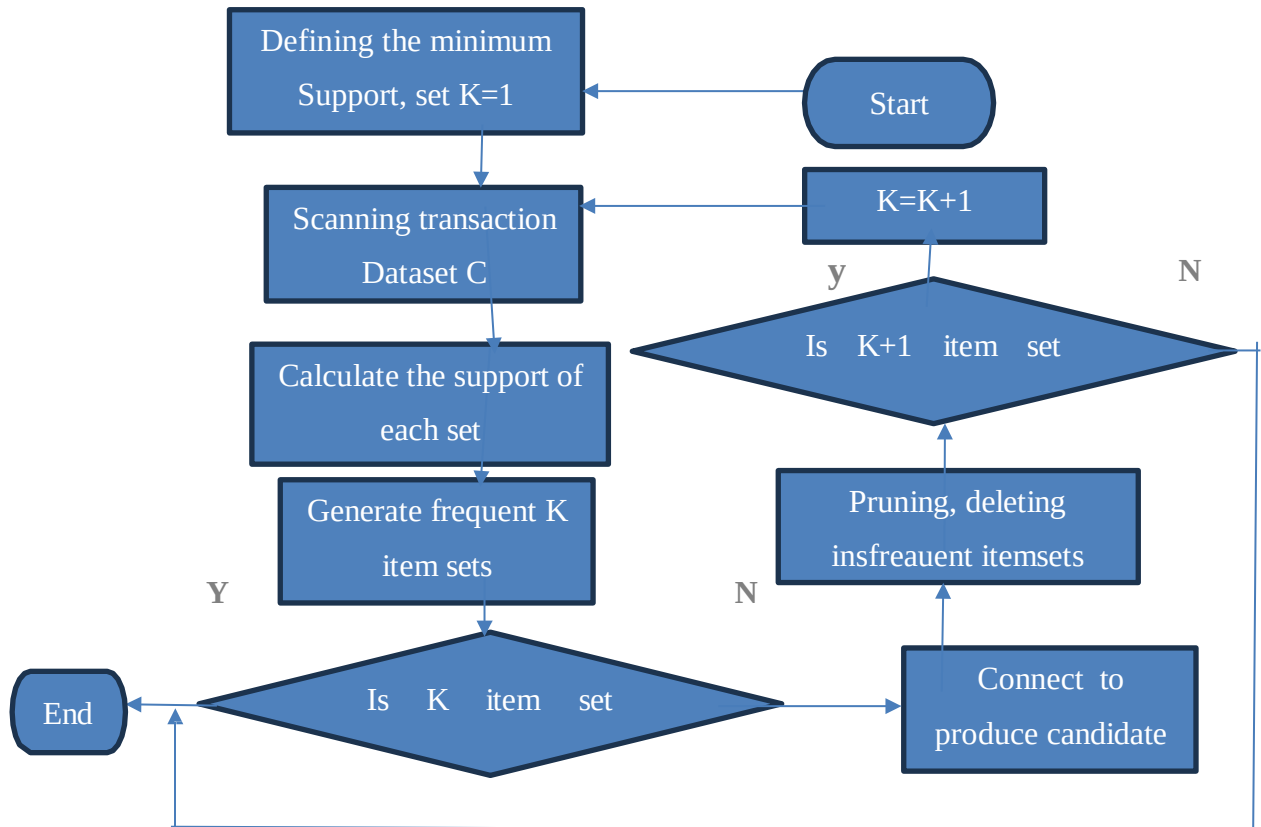


Figure 2: Apriori flowchart

➤ FP-Growth:

FP-Growth is a more scalable and faster method that constructs a compressed structure called the FP-Tree, which eliminates the need for candidate generation by directly mining frequent patterns from the tree and the steps of this method is:

1. Scan the Dataset: Count the frequency of each item.
2. Sort Items: Sort items in descending order of frequency.
3. Construct FP-Tree: Insert transactions into the tree and Shared branches represent common itemsets.
4. Mine Frequent Itemsets: Use recursive pattern growth to extract frequent itemsets from the tree.

5. Generate Association Rules: Derive rules from the mined itemsets.

The process as proposed in [20] is presented in the following organigram:

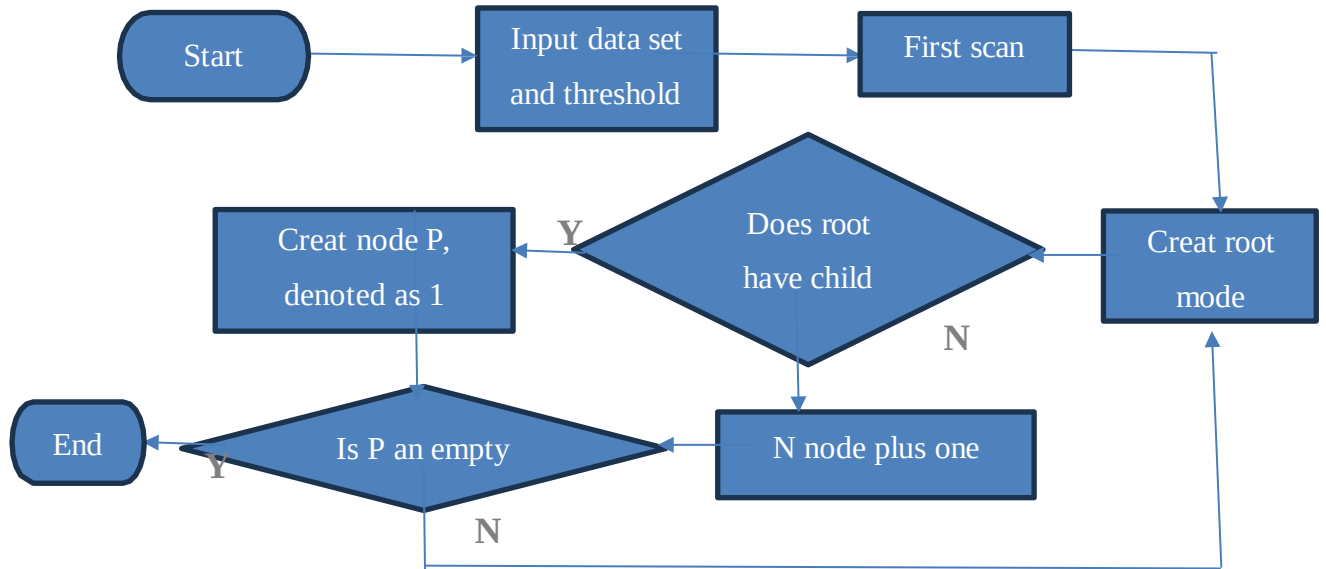


Figure 3: FP-Growth flowchart

1.3.1.3 Sentiment Analysis

This method is a Natural Language Processing (NLP) technique used to automatically extract emotions, opinions, and attitudes from textual data. It is particularly useful in analyzing customer feedback, reviews, and social media content to determine whether the expressed sentiment is positive, negative, or neutral [21]. This analysis allows businesses to better understand customer satisfaction and improve decision-making processes, using two algorithm is:

- **Naive Bayes:** A simple yet effective probabilistic classifier that categorizes text based on word frequency and likelihood, making it well-suited for sentiment classification [22].
- **LSTM (Long Short-Term Memory):** A powerful deep learning model capable of capturing long-range dependencies and context in sequences, such as customer reviews, enabling more accurate sentiment detection, especially in complex or lengthy texts [23].

1.3.2 Collaborative Filtering

Collaborative Filtering is one of the most widely used techniques in recommendation systems. It works by analyzing the behavior and preferences of users to recommend items they might like, based on similarities between users or items. Instead of relying on the content of the items themselves, collaborative filtering focuses on user interactions such as ratings, purchases, or clicks to uncover hidden patterns and suggest relevant options [24].

Several algorithms are commonly used to implement collaborative filtering. Matrix Factorization methods like:

- **Singular Value Decomposition (SVD) or Alternating Least Squares (ALS):** are effective in reducing the dimensionality of large user-item interaction matrices and uncovering latent features that influence preferences [25].
- **k-Nearest Neighbors (k-NN):** is another intuitive approach that recommends items by finding similar users or items based on distance metrics in feature space [26].
- **Cosine Similarity:** is often used to measure how similar two users or items are by comparing the angle between their feature vectors, allowing for accurate recommendations based on user behavior patterns [27].

1.3.3 Clustering Algorithms

are used in recommendation systems to group users or items based on similar behavior or features. These groups can then be used to make group-based recommendations, especially when individual data is limited or noisy [28], Common Algorithms is:

- **K-Means:** Divides users or items into k clusters based on similarity, helping recommend items popular within each group [29].
- **DBSCAN (Density-Based Spatial Clustering):** is a powerful clustering algorithm used to identify dense regions of data without requiring the number of clusters in advance. It is especially effective for discovering clusters of arbitrary shape and handling noise (outliers). DBSCAN groups points that are closely packed together while marking points in low-density areas as outliers [30].

This method goes through several steps:

1. Initialize Parameters: Set ϵ (epsilon – neighborhood radius) and MinPts (minimum points to form a dense region).
2. Label All Points: Start with all points marked as unvisited.
3. For Each Unvisited Point: Mark it as visited and Find all neighbors within ϵ .
4. Check Density:
 - If neighbors $\geq$ MinPts:
 - I. Start a new cluster.
 - II. Add the point and expand the cluster by checking neighbors of neighbors.
 - Continue adding connected dense points to the same cluster.
5. Repeat: Process all points until all are visited.
6. Output: Return all formed clusters and mark remaining points as noise.

We propose the process of the method in the following organigram:

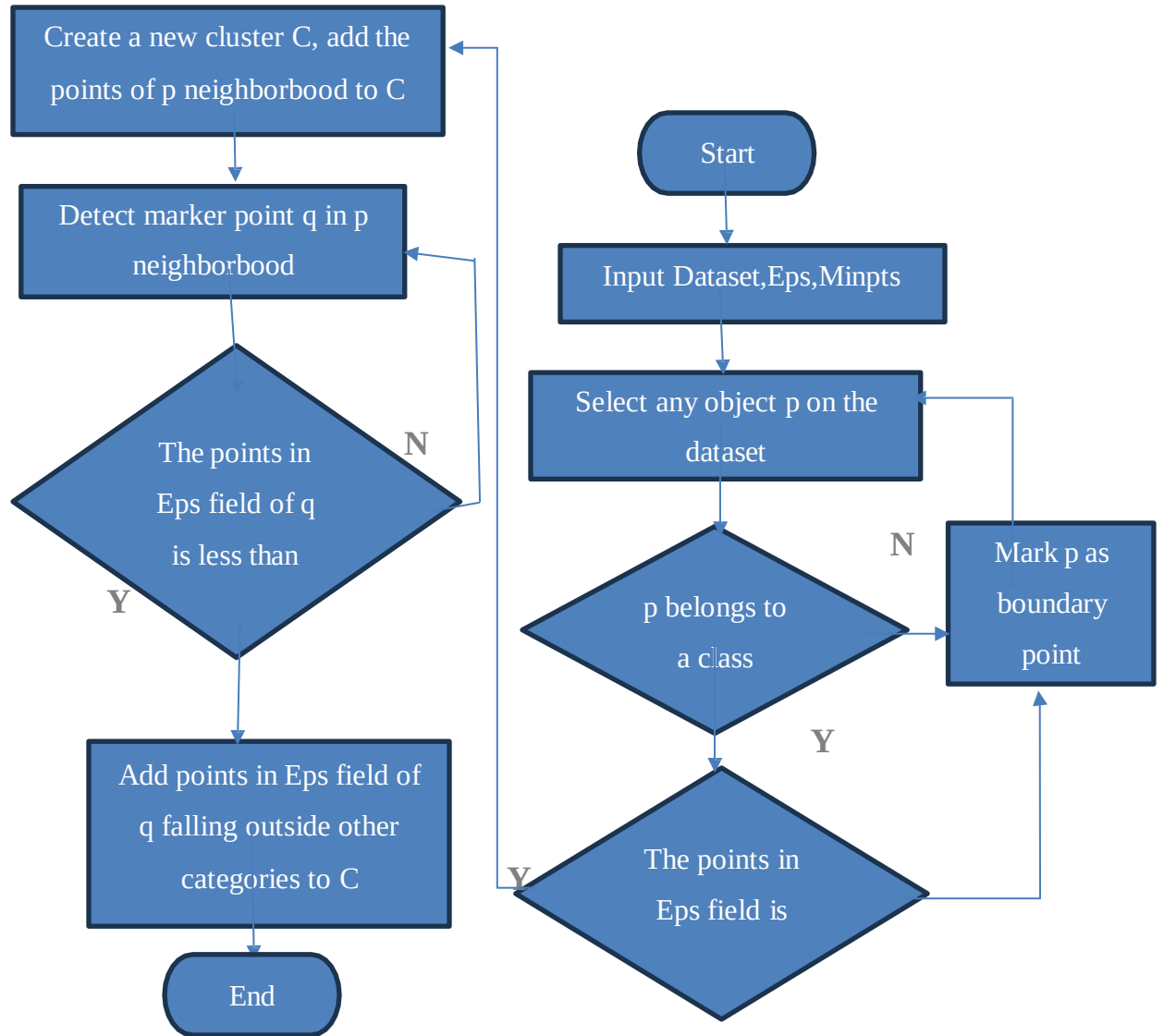


Figure 4: DBSCAN flowchart

1.3.4 Deep Learning Methods

1.3.4.1 Methods Based on Neural Networks

Neural network-based methods in deep learning are highly effective for capturing complex, non-linear relationships in user-item interactions. These models can learn deep features from various data types such as images, text, or sequences, making them especially useful in recommendation systems with rich content [31]. Common Algorithms is:

- **Convolutional Neural Networks (CNNs):** Effective for visual-based recommendations, such as analyzing cake or dessert images to suggest custom designs [32].
- **Recurrent Neural Networks (RNNs):** Best suited for modeling sequential data like user browsing or ordering histories over time [33].

1.3.4.2 Reinforcement Learning

Reinforcement Learning (RL) is a dynamic approach in recommendation systems where an agent learns to make optimal decisions by interacting with users and receiving feedback in the form of rewards. Instead of learning from static data, RL adapts over time, aiming to maximize cumulative rewards by recommending items that lead to positive user responses (like clicks or purchases) [34], Key Algorithms is:

- **Deep Q-Learning [35]:** Learns the best recommendation strategies using trial-and-error.
 - Inputs: User states (e.g., preferences, history), actions (recommendations), reward signals, and a Q-network to estimate action values.
 - Steps:
 1. Initialize Q-network and exploration rate (ϵ).
 2. For each interaction:
 - Observe state, choose an action (random or based on Q), receive reward, observe new state.
 - Update Q-values using the Bellman equation.
 3. Repeat until the model converges to an optimal recommendation policy.

We propose the process of the method in the following organigram:

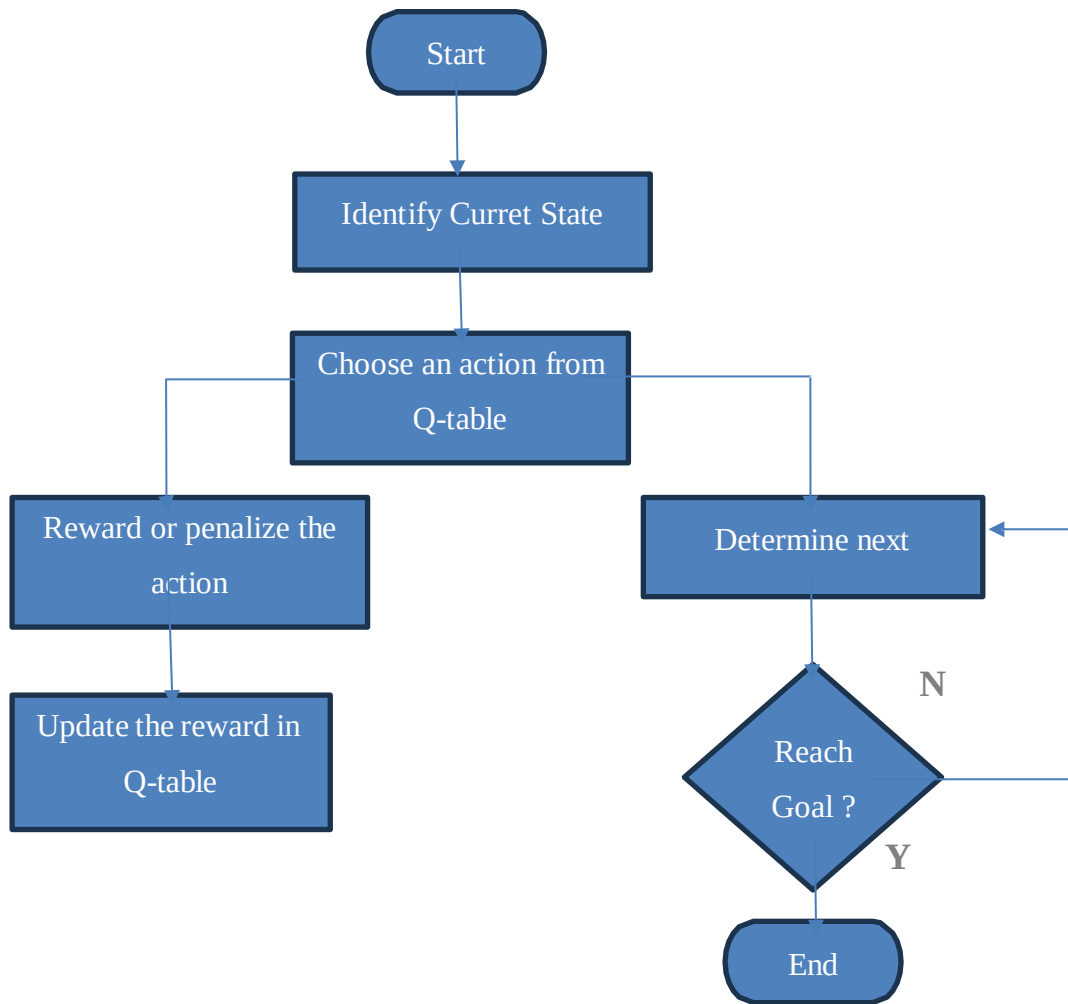


Figure 5: Deep Q-Learning flowchart

- **Policy Gradient Methods [36]:** Directly optimize the recommendation policy based on user feedback.
 - Inputs:
 - User states, action space, a policy function $\pi(s, a, \theta)$, and a reward function.
 - Steps:
 1. Initialize policy parameters θ .
 2. For each interaction:
 - Sample an action from the policy, execute it, and receive a reward.
 - Compute cumulative rewards and update the policy using gradients.
 3. Repeat until convergence for optimal recommendations.

We propose the process of the method in the following organigram

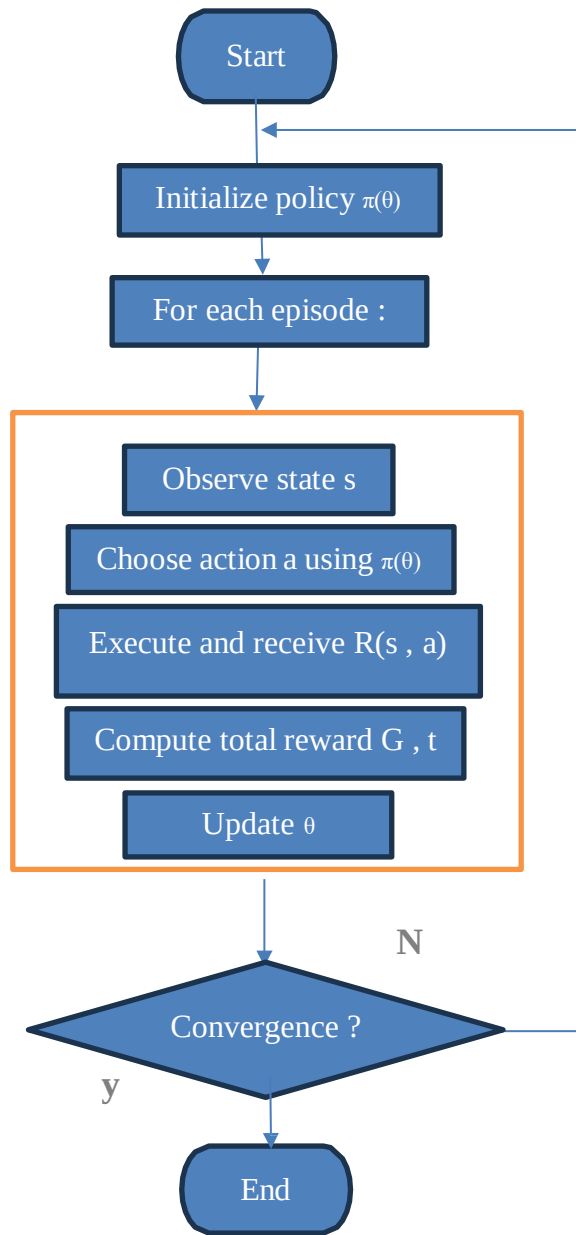


Figure 6: Policy Gradient flowchart

1.3.4.2.1 Advantages and Disadvantages

The following table provide a concise comparison between two pairs of widely used machine learning techniques. The table contrasts Deep Q-Learning and Policy Gradient—two popular reinforcement learning methods—highlighting their learning approaches, strengths, and limitations. Deep Q-Learning is more suited for discrete environments due to its value-based learning, while Policy Gradient methods are better for continuous action spaces thanks to their

direct policy optimization, This (Table 1) is result of our understanding Deep Q-learning & Policy Gradient methods.

Table 1: the difference between Deep Q-learning & Policy Gradient methods

Method	Approach	Advantages	Disadvantages
Deep Q-learning	Learns the value of actions $Q(s,a)$	More stable and efficient in discrete environments	Doesn't work well with continuous actions
Policy Gradient	Directly learns the policy $\pi(s,a)$	Works better for continuous actions, better exploration	More unstable and requires more computation

1.3.5 Recommendation Methods

1.3.5.1 Recommendation based on matrix (Factorization Machines)

Factorization Machines (FM) are a powerful recommendation technique that extends traditional matrix factorization by modeling interactions not just between users and items, but also between additional features (like context, time, or user demographics). This makes FM highly effective for sparse data and allows it to capture complex patterns that improve recommendation accuracy [37], Key Algorithm is:

➤ Factorization Machines (FM) [38]:

- Learns latent factors for features and their pairwise interactions.
- Can handle high-dimensional and sparse datasets efficiently.
- Suitable for both regression and classification tasks in recommendation.

➤ Field-Aware Factorization Machines (FFM):

- A variant of FM that considers fields (groups of features) for better interaction modeling.

- Often used in click-through rate (CTR) prediction and personalized recommendations [39].

1.3.5.1.1 Features of FM and FFM

The following table compares Factorization Machines (FM) and Field-aware Factorization Machines (FFM), both used in recommendation systems. FM handles general feature interactions efficiently, whereas FFM provides better accuracy by considering the context (field-awareness), at the cost of increased computational complexity. This (Table 2) is result of our understanding features of FM & FFM.

Table 2: the features of FM & FFM

Feature	FM	FFM
Feature interactions	yes	Yes (Field-aware)
Accuracy	Moderate	Higher
Computation Time	Faster	Slower (more complex)
Use Case	Generic recommendations	Context-aware recommendations

1.3.5.2 Recommendation based on graph

Graph-based recommendation methods model the interactions between users and items as a graph structure, where nodes represent entities (e.g., users, products) and edges denote relationships (e.g., purchases, likes). This approach allows the system to leverage rich connection patterns and contextual information to enhance recommendation accuracy [40].

- **Graph Neural Networks (GNNs):** are particularly effective in this setting, as they aggregate information from a node's neighbors, capturing both direct and indirect user-item relationships. GNNs are highly suitable for large-scale, sparse, and complex datasets common in recommendation systems [41].

1.3.5.3 Recommendation based on pattern (Time-Series Forecasting)

Pattern-based recommendation methods leverage historical data to predict future trends in user behavior or item demand. These approaches are especially useful for time-dependent data, such as seasonal preferences or recurring purchases. By analyzing past order patterns, the system can forecast future interactions, improving planning and personalization. Time-series forecasting algorithms like ARIMA and LSTM are commonly used for this purpose [42], Key Algorithms:

- **ARIMA (Auto-Regressive Integrated Moving Average):**
 - Models temporal structures in data.
 - Effective for short-term forecasting of future orders or product demand [43].
- **LSTM (Long Short-Term Memory):**
 - A type of recurrent neural network (RNN) that captures long-term dependencies.
 - Ideal for sequential modeling of user behavior over time [23].

1.6 USE CASE

In an increasingly competitive landscape, understanding customer preferences and behaviors is crucial for effective menu design and recommendation strategies in the food and beverage industry. This chapter explores a variety of use cases that leverage advanced analytics and machine learning techniques to enhance customer experiences and drive sales. From analyzing customer reviews through sentiment analysis [44] to dynamically adjusting recommendations based on real-time interactions using reinforcement learning [45], these methodologies provide valuable insights for tailoring offerings to meet evolving consumer demands. By integrating techniques such as collaborative filtering [46] and time-series forecasting [47], businesses can build a more responsive and personalized environment that delights customers and fosters loyalty. Through a detailed examination of these strategies, this chapter aims to provide actionable insights for practitioners seeking to harness the power of data in crafting innovative culinary experiences.

The following table presents a set of common AI and machine learning use cases, along with the most appropriate methods and real-life examples for each. This overview is useful for quickly identifying the right approach depending on the problem we want to solve, this (Table 3) is result of our understanding use case of methods

Table 3: use case of methods

Use case	Methods	Examples
Analyzing customer reviews for better recommendations	Word2Vec, Sentiment Analysis	Analyzing sentiments in reviews like "I love this cake" vs. "This cake is too dry."
Finding similar dishes using descriptions	Word2Vec	"Chocolate lava cake" is similar to "Rich cocoa muffin" due to shared keywords.
Suggesting dish combinations often ordered together	Association Rule Mining (Apriori, FP-Growth)	Customers who buy "bread" and "butter" tend to also buy "jam."
Improving recommendations by analyzing customer feedback	Sentiment Analysis (Naive Bayes, LSTM)	Detecting negative sentiments in reviews about an item leads to removing it from the menu.
Recommending dishes based on other customers' preferences	Collaborative Filtering (SVD, k-NN, Cosine Similarity)	If a customer enjoys "Chocolate cake," recommend "Dark chocolate muffin" based on other users' similar tastes.
Segmenting customers into clusters for tailored recommendations	Clustering Algorithms (K-Means, DBSCAN)	Grouping customers as "budget-conscious" or "luxury-focused" based on their purchase history.
Recommending dishes based on user preferences and trends	Deep Learning Methods (CNNs, RNNs)	Recommending seasonal items like pumpkin spice desserts in the fall based on past ordering trends.
Dynamically adjusting recommendations based on real-time customer interactions	Reinforcement Learning (Deep Q-Learning, Policy Gradient Methods)	Adjusting recommendations in real-time based on user actions, like showing "pizza" if they click on "Italian food."
Making predictions by incorporating additional features	Factorization Machines (FM, FFM)	Using features like event type (e.g., "birthday") and location to recommend specific dishes.
Recommending menus by exploring relationships between	Graph Neural Networks (GNNs)	Suggesting meals based on users who liked similar

user preferences and item popularity		items or based on popular dishes in social circles.
Forecasting demand for specific items based on seasonal trends	Time-Series Forecasting (ARIMA, LSTM)	Predicting higher demand for "chocolate cakes" around Valentine's Day based on historical data.

1.7 CONCLUSION

The integration of advanced analytics and artificial intelligence (AI) methods into the food and beverage industry has significant implications for enhancing customer experiences, optimizing menu design, and increasing revenue. By employing techniques such as sentiment analysis, collaborative filtering, and reinforcement learning, businesses can gain deeper insights into consumer preferences and behaviors, allowing for personalized recommendations that resonate with individual tastes [48]. This personalization not only improves customer satisfaction but also fosters loyalty, leading to long-term patronage [49].

Furthermore, the use of real-time data and predictive analytics enables businesses to anticipate trends and adjust their offerings dynamically, ensuring that their menus remain relevant and appealing in a fast-changing market [47]. The ability to analyze vast amounts of data from customer interactions, social media, and reviews empowers restaurants and brands to craft culinary experiences that align with customer desires, ultimately maximizing both consumer engagement and profitability.

As the industry continues to evolve, the successful application of AI and data-driven strategies will play a pivotal role in shaping the future of food and beverage offerings. Companies that embrace these methods stand to gain a competitive edge, paving the way for innovative applications and enhanced customer relationships.

CHAPTER 02

SYSTEME OF RECOMMENDATION BASE ON AI METHOD

2.1 Introduction

In this chapter, we explore various intelligent recommendation systems that leverage artificial intelligence techniques to deliver personalized and context-aware suggestions to users. The aim is to examine how modern machine learning models—ranging from reinforcement learning to transformers and graph neural networks—are transforming the recommendation landscape. Recommendation systems play a critical role in today's digital ecosystem. They help users discover relevant content, reduce decision fatigue, and improve engagement across platforms such as e-commerce, media streaming, and social networks. AI-based recommendation techniques are especially powerful because they adapt over time, learn from user interactions, and capture complex behavioral patterns.

2.2 Reinforcement Learning Recommender System using Bi-Clustering and Markov Decision Process

This method proposes a hybrid AI approach that combines bi-clustering with reinforcement learning through Markov Decision Processes (MDPs). The goal is to transform the recommendation task into a sequential decision-making problem and enhance adaptability to changing user behavior. [50]

2.2.1 Method Objective

This method aims to overcome the limitations of traditional recommendation systems by addressing challenges such as changing user preferences, data sparsity, the cold start problem, and the need to consider sequential user interactions. It models the recommendation task as a sequential decision-making process using Markov Decision Processes (MDP) and leverages reinforcement learning to optimize the recommendation policy effectively.

2.2.3 Input Data Type

The input data includes a user-item rating matrix, where each cell represents the interaction (e.g., rating) between a user and an item. This matrix is transformed into a binary matrix (1 for interaction, 0 for no interaction).

Two datasets are used in this study:

- MovieLens ML-100K (943 users, 1,682 movies, 100,000 ratings).
- FilmTrust (1,508 users, 2,071 movies, 35,497 ratings).

2.2.4 Proposed Solution of the Method

The method simplifies user-item interactions into a binary matrix, groups similar users and items using bi-clustering, and optimizes recommendations through reinforcement learning based on Markov Decision Processes.

This figure describes a pipeline for creating and optimizing bi-clusters, evaluating their quality, and integrating them into a recommender system powered by reinforcement learning. The squared grid acts as a foundational structure for both cluster organization and the MDP states, while user interactions drive the system's recommendations.

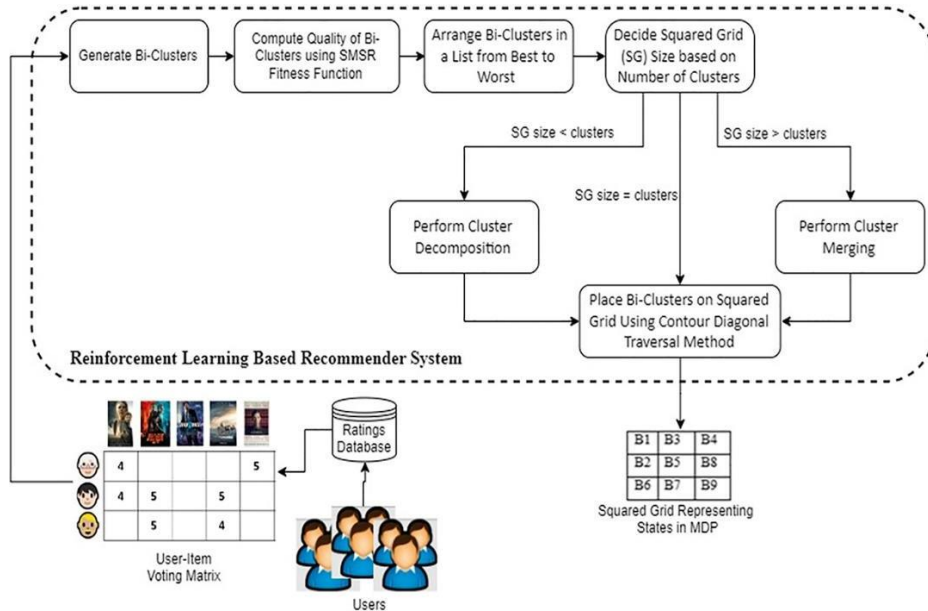


Figure 7: Flow diagram of the proposed methodology [50]

2.2.5 Methods, Approaches, and Algorithms Used

2.2.5.1 Methods Used

- **Bi-clustering:** Identifies dense blocks in the binary matrix, reducing dimensionality and revealing latent patterns.
- **Markov Decision Process (MDP):** Models the recommendation process as a set of state transitions based on user-item interactions.
- **Reinforcement Learning (RL):** Learns an optimal policy to maximize long-term user satisfaction.
- **ITR Similarity Measure:** Used to determine the best initial state for recommendation.
- **Cantor-diagonal traversal:** Orders bi-clusters spatially in a square grid.

2.2.5.2 Algorithms Used

2.2.5.2.1 Bi-clustering Algorithms

To enhance recommendations, integrated techniques analyze interactions, optimize decisions, and boost accuracy and personalization.

- **BiMax and Bibit:** Algorithms for identifying binary subgroups within the matrix.
- **Q-Learning:** Off-policy RL algorithm for discovering optimal actions.
- **SARSA:** On-policy RL algorithm for refining decisions based on current policies.
- **Jaccard Similarity:** Defines the reward function based on overlapping users/items.
- **ϵ -greedy Strategy:** Balances exploration and exploitation during learning.

2.2.5.2.2 Reinforcement Learning Algorithms

The system uses Q-Learning for off-policy learning to find optimal actions and SARSA for on-policy learning to adjust strategies in real time. A reward function based on Jaccard similarity guides learning, while an ϵ -greedy policy ensures a balance between exploration and exploitation.

2.2.6 Method Steps

2.2.6.1 Data Preprocessing

The process starts by converting the user-item matrix into a binary format, then applies bi-clustering (BiMax and Bibit) to group similar users and items, enabling more focused recommendations.

2.2.6.2 Building the MDP Model

After identifying user-item groups through bi-clustering, the next step organizes these groups to simplify decision-making. This involves mapping the bi-clusters onto a square grid, merging them to reduce complexity, and clearly defining the states, actions, and rewards for the recommendation model.

2.2.6.3 Determining the Initial State

To effectively initialize the recommendation process, the system determines an appropriate starting point. This is achieved by using the ITR similarity measure to identify the most relevant initial state based on user and item characteristics.

2.2.6.4 Applying Reinforcement Learning

The reward function is defined based on the overlap of users and items between neighboring states. To learn the optimal recommendation policy, both Q-Learning and SARSA algorithms are employed. Additionally, an ϵ -greedy policy is implemented to balance exploration of new actions with exploitation of known successful strategies.

2.2.6.5 Generating Recommendations

Once the optimal policy has been learned through reinforcement learning, it can be applied to make personalized recommendations. This policy guides the system in suggesting the most relevant items to users based on their preferences and past interactions.

2.2.7 Method Advantages

This approach offers several key advantages. It enables dynamic adaptation to evolving user preferences through reinforcement learning, leading to improved precision and recall compared

to traditional methods. By using bi-clustering, the model effectively reduces the state space, addressing data sparsity and enhancing efficiency. The ϵ -greedy policy ensures a balanced exploration of new recommendations while exploiting known successful ones. Additionally, the model provides explainability by clarifying why a particular recommendation is made to a user.

2.2.8 Method Limitation

Despite its benefits, this method faces some challenges. It involves high computational costs due to bi-clustering and Markov Decision Process (MDP) implementation, and training Q-Learning and SARSA algorithms can be time-consuming. The model's performance is sensitive to parameter choices, such as bi-clustering thresholds and the optimization of ϵ -greedy strategy, learning rate (α), and discount factor (γ). Additionally, selecting the appropriate bi-cluster size is critical, as clusters that are too large may lose recommendation detail, while overly small clusters can negatively impact recommendation quality.

2.3 Text Understanding and Generation Using Transformer Models for Intelligent E-commerce Recommendations [51]

This method focuses on enhancing e-commerce recommendation systems by leveraging the capabilities of **Transformer-based models**, which are highly effective in understanding and generating natural language. These models capture deep contextual relationships in product information and user behavior, thereby improving recommendation precision and personalization.

2.3.1 Method Objective

The main objective of this method is to utilize Transformer models to overcome the limitations of traditional recommendation systems, such as:

- The loss of semantic information when using ID-based representations.
- The cold-start problem and data sparsity, particularly for new users or products.
- The inability to capture long-term and complex dependencies in user-product interactions.

By understanding textual data and modeling sequential behaviors, this method aims to improve both the relevance and diversity of recommendations.

2.3.3 Input Data Type

The system uses a rich and varied set of input data:

- **Textual data:** Includes product titles, detailed descriptions, and user reviews. These provide semantic insights into item characteristics.
- **User behavior data:** Such as browsing history, purchase records, and click sequences, capturing dynamic interest patterns.
- **Product metadata:** Includes categorical tags, attributes, and labels that support item classification and filtering.

These datasets are processed to extract embeddings and sequential patterns essential for training the Transformer models.

2.3.4 Proposed Methodology

Pre-trained Transformer models are utilized to process and understand textual data effectively, leveraging their deep contextual understanding to improve language-related tasks.

The figure bellow depicts the transformer model architecture.

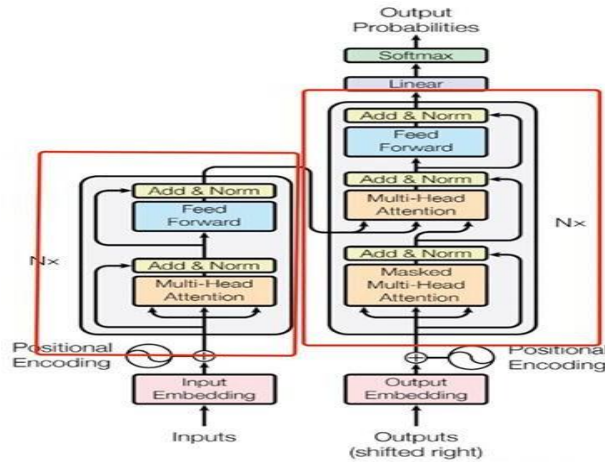


Figure 8: Transformer model architecture diagram [51]

To improve recommendation systems, this approach integrates Transformer models to better capture complex user behaviors and preferences. The method will be tested on Taobao, a leading Chinese e-commerce platform, to evaluate its scalability and effectiveness in real-world, high-traffic scenarios.

2.3.5 Methods, Approaches, and Algorithms Used

2.3.5.1 Methods Used

This method starts with data preprocessing (cleaning, vectorization, augmentation) to prepare high-quality inputs. Transformers then model user behavior sequences, which are integrated into the recommendation system. Finally, multi-task learning simultaneously predicts key metrics like CTR and conversion rate for optimized performance. The entire pipeline delivers powerful, personalized recommendations.

2.3.5.2 Algorithms Used

The proposed recommendation system leverages advanced deep learning techniques to enhance performance. Transformer architectures, including self-attention and multi-head attention mechanisms, are employed to analyze complex user-product interactions. A specialized Behavior Sequence Transformer (BST) variant is used to model temporal user behavior patterns. For text representation, embedding techniques like Word2Vec, GloVe, and BERT are applied. The system is benchmarked against traditional models such as Wide & Deep Learning (WDL) and Deep Interest Network (DIN). Finally, a sigmoid function ensures normalized predictions in the output layer.

2.3.6 Method Steps

2.3.6.1 Data Preprocessing

The text data undergoes cleaning and standardization, including the removal of special characters and stopwords, to ensure consistency. Next, the processed text is converted into dense vector representations using embedding techniques like Word2Vec, GloVe, or BERT. The data is then sequenced and padded to fit the input requirements of Transformer models. Finally, data augmentation techniques, such as random text modifications, are applied to enhance the robustness and generalization of the training dataset.

2.3.6.2 Building the Recommendation Model

The recommendation system follows a structured pipeline: User and product data is first encoded into embedding vectors to capture latent features. These embeddings are processed by a Behavior Sequence Transformer (BST) to model temporal interaction patterns through self-attention. A Multi-Layer Perceptron (MLP) is then integrated to capture non-linear feature interactions, enhancing recommendation accuracy. This hybrid approach effectively combines sequential modeling with deep learning for personalized recommendations.

2.3.6.3 Model Training

The system uses multi-task learning to predict both CTR and conversion rates simultaneously, while self-attention identifies relationships between clicked products for smarter recommendations.

2.3.6.4 Generating Recommendations

The trained model generates personalized recommendations by analyzing user behavior and preferences. Its performance is then benchmarked against other approaches to assess accuracy and effectiveness in real-world scenarios.

2.3.7 Method Advantages

The Transformer-based recommendation system delivers significant advantages over traditional approaches. By leveraging better sequence modeling, it effectively captures complex user interactions and behavioral patterns. This leads to improved recommendation accuracy, with measurable increases in both click-through rates (CTR) and conversion rates. The system provides advanced personalization by dynamically adapting to individual user behavior and contextual data. Additionally, it leverages semantic knowledge from textual product information to enhance recommendation relevance. Benchmark tests demonstrate that the Behavior Sequence Transformer (BST) with Transformer architecture consistently outperforms traditional models like Wide & Deep Learning (WDL) and Deep Interest Network (DIN) across key metrics.

2.3.8 Method Limitation

While Transformer-based recommendation systems offer significant advantages, they also present several key challenges. The approach requires high computational resources, as training Transformers and Behavior Sequence Transformers (BST) demands powerful GPU infrastructure and results in higher latency compared to classical recommendation systems. Additionally, these models depend heavily on high-quality data—large, clean datasets of user behavior and product text are essential for optimal performance, as noisy or sparse data can substantially degrade recommendation accuracy. Another critical limitation is generalization: the models may struggle with out-of-distribution scenarios, and inherent biases in training data can negatively influence recommendations, potentially leading to unfair or irrelevant suggestions.

2.4 Intelligent Classification and Personalized Recommendation of E-commerce Products Based on Machine Learning

This method combines natural language processing (NLP) and machine learning (ML) to improve classification and personalized recommendation of products in e-commerce platforms. It introduces a hybrid approach that utilizes the BERT language model for textual understanding and the k-Nearest Neighbors (k-NN) algorithm for identifying similar products, tailored to user preferences. [52]

2.4.1 Method Objective

The method aims to improve the user experience and decision-making process in e-commerce by:

- Addressing information overload caused by vast product catalogs.
- Enhancing the precision of recommendations through semantic understanding of product descriptions.
- Comparing traditional classification methods with more intelligent, personalized systems.

Proposing a hybrid solution that combines BERT for feature extraction and k-NN for similarity-based recommendation.

2.4.3 Input Data Type

The system uses diverse types of input data to build robust recommendations:

- **Product textual data:** Titles and detailed descriptions of items, which provide semantic richness.
- **User data:** Includes browsing history, purchase behavior, and explicit preferences.
- **Transactional data:** Purchase records and item categories help map user interests.
- **Product metadata:** Structured fields like categories, brands, and price ranges that support classification.

This data is cleaned, enriched, and transformed into embeddings for use in ML models.

2.4.4 Proposed Solution of the method

The system uses BERT to convert product text into semantic vectors, capturing nuanced meanings for better recommendations. This improves matching between user preferences and product features, especially for new items.

This figure appears to represent a structured matrix or grid, possibly related to a machine learning or data analysis process.

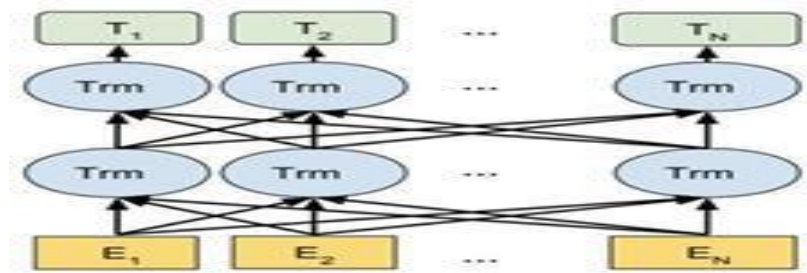


Figure 9: Training architecture diagram of BERT model [52]

The system implements k-NN algorithm to recommend similar products by analyzing purchase history patterns. This approach will be deployed and tested on eBay's platform to evaluate its real-world recommendation effectiveness and scalability.

2.4.5 Methods, Approaches, and Algorithms Used

2.4.5.1 Methods Used

The described methods combine structured data analysis and machine learning techniques to enhance product classification and recommendation systems. Traditional approaches rely on predefined attributes for basic categorization, while modern methods leverage user behavior, deep learning, and similarity-based algorithms to deliver personalized suggestions. Below is a breakdown of the key methodologies and their roles:

- Traditional product classification: Using structured attributes for initial categorization.
- Personalized recommendation: Based on user behavioral data.
- Text processing with BERT: Extracting deep semantic meaning from product descriptions.
- Similarity-based retrieval with k-NN: Finding and recommending the most similar items to past purchases.

2.4.5.2 Algorithms Used

This method used:

- **BERT (Bidirectional Encoder Representations from Transformers):** For deep text understanding and semantic vector generation.
- **Masked Language Model (MLM):** A BERT pretraining strategy that improves text prediction and context capture.
- **Next Purchase Prediction (NPP):** Predicts a user's future purchases based on past session patterns.
- **k-Nearest Neighbors (k-NN):** Recommends products based on proximity in the embedding space.
- **Data preprocessing techniques:** Cleaning, imputation, tokenization, and embedding.

2.4.6 Method Steps

2.4.6.1 Data Preprocessing

The text data first undergoes cleaning by removing special characters and stopwords for consistency. Next, BERT embeddings transform product descriptions into rich semantic vectors that capture contextual meaning. Finally, these embeddings are merged with other product information (like categories or attributes) to create comprehensive, high-quality input features for the recommendation model. This preprocessing pipeline ensures the system leverages both textual and structural product data effectively.

2.4.6.2 Model Training

To further refine the recommendation system, state-of-the-art natural language processing (NLP) and predictive modeling techniques are employed. These methods leverage deep learning to understand user intent and anticipate future behavior, ensuring recommendations are both contextually relevant and forward-looking. The key steps include:

- Apply Masked Language Model (MLM) to fine-tune BERT on e-commerce data.
- Use Next Purchase Prediction (NPP) to predict future purchases.
- Train on an eBay dataset to validate the model.

2.4.6.3 Product Recommendation

The system uses BERT to create product vectors, then applies k-NN to find similar items based on purchase history, generating personalized recommendations.

2.4.6.4 Performance Evaluation

The system evaluates recommendation quality using Mean Average Precision at 12 (MAP@12), comparing its performance against traditional classification models to measure improvements. Additionally, manual validation ensures the practical relevance and accuracy of the recommendations, providing an extra layer of quality assurance.

2.4.7 Method Advantages

BERT boosts recommendations with smarter product matching, cuts cold-start issues, and scales for large catalogs. It personalizes suggestions while reducing returns and carbon footprint.

2.4.8 Method Limitation

While BERT-enhanced recommendation systems offer clear advantages, they also come with notable challenges. The approach requires substantial computational resources, as BERT demands powerful hardware for both training and real-time inference, resulting in higher latency compared to traditional recommendation methods. Additionally, these systems depend on large volumes of high-quality data extensive text descriptions and user transaction histories are essential for accuracy, while biased or incomplete training data can skew results. Another key limitation is explainability: BERT's complex architecture makes it difficult to interpret why specific recommendations are generated, posing transparency issues for users and businesses alike.

2.5 Leveraging Large Language Models to Enhance Personalized Recommendations in E-commerce [53]

This method explores the integration of Large Language Models (LLMs) such as BERT and GPT-like models into hybrid recommendation systems. It emphasizes enhancing both the accuracy and the diversity of recommendations by fusing semantic understanding with collaborative and content-based techniques. The goal is to deliver intelligent, personalized suggestions that better reflect users' evolving preferences.

2.5.1 Method Objective

The method aims to address several key limitations of classical recommendation algorithms:

- The sparsity of interaction data, especially in new or niche product categories.
 - The cold start problem, where new users or products lack historical data.
 - The lack of diversity in conventional recommendations, often caused by repetitive suggestions.
 - The difficulty in understanding implicit user needs using traditional techniques.
- By combining LLMs with traditional methods, the model leverages natural language data to make more informed and adaptable recommendations.

2.5.2 Input Data Type

The method relies on four main categories of data:

- **User behavior data:** Includes clicks, searches, and purchases that reveal user intent.
- **Product data:** Textual descriptions, structured attributes, and product categories.
- **Contextual data:** Environmental and session-based variables like time, device, and location.
- **Textual data:** User-generated content such as reviews and feedback, which is processed to extract insights.

These inputs are normalized, embedded, and integrated into a multi-source recommender framework.

2.5.3 Proposed Solution of the method

To address the limitations of pure BERT-based systems while maintaining its strengths, we propose a hybrid recommendation framework. This approach intelligently combines large language models (LLMs) with traditional content-based filtering (CBF) and collaborative filtering (CF) techniques. The system leverages semantic analysis of product descriptions and reviews to better understand item characteristics and user preferences. It further enhances personalization by incorporating contextual signals like time, location, and browsing behavior. To prevent recommendation monotony, the model architecture includes diversity optimization through specialized loss function constraints. This balanced methodology aims to

deliver accurate yet explainable recommendations while mitigating computational overhead through strategic model hybridization.

This figure outlines the **user journey** and **backend processes** of an e-commerce recommendation system. Here's a step-by-step breakdown:

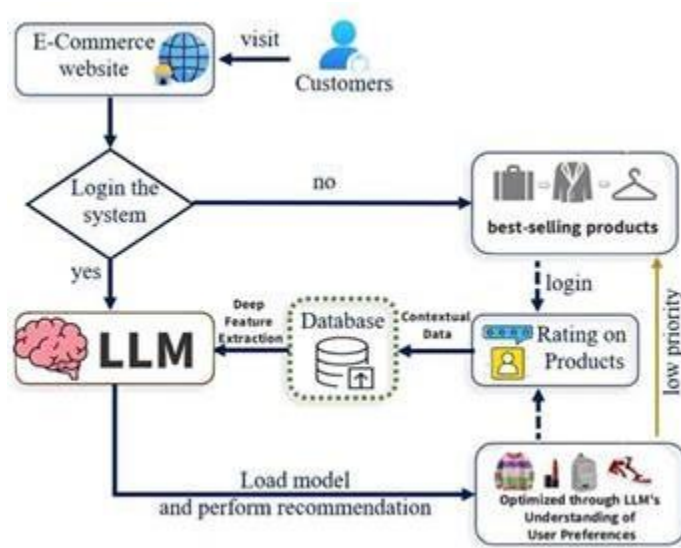


Figure 10: Application of LLM in e-commerce platform recommendation system [53]

2.5.4 Methods, Approaches, and Algorithms Used

2.5.4.1 Methods Used

The recommendation system combines multiple approaches for comprehensive performance.

- **Content-Based Filtering (CBF):** Analyzes user preferences using item features.
- **Collaborative Filtering (CF):** Utilizes interactions between similar users and products.
- **Semantic enrichment using LLMs:** Enhances understanding of products and user intent through NLP.
- **Hybrid modeling:** Merges multiple scoring systems into a unified output.

2.5.4.2 Algorithms Used

The system uses pre-trained LLMs to encode text into embeddings, measures similarity via cosine distance, and employs multi-task learning to optimize CTR, conversions, and diversity through custom loss functions.

2.5.5 Method Steps

2.5.5.1 Data Preprocessing

The system begins by cleaning both textual data (product descriptions, reviews) and user interaction data to ensure consistency and remove noise. Next, it normalizes the processed data (scaling numerical features, standardizing text) and extracts key features, including keywords and semantic embeddings, to create structured inputs for the recommendation models. This preprocessing pipeline ensures high-quality data for accurate and efficient recommendations.

2.5.5.2 Recommendation Model Fusion

The system generates individual recommendation scores using content-based filtering (CBF), collaborative filtering (CF), and LLM-based semantic analysis. These scores are then weighted and combined balancing item relevance, user similarity, and contextual understanding to produce a final, optimized recommendation list.

5.5.5.3 Optimization and Personalization

The model is trained using a custom loss function that explicitly optimizes for recommendation diversity alongside accuracy, preventing repetitive suggestions. It also dynamically adapts outputs in real-time based on evolving user context including current session behavior, device, time, and location to maintain personalized relevance throughout the user journey. This dual approach ensures recommendations stay both varied and precisely tailored to immediate needs.

2.5.5.4 Performance Evaluation

the system's performance is rigorously evaluated using multiple metrics to ensure comprehensive assessment:

- Metrics used: Precision, recall, F1-score, CTR, diversity of recommendations.
- Compare performance with traditional models (collaborative and content-based filtering alone).

2.5.6 Method Advantages

the hybrid recommendation system delivers measurable improvements across key performance indicators. It achieves higher precision and recall by combining multiple signals, resulting in more accurate suggestions that better match user preferences. The system actively increases recommendation diversity through constrained optimization, preventing repetitive or narrow suggestions. By leveraging NLP-powered analysis of unstructured data (reviews, queries), it detects subtle user needs that traditional methods miss. The integration of LLMs helps overcome data sparsity by extracting insights from product descriptions when interaction data is limited. Finally, its dynamic personalization continuously refines outputs based on real-time behavior (clicks, dwell time) and contextual signals (location, device), ensuring recommendations remain relevant throughout the user journey.

2.5. Method Limitation

While the hybrid recommendation system offers significant advantages, it also faces several key challenges. The approach incurs high computational costs, as large language models (LLMs) demand powerful GPUs and extended training times, making deployment resource-intensive. Additionally, the system heavily depends on high-quality textual data such as detailed product descriptions and user reviews and may underperform when such data is scarce, incomplete, or noisy. Another critical limitation is limited explainability: the complex decision-making processes of LLMs can be opaque, making it difficult to provide clear justifications for why specific products are recommended, which may reduce user trust and hinder debugging efforts.

2.6. Graph Neural Network Recommendation System for Football Formation [54]

This method introduces an innovative recommendation system that applies Graph Neural Networks (GNNs) and Reinforcement Learning (RL) to optimize football team formations. Instead of relying solely on statistical summaries, it models a football match as a graph of player interactions, such as passes and positioning, to derive tactical recommendations. This system provides a new perspective on how data-driven intelligence can support coaching decisions and improve team performance

2.6.1 Method Objective

The key objectives of this method are:

- To analyze the interactions between players during matches and identify key structural patterns.
- To model football formations as graphs, where nodes represent players and edges represent passes or tactical relations.
- To apply GNNs for learning from these graphs and extract optimal formations.
- To use Reinforcement Learning to dynamically adjust player positions and team structure in various scenarios.
- To assist coaches and analysts by offering AI-powered insights into team coordination and effectiveness.

2.6.2 Input Data Type

This method processes a variety of match-related data:

- Match data: Includes passing history, ball possession statistics, and movement across zones.
- Player data: Positions on the field, individual metrics (e.g., speed, pass accuracy), and interactions.
- Contextual data: Match type (friendly, league, knockout), opponent strength, game context (score, time).
- Passing network graphs: Represent players as nodes and passes as directed edges, forming a tactical graph for each team.

This data is transformed into graph structures to be used as input to GNNs.

2.6.3 Proposed Solution of the method

The system models the football team as a graph, with players represented as nodes and passes as edges, to analyze team dynamics. Clustering techniques identify key players and enable dynamic formation adjustments based on real-time performance. Graph Neural Networks (GNNs) extract optimal gameplay patterns from historical data, enhancing tactical strategies. Finally, reinforcement learning optimizes formations by simulating match scenarios, improving in-game decision-making for better outcomes.

2.6.4 Methods, Approaches, and Algorithms Used

2.6.4.1 Methods Used

The framework begins by representing match data as a directed graph, where players are nodes and passes form edges between them. Clustering coefficient analysis then quantifies each player's centrality, revealing key playmakers and high-density interaction zones. Graph Neural Networks (GNNs) process these patterns to detect optimal player synergies and formation templates. Finally, Reinforcement Learning (RL) iteratively evaluates and refines formations by simulating match scenarios, using performance-based reward signals to guide tactical adjustments [Samir Elhedhli, Zichao Li]. This end-to-end approach transforms raw pass networks into actionable insights for dynamic strategy optimization.

2.6.4.2 Algorithms Used

The system employs Graph Neural Networks (GNNs) to analyze player interactions, learning both individual node (player) and team-level representations to assess roles and importance. Clustering coefficient metrics identify key hubs by detecting triangular pass patterns, highlighting players who frequently connect teammates. Semi-supervised learning enhances the model's adaptability by leveraging both labeled (e.g., coach annotations) and unlabeled (raw match data) inputs. Finally, reinforcement learning dynamically optimizes formations through simulated match scenarios, enabling real-time tactical adjustments based on performance feedback. Together, these techniques transform raw data into actionable insights for adaptive strategy development.

2.6.4.3 Method Steps

2.6.4.3.1 Data Preprocessing

The system extracts player interactions and passing sequences from match data, then constructs a passing network graph to model team dynamics. In this graph, players are represented as nodes, while passes between them form directed edges, capturing the flow and structure of gameplay. This graph-based representation enables quantitative analysis of team cohesion, player roles, and tactical patterns.

2.6.4.3.2 Analysis of Passing Relationships

The system calculates clustering coefficients to quantify each player's connectivity, revealing central playmakers who frequently participate in triangular passing patterns. Simultaneously, it detects tactical subgroups by analyzing dense passing clusters, uncovering natural

partnerships or positional units within the team's overall structure. These metrics help coaches objectively identify key influencers and evaluate team organization.

2.6.4.3.3 Training GNNs

The system trains a Graph Neural Network (GNN) on historical match data to learn patterns of effective team formations, capturing how player interactions influence performance. Once trained, the model dynamically adjusts its formation recommendations by factoring in real-time inputs like opponent tactics, match context (e.g., scoreline, time remaining), and environmental conditions (e.g., home/away). This enables adaptive strategy planning tailored to specific game scenarios.

2.6.4.3.4 Reinforcement Learning Optimization

The system leverages reinforcement learning to simulate thousands of formation and strategy variations, identifying high-performing configurations through iterative trial-and-error. During matches, it dynamically adjusts player positioning by processing real-time data (pass accuracy, opponent movements, fatigue levels) to optimize tactical responses. This closed-loop approach merges pre-match planning with in-game adaptability.

2.6.5 Method Advantages

The AI system analyzes player interactions to optimize formations using real match data. It dynamically adapts to game conditions, offering coaches data-driven insights while reducing human bias.

2.6.6 Method Limitation

While AI-driven football analytics offer powerful insights, they come with notable challenges.

- High Computational Complexity:

GNN models and reinforcement learning require significant computational power.

- Dependence on Match Data:

Requires detailed match data (passing, positioning, tactics).

Less effective with incomplete or noisy data.

- Limited Explainability:

Coaches may struggle to interpret certain AI-generated suggestions.

Needs visualization tools for better result interpretation.

2.7 Conclusion

In this chapter, we explored several cutting-edge recommendation systems built upon modern AI methods such as Reinforcement Learning, Transformers, Large Language Models, and Graph Neural Networks. Each method addresses critical challenges in personalization, sparsity, semantic understanding, or dynamic adaptation.

The use of AI in recommendation systems brings significant advantages:

- Improved accuracy and user satisfaction.
- Dynamic and personalized experiences.
- Adaptability to context and changing behavior.
- Ability to learn from rich, multimodal data (text, graphs, sequences).

Despite some limitations mainly related to computational cost and interpretability these intelligent models demonstrate strong potential to transform how decisions are made in domains ranging from e-commerce to sports analytics.

In the next chapter, we will present Architecture Description for your Smart Application for Personal Food Catering Service (NB-Delights), combining both technical and functional components, suitable for documentation or presentation.

CHAPTER 03

CONCEPTION OF PERSONAL FOOD CATERING

SERVICE (NB-DELIGHTS)

3.1 Introduction

Personalized catering services have become a growing demand, especially for special events such as birthdays, graduations, weddings, and corporate gatherings. Clients no longer seek generic food options they desire tailor-made menus that align with their tastes, dietary restrictions, cultural preferences, and event themes. To meet this need, we propose NB-Delights, a Smart Application for Personal Food Catering Service, integrating intelligent systems, real-time interaction, and modern delivery solutions.

We propose an operational model of the system that integrate IA methods in (Figure 10).

3.2. NB-delights Model Architecture

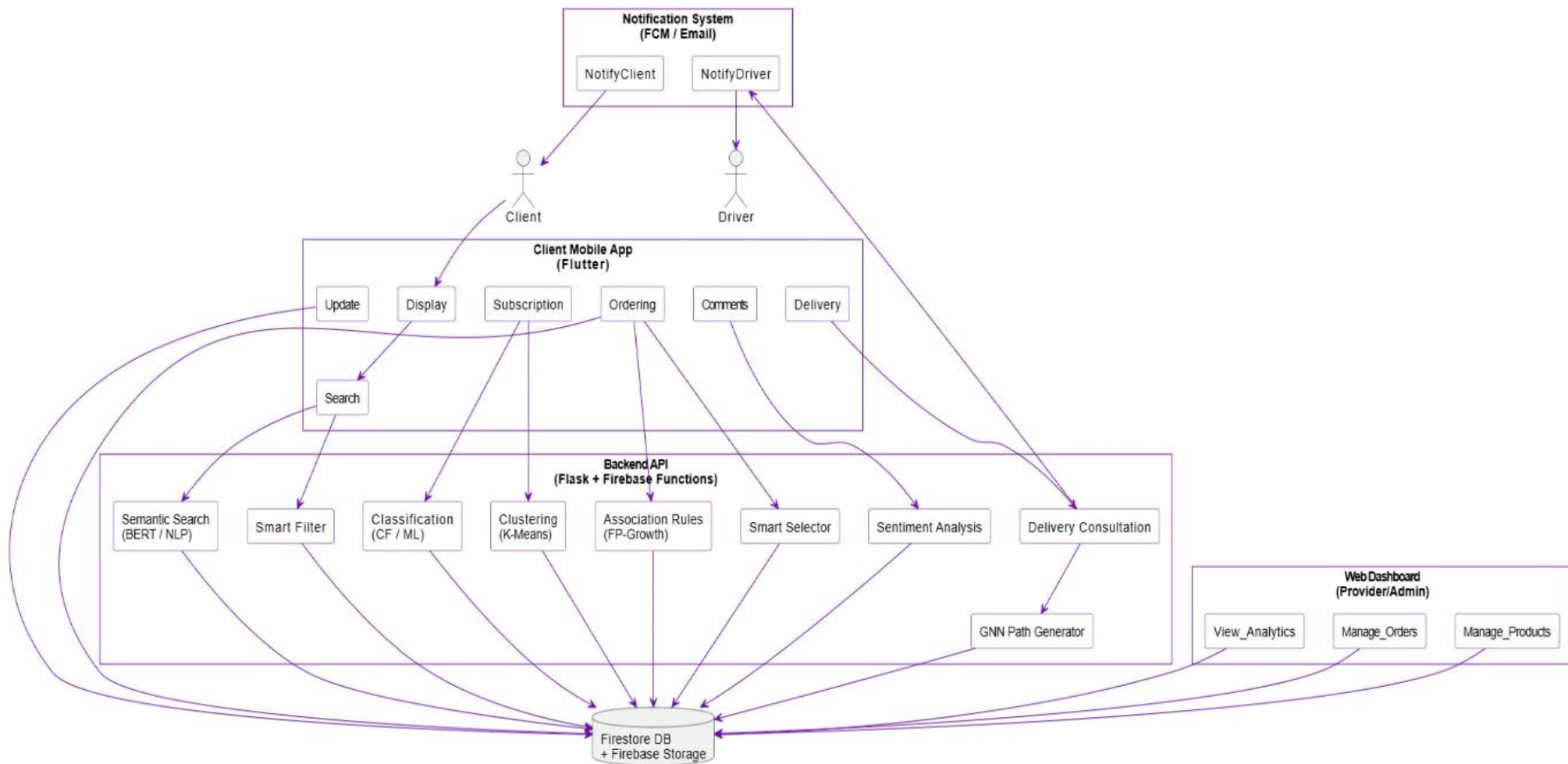


Figure 11: general architecture of NB-delights

The purpose of this architecture is to leverage the complementary information of NB-DELIGHTS system, is a multi-platform application:

- Mobile for clients
- Web for provider/admin
- Backend for IA methods
- Firestore for data base

Providing smart catering solutions tailored for diverse events, the system integrates artificial intelligence, cloud storage, and real-time databases to enable personalized food recommendations, streamlined ordering processes, and effective service coordination.

3.3. NB-delights Class diagram

A class diagram provides a global view of NB-DELIGHTS system by presenting its classes, interfaces and relationships between them. Class diagrams are static: they show what interacts but not what happens during interaction [55].

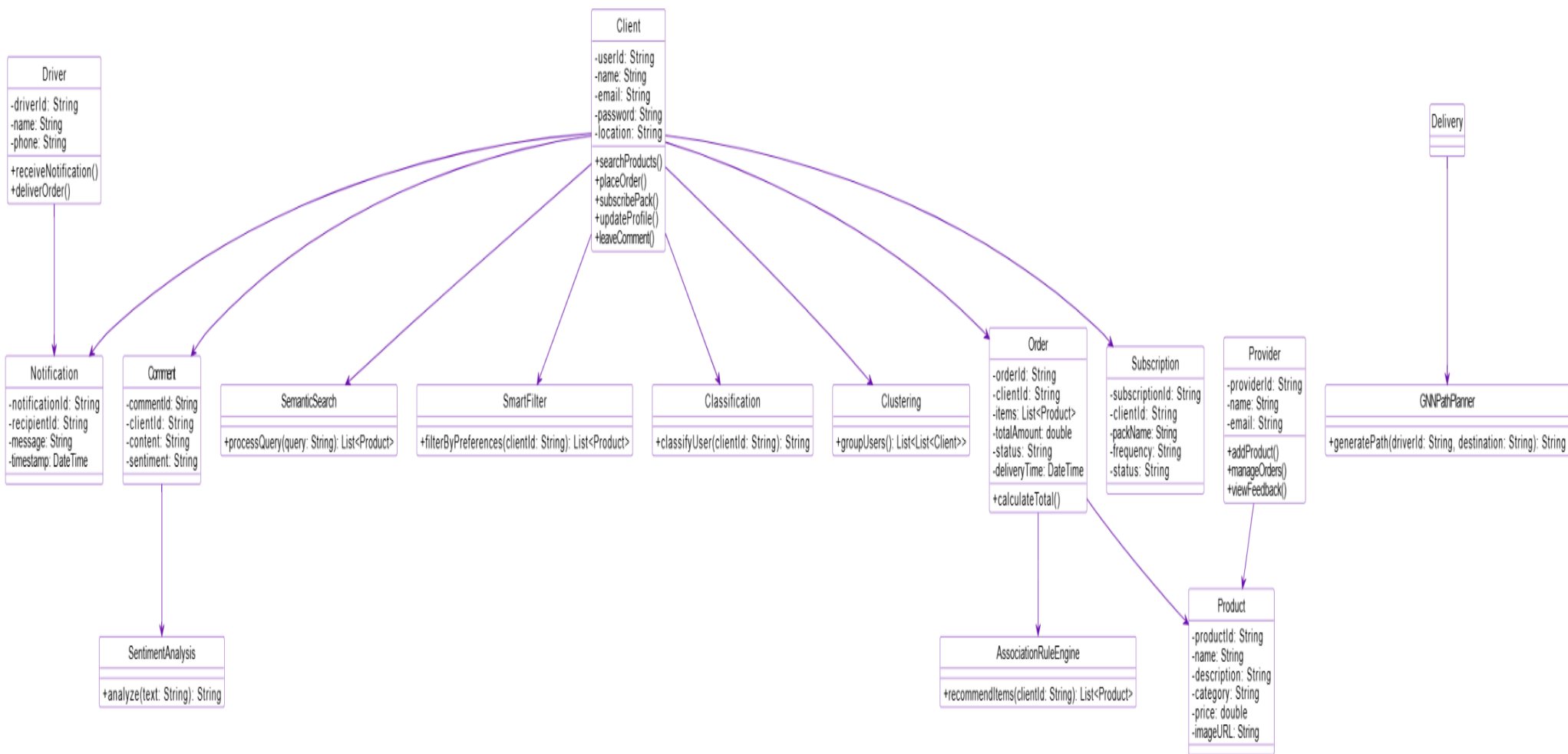


Figure 12: NB-delights Class diagram

3.3.1 Client Mobile Application

This is the user-facing side of the system, the clients interact with all app features: searching, subscribing, ordering, updating, commenting, and tracking delivery.

3.3.1.1 Sequence Diagrams

Sequence diagrams are a popular dynamic modeling solution in UML language, as they focus more precisely on lifelines, processes and objects that live simultaneously, and the messages they exchange with each other to perform a function before the end of the lifeline. [56]

3.3.1.1.1 Display Sequence Diagram

- Initial screen showing menus, promotions, or categories.
- Connects users to different functionalities like search or subscription.

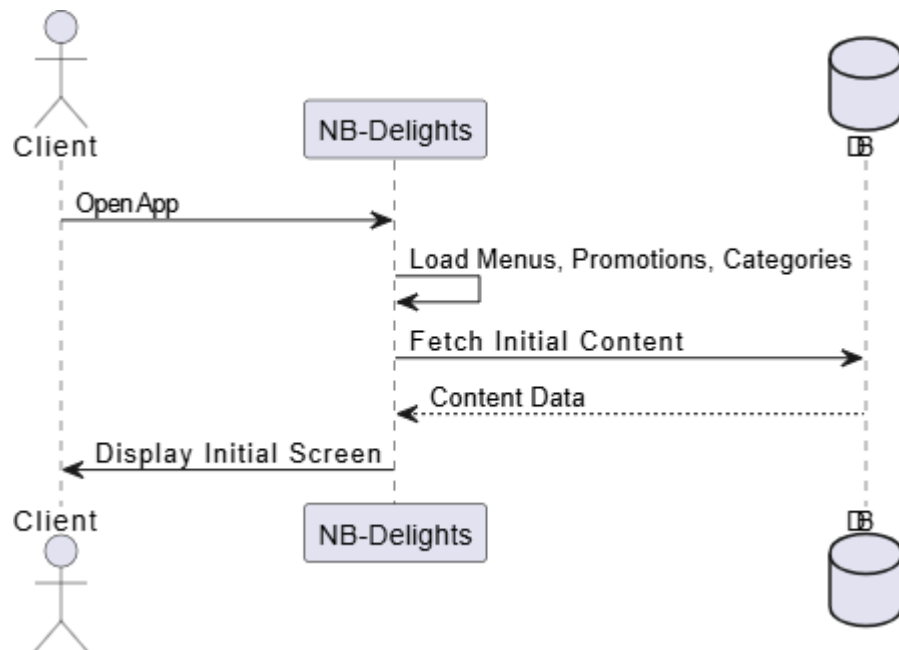


Figure 13: display Sequence diagram

3.3.1.1.2 Search Sequence Diagram

- Lets users look for products using semantic search powered by NLP (e.g., BERT).
- Sends user queries to the backend AI and data base.

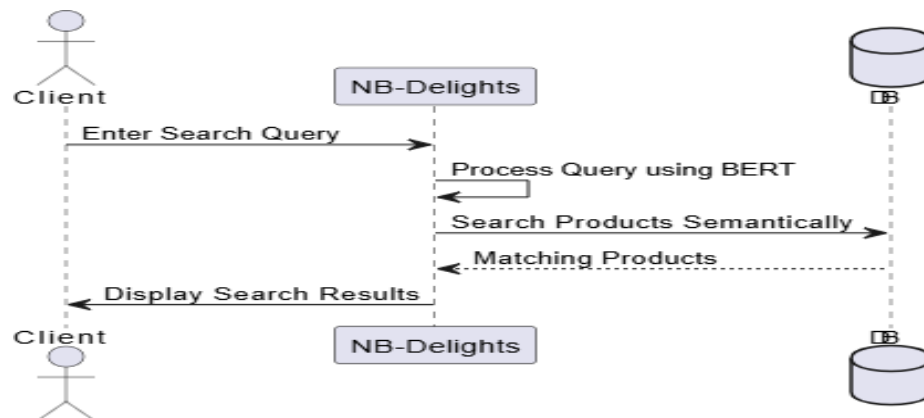


Figure 14: search Sequence diagram

3.3.1.1.3 Subscription Sequence Diagram

- Allows users to subscribe to pre-designed event packs or food plans.
- Uses AI to classify preferences and cluster users (K-Means, Collaborative Filtering).

2025-06-26 22:02:18

The research is without LOGIN

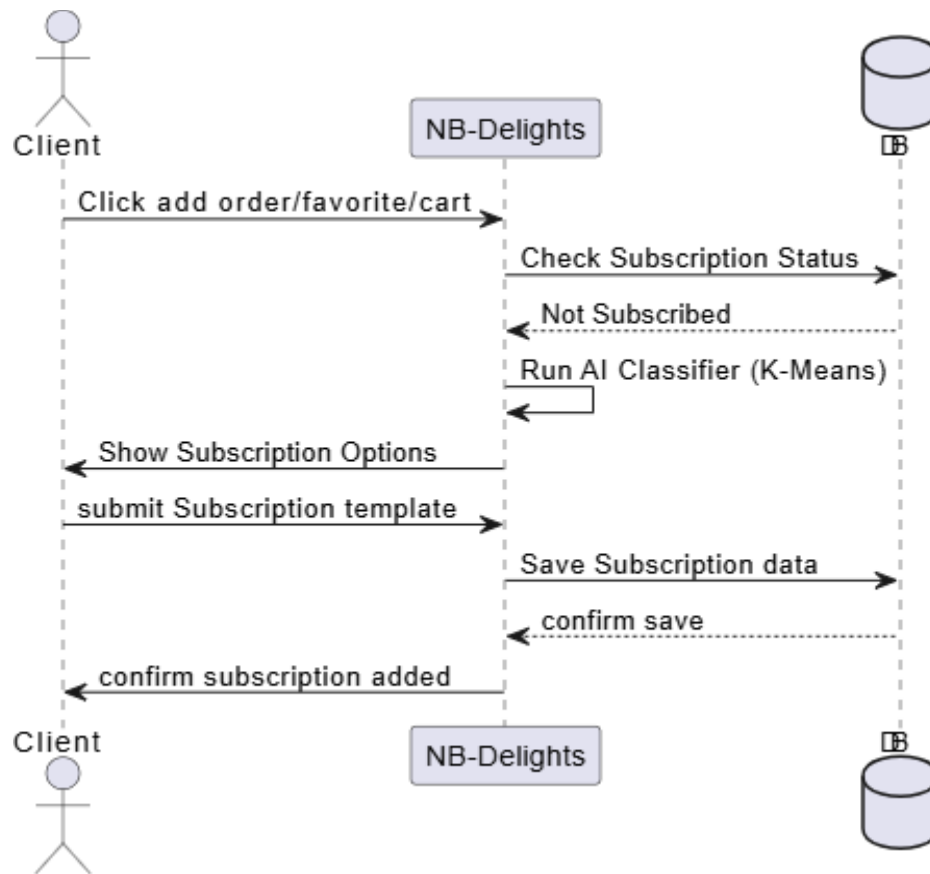


Figure 15: subscription Sequence diagram

3.3.1.1.4 Ordering Sequence Diagram

- Main ordering feature where users add items to their cart and confirm purchases.
- Integrates Association Rule Mining and Smart Selector AI to suggest bundles or related products.



Figure 16: ordering Sequence diagram

3.3.1.1.5 Update Sequence Diagram

- Enables users to update their profile, cart, preferences, or events.
- All updates are synchronized with the backend and stored in firestore.

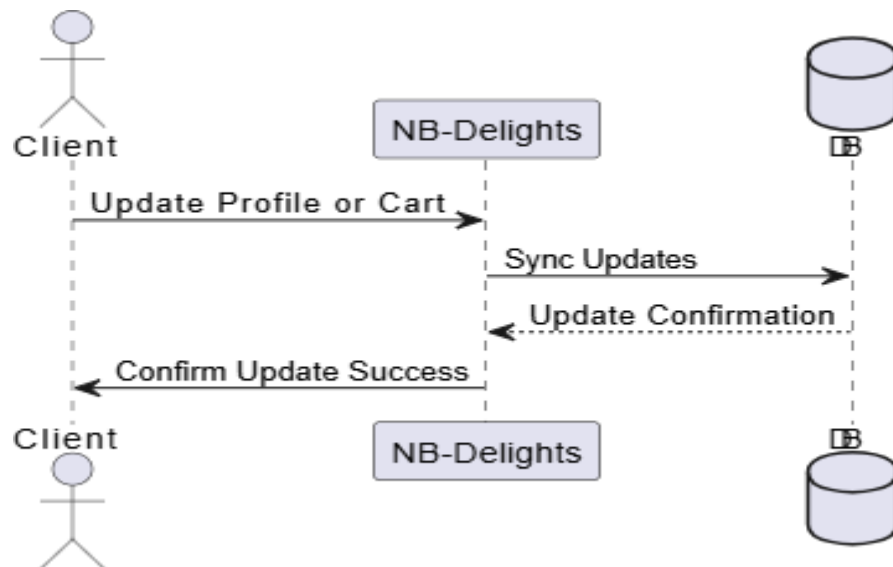


Figure 17: update Sequence diagram

3.3.1.1.6 Delivery Sequence Diagram

- Initiates the delivery flow after an order is placed.
- Communicates with backend AI to calculate delivery options and trigger notifications

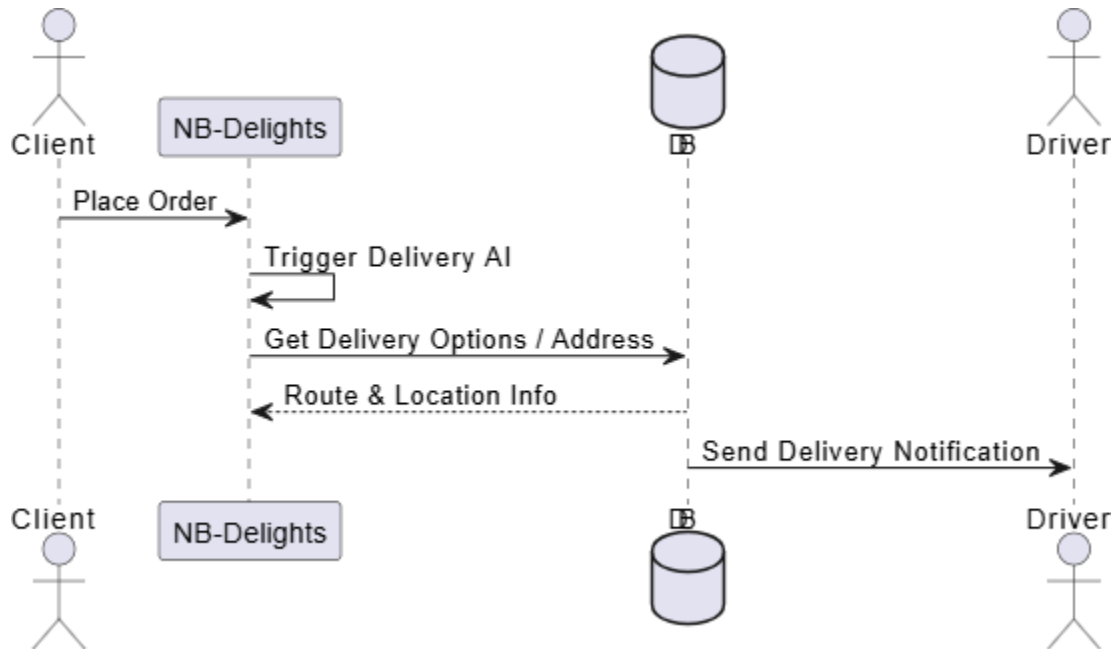


Figure 18: delivery Sequence diagram

3.3.2 Backend API

This is the brain of NB-DELIGHTS system:

3.3.2.1 Semantic Search

Understands and processes search queries in natural language to return relevant results used IA methods BERT and NLP.

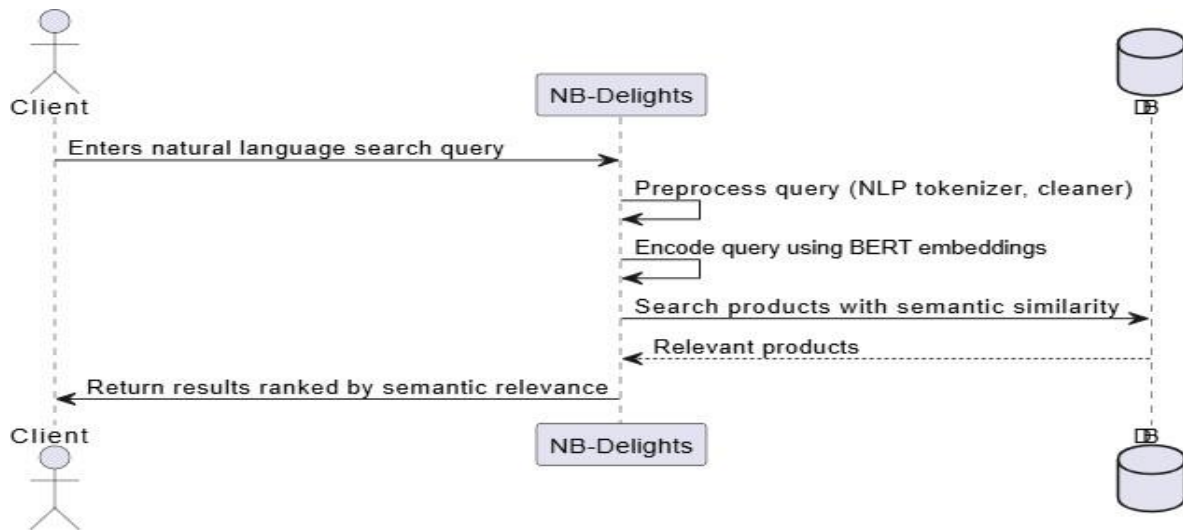


Figure 19: semantic search diagram

3.3.2.2 Smart filter

Filters results based on user preferences, dietary restrictions, past behavior, etc. used collaborative filtering Method with cosine similarity

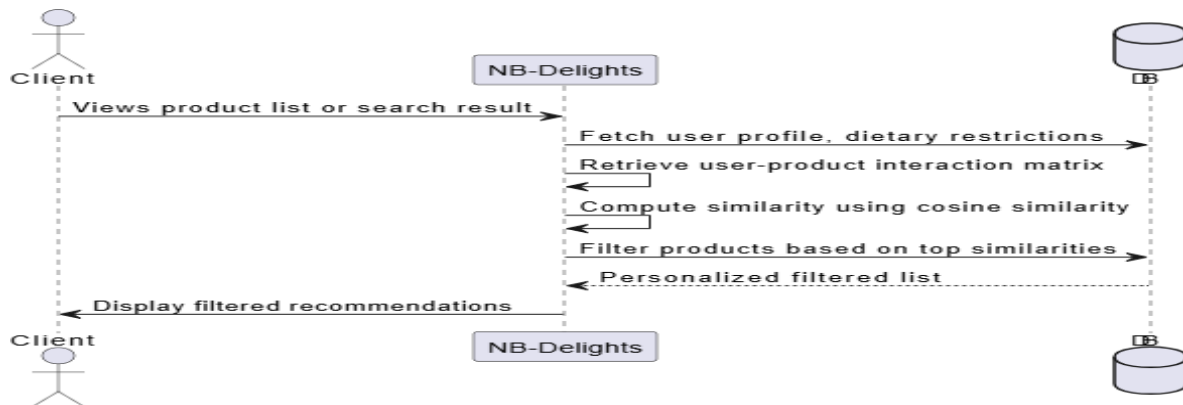


Figure 20: smart filter diagram

3.3.2.3 classification and clustering

Classifies user types, event categories, or product themes to generate smarter results used in methods collaborative filtering with cosine similarity and Segments users based on behavior or preferences for better personalization using K-means.

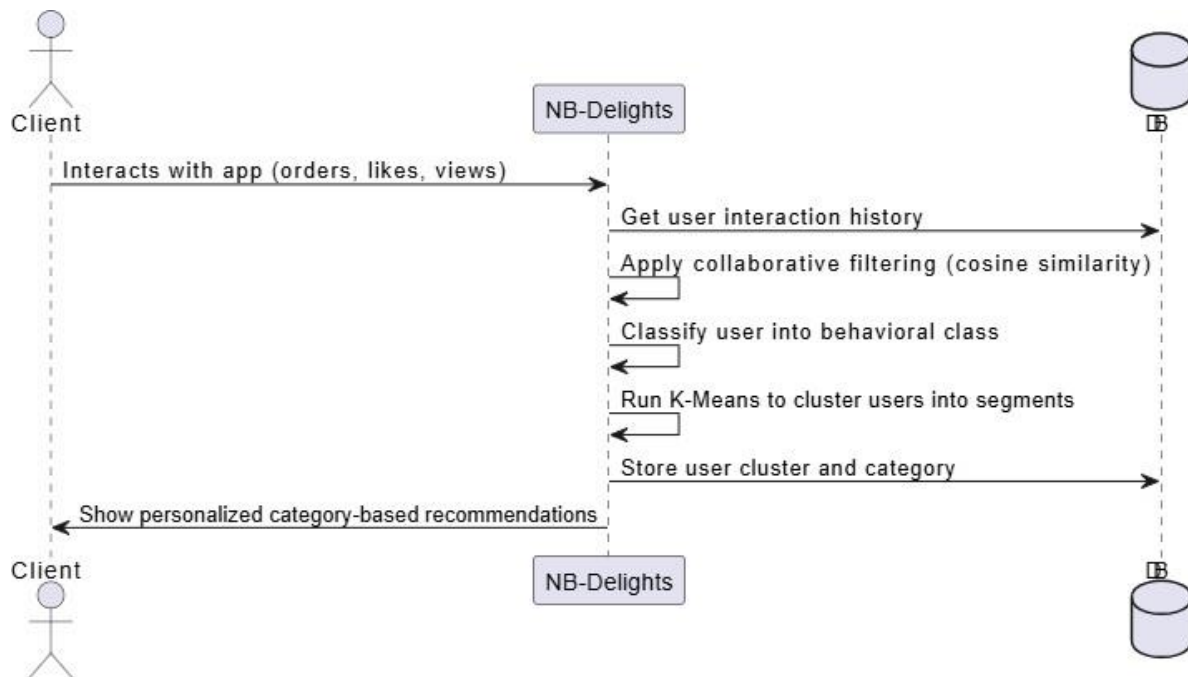


Figure 21: classification and clustering diagram

3.3.2.4 Smart Selector

Chooses the most appropriate product or pack based on AI analysis LSTM and Finds frequent item combinations to recommend popular menus or bundles used FP-growth.

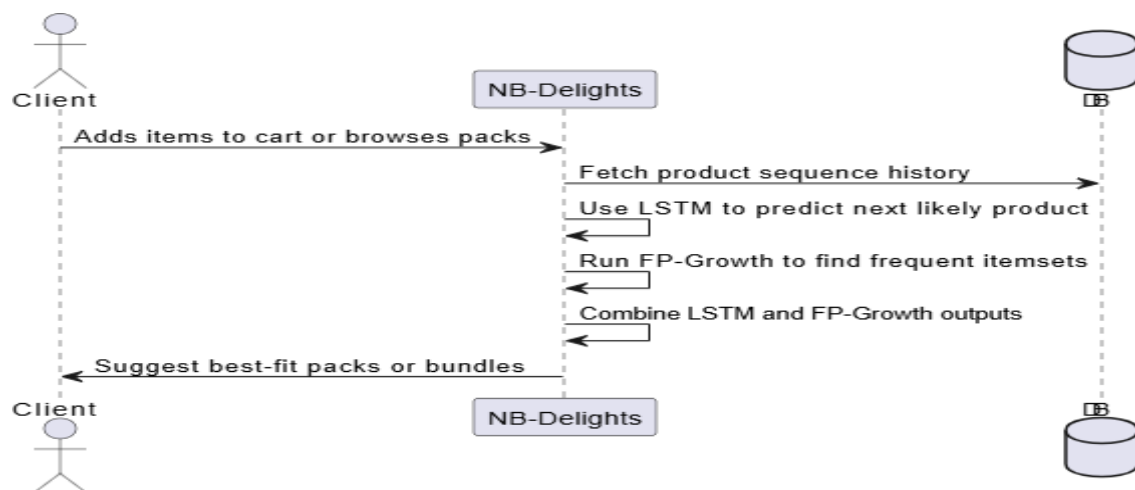


Figure 22: smart selector diagram

3.3.2.5 Delivery Planification

Connects delivery data, order status, and path optimization to provide real-time logistics insights Used GNN to optimize delivery routes.

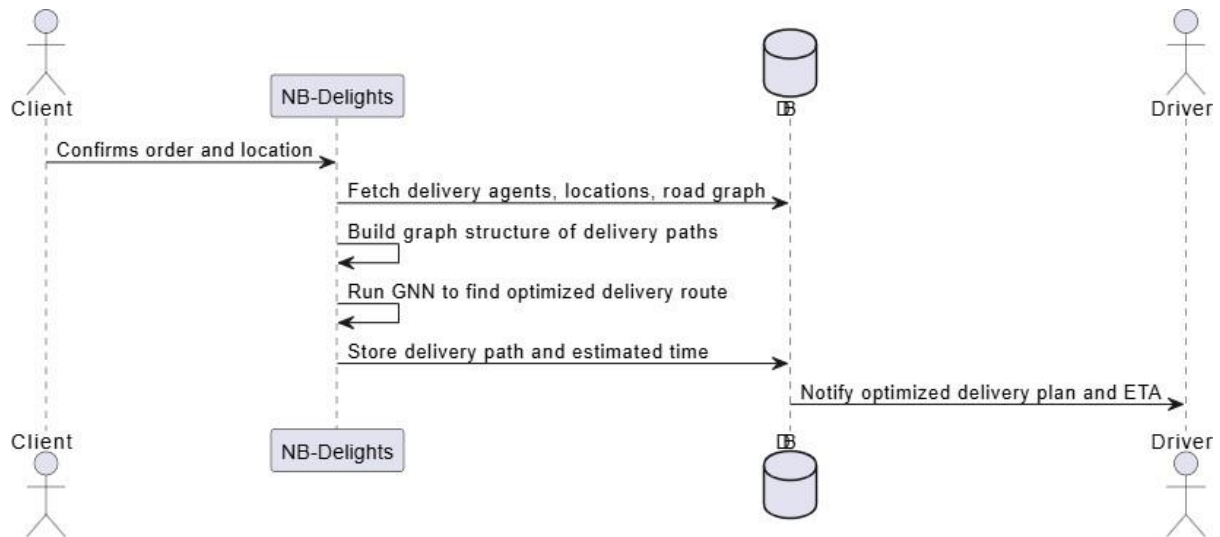


Figure 23: delivery planification diagram

3.3.3 Data base

This simplified diagram demonstrates how the NB-Delights system integrates with Firebase components. The client, provider, and driver interfaces interact with Firebase services to manage authentication, store and retrieve data, handle media, and enable push notifications — ensuring real-time, secure, and scalable communication across the system.

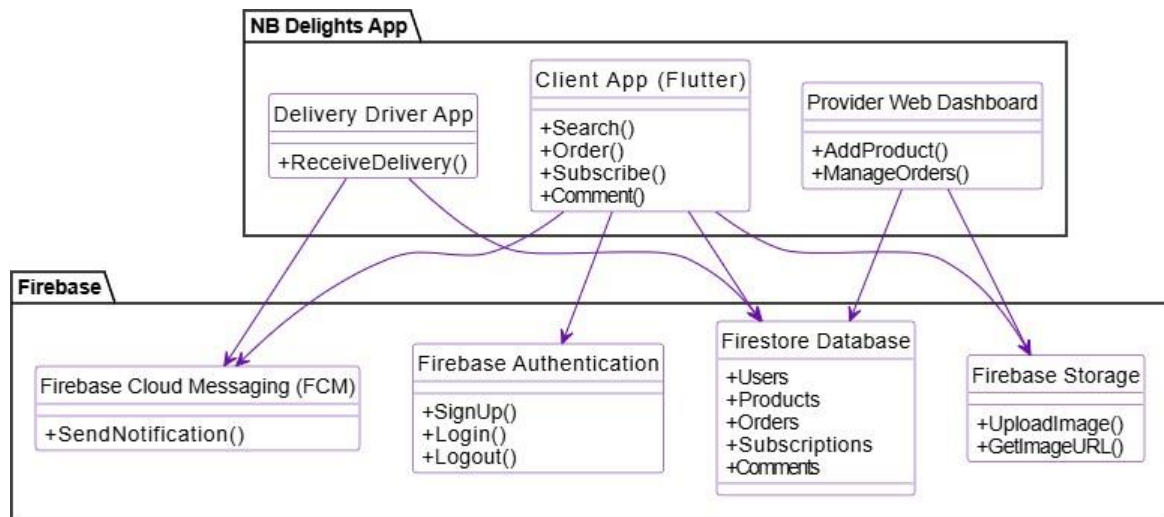


Figure 24: firebase diagram

3.3.4 Use case Diagram

Use case diagrams (DCU) are UML diagrams used for a representation of the functional behavior of a software system. They are useful for presentations to management or project stakeholders [57]

3.3.4.1 Provider Web Dashboard Use Case Diagram

Used by service providers and admins to manage content and monitor operations

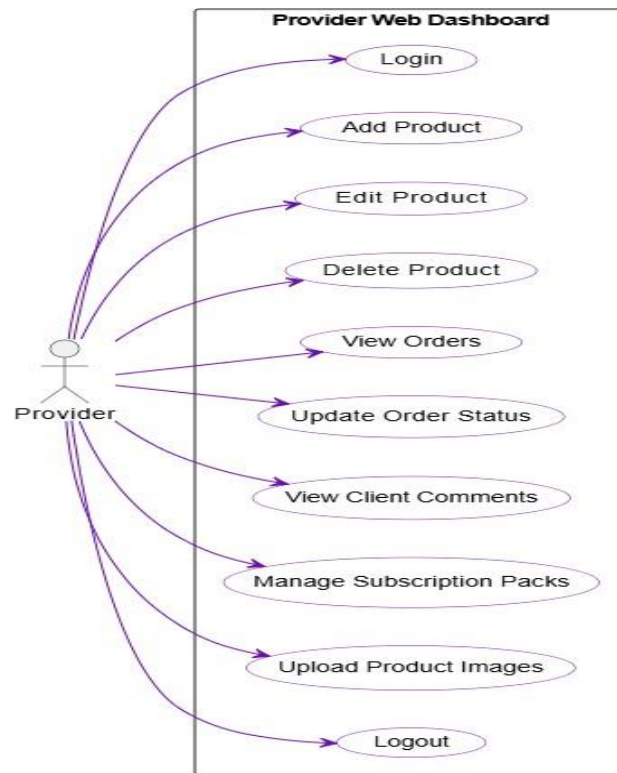


Figure 25: provider use case diagram

3.3.5 Notification System

➤ **NotifyClient**

Sends order confirmations, delivery alerts, and promotions.

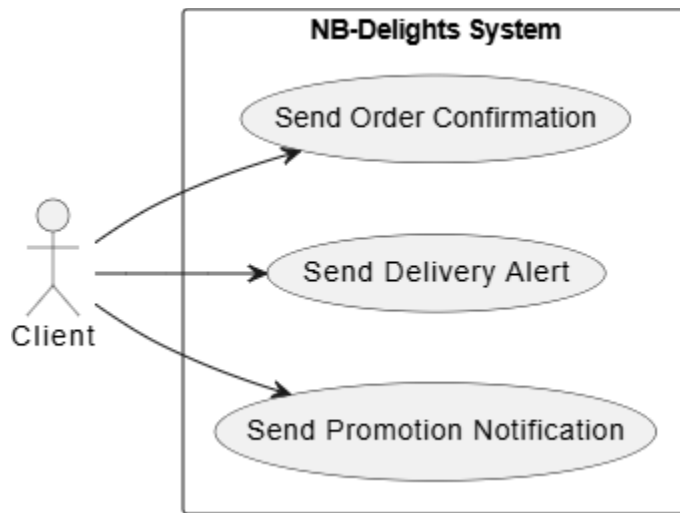


Figure 26: notify client use case diagram

➤ **NotifyDriver**

Notifies drivers of new deliveries or updates in the delivery path.

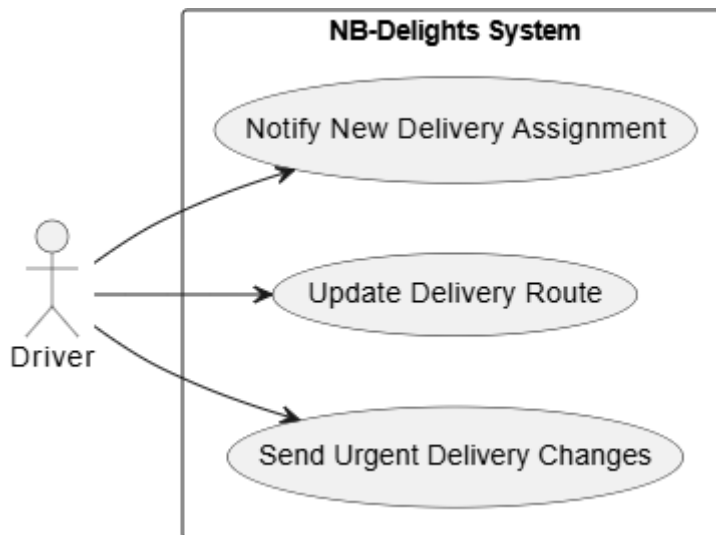


Figure 27: notify driver use case diagram

3.4. Conclusion

In this chapter, we have proposed model consists of three main components: the client mobile app, the provider web dashboard, and the delivery interface, all connected through Firebase services and supported by AI modules like recommendation and semantic search.

The goal of this model is to offer a smart, personalized catering service by simplifying order management, enhancing user experience, and automating provider and delivery tasks.

In the next chapter, we will present the implementation of the system, including the technologies used and the integration of AI to bring the proposed functionalities to life.

CHAPTER 04

IMPLEMENTATION OF PERSONAL FOOD

CATERING SERVICE (NB-DELIGHTS)

4.1 Introduction

This chapter presents the practical implementation of the NB Delights system. After designing the architecture and defining the main functionalities, this part focuses on the technologies used, the development environment, and the concrete realization of the mobile and web applications. We also detail the integration of artificial intelligence (AI) components and Firebase services.

4.2. Development environment

4.2.1 Development Software

- **Visual Studio Code:** A powerful and lightweight source-code editor developed by Microsoft, supporting a wide range of programming languages. It is used as the primary IDE for Flutter (Dart) and Flask (Python) development [58]
- **PowerPoint:** Used for creating visual presentations, including UI/UX flows and project overviews [59]
- **Microsoft Word:** Employed for writing and formatting technical documentation, project reports, and business plans [60]
- **Mockup Tools:** Utilized to create mobile UI mockups for early-stage design visualization.
- **Adobe Illustrator:** A vector graphics editor used for designing the application logo, branding, and visual elements [61]

4.2.2 Modeling Software

- **Draw.io:** An online diagramming tool used to model system architecture, use case diagrams, and flowcharts. It is particularly useful for visually representing components of the mobile app and backend integration. [62]

4.2.3 Programming Languages

- **Dart:** The primary programming language used in developing the mobile application through the Flutter SDK [63]
- **Python:** Used for building the Flask-based backend API and implementing AI models such as BERT for semantic search [64]
- **JavaScript:** Applied in the admin web interface or for enhancing front-end interactivity [65]
- **HTML/CSS:** Used to structure and style web pages, particularly in admin dashboards or documentation sites [66]

4.2.4 Frameworks and Tools

- **Flutter:** A cross-platform UI toolkit developed by Google, used to build the client-side mobile application for NB Delights [67]
- **Flask:** A lightweight Python web framework employed to develop the backend API. It serves the AI models and communicates with Firestore [68]
- **Firebase:** A Backend-as-a-Service (BaaS) platform used for authentication, real-time database (Cloud Firestore), cloud storage, and app analytics [69]
- **Firebase Admin SDK:** Integrated into the Flask backend to securely access and manage Firestore data from the server side [70]

4.2.5 Libraries and Dependencies

- **transformers (HuggingFace):** Utilized to load and use pre-trained BERT models for natural language understanding in product search [71]
- **torch:** Core PyTorch library used to handle tensor computations and run deep learning models [72]
- **scikit-learn:** Employed for cosine similarity computations in recommendation and search tasks [73]
- **firebase-admin:** Allows secure server-side interaction with Firebase services such as Firestore [74]
- **numpy:** A fundamental library for numerical operations, used particularly in embedding computations [75]
- **pluralizer:** A Python library used to manage singular/plural word forms for more accurate natural language search. [76]
- **flask_cors:** Handles Cross-Origin Resource Sharing (CORS) to enable requests from the mobile frontend [77]
- **Firebase_auth & cloud_firestore (Flutter packages):** For integrating authentication and data storage in the mobile app [78] [79]
- **http (Flutter package):** For sending HTTP requests to the Flask API from the mobile app [80]

4.2.6 AI Model Files Used in Backend

- **config.json, pytorch_model.bin, sentence_bert_config.json:** Configuration and weights of the pre-trained Sentence-BERT model.
- **tokenizer.json, tokenize_config.json, vocab.txt:** Used for tokenization and encoding of user queries and product names. [81]

4.2.7 Hardware Requirements

- **Processor:** Intel Core i5-1135G7 (11th Gen), 2.40 GHz – a quad-core processor offering decent performance for application development, testing, and lightweight AI model inference.
- **RAM:** 8 GB– sufficient for mobile development with Flutter and web backend with Flask, though optimization is required when dealing with AI models or larger datasets.
- **Operating System:** Windows 11 Professionnel, version 23H2 – supports development tools like Visual Studio Code, Python, Flutter SDK, and Android Studio.
- **System Type:** 64-bit Operating System, x64-based processor – compatible with all modern development frameworks and libraries.
- **Graphics:** Integrated Intel Iris Xe Graphics (inferred from CPU model) – suitable for development and light model testing, but not recommended for training large AI models or using GPU-accelerated deep learning frameworks

4.3. Demonstration of interfaces

NB Delights is a smart catering platform that offers a mobile app for clients and a web-based system for providers, it leverages AI to making the whole catering process intelligent, smooth, and user-friendly.

4.3.1 Admin Dashboard Of NB-Delights

Figure presents the Admin Dashboard, is the central control panel for managing and monitoring operations within the NB Delights smart catering service platform. It is designed with an intuitive interface and key performance indicators (KPIs) to help administrators make informed decisions in real-time build with HTML, CSS, JAVA SCRIPT and firebase.

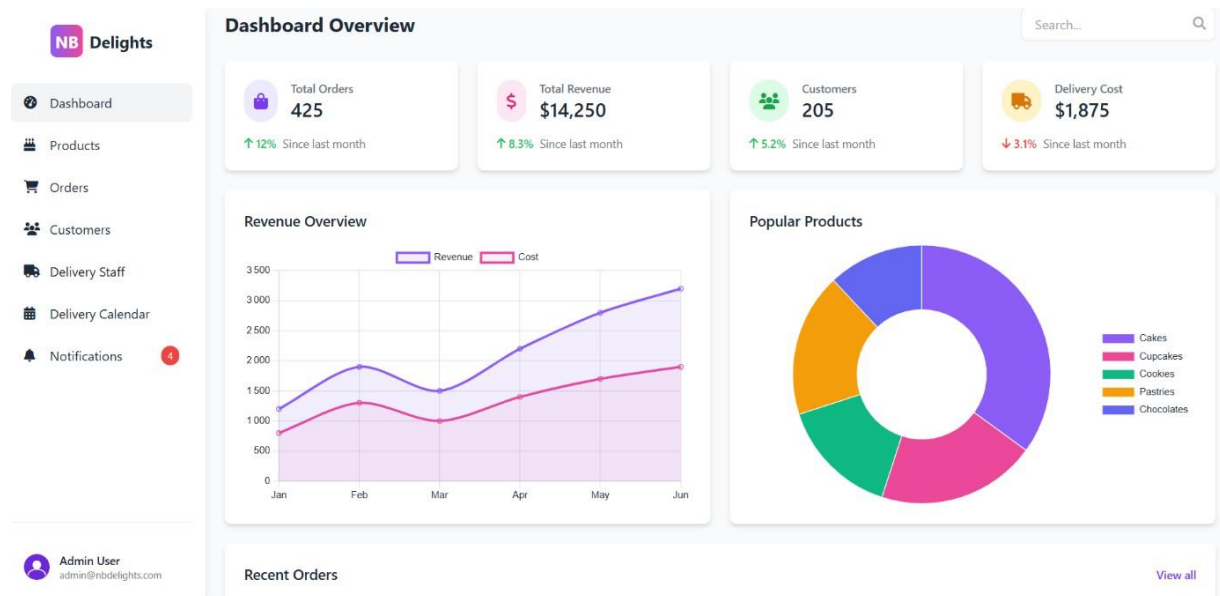


Figure 28: admin dashboard of NB-Delights

4.3.1.1 Dashboard Overview

Upon login, the admin is presented with a comprehensive overview summarizing platform activity:

- **Total Orders:**
Displays the number of customer orders received (e.g., 425 orders). A green upward arrow indicates a 12% increase compared to the previous month.
- **Total Revenue:**
Shows the revenue generated (e.g., \$14,250), along with an 8.3% growth trend.
- **Customers:**
Indicates the total registered users (e.g., 205), with a 5.2% growth trend.
- **Delivery Cost:**
Highlights logistics expenses (e.g., \$1,875), with a red downward trend showing a 3.1% reduction from last month.

4.3.1.2 Revenue Overview

This interactive line chart provides a month-by-month comparison of revenue versus cost:

- **Purple line:** Revenue
- **Pink line:** Operational Costs

The visual helps the admin track financial performance, profitability trends, and forecast future margins.

4.3.1.3 Popular Products Analysis

A donut chart offers insights into the most demanded products on the platform:

- **Cakes (most popular)**
- **Cupcakes**
- **Cookies**
- **Pastries**
- **Chocolates**

This data can be leveraged for personalized marketing, adjusting inventory, and forecasting demand.

4.3.1.4 Navigation Menu

The left-side panel provides quick access to various management modules:

4.3.1.4.1 Dashboard

is central control, home screen with KPIs (Figure 28)

4.3.1.4.2 Products

figures present Manage product listings:

- **Add**
- **Edit**
- **remove**

and manage details:

- **images**
- **categories**
- **tags**
- **prices**

Products									
Search products... + Add Product									
All Categories Pack Sweets Sales									
PRODUCT	TYPE/TH	CATEGORY	DESCRIPTION	QUANTITY	PRICE	COST	TAGS	ACTIONS	
salles prestige	cakes	sales	chicken with cheese	20	150 DA	75 DA	salles anniversaire prestige buffet		
dates stuffed	pastries	sweets	dates with almonds	0	100 DA	50 DA	prestige buffet dates chocolate omra		
gateaux	pastries	pack	gateaux prestige	20	100 DA	50 DA	anniversaire prestige buffet dates omra		
BAC cake	cakes	sweets	chocolate cake with banana , Nuts and chocolate mousse theme elsa barbie	0	5500 DA	2000 DA	CAKE theme rose bac soutienance BEM		
barbie elsa cake	cakes	sweets	chocolate cake with banana ,Nuts and chocolate mousse	2	5500 DA	3000 DA	cake anniversaire barbie elsa birthday kids baby bleu		
salles prestige	pastries	sales	with chicken and cheese	20	150 DA	80 DA	birthday rose prestige cheese chi		

Figure 29: product managing of NB-Delights

Add New Product

Product Name

Category

Select a category

Type

Select a type

Description

Price (\$)

Cost (\$)

Quantity

Tags

Enter tags separated by commas

Product Image



Upload a file or drag and drop

PNG, JPG, GIF up to 10MB

PNG, JPG, GIF up to 10MB

Cancel

Add Product

Figure 30: add new product

Update Product

Product Name

salles prestige

Category

Sales

Type

Cakes

Description

chicken with cheese

Price (\$)

Cost (\$)

Quantity

150

75

20

Tags

salles, anniversaire, prestige, buffet

Product Image



Upload a file or drag and drop

PNG, JPG, GIF up to 10MB

PNG, JPG, GIF up to 10MB

Cancel

Update Product

Figure 31: update product

4.3.1.4.3 Orders

Figure presents view and track customer orders in real-time

Orders

ORDER ID	CUSTOMER	PRODUCTS	ITEMS TOTAL	ORDER DATE	EVENT TYPE	EVENT DATE	DELIVERY	DELIVERY STAFF	STATUS	TOTAL	ACTIONS
#ord_1	mohamed 662545985	cake nouha (3) cake nouha (2) Note: extra chocolate	\$12700.00	20/05/2025	Birthday	20/05/2025	29/05/2025 20:45 kharouba hai salem 2	Delivery Fee: \$0.00	pending	\$0.00	

Showing 1 to 5 of 42 results

Previous 1 2 3 Next

Admin User
admin@nbdelights.com

Figure 32: orders of NB-Delights

And this figure present modify orders

Modify Order

Delivery Time

29/05/2025 19:45

Delivery Address

kharouba hai salem 2

cake nouha	▼	3	2100
cake nouha	▼	2	3200

Total Items Price: 12700 DA

[+ Add Product](#)

[Cancel](#) [Save Changes](#)

Figure 33: modify orders of NB-Delights

4.3.1.4.4 Customers

Figure presents access customer profiles and activity

CUSTOMER	CONTACT	ADDRESS	ORDERS	SPENT	STATUS	ACTIONS
 No Name Joined: 07/05/2025	662545985 ya@nbdelights.com	kharouba, mostaganem	5	3200 DA	Active	

Showing 1 to 5 of 205 results

Previous 1 2 3 Next

 Admin User
admin@nbdelights.com

Figure 34: customers of NB-Delights

4.3.1.4.5 Delivery Staff

Figure presents manage delivery personnel, with add staff, see delivery performance:

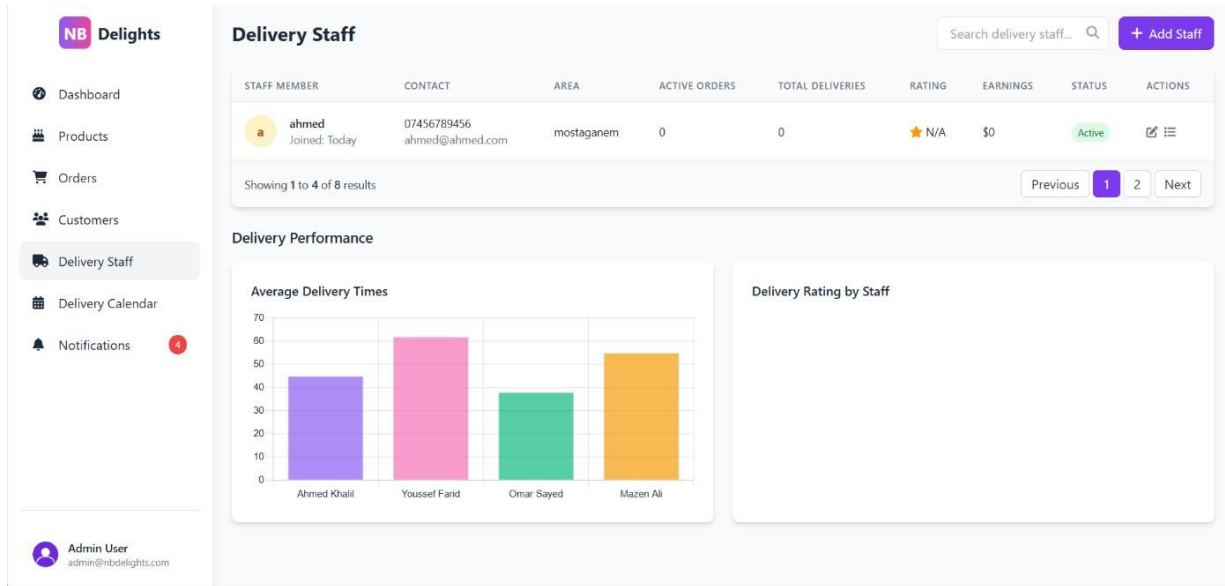


Figure 35: delivery staff of NB-Delights

4.3.1.4.6 Delivery Calendar

Figures present schedule and assign delivery slots by:

- Day
- Week
- Month

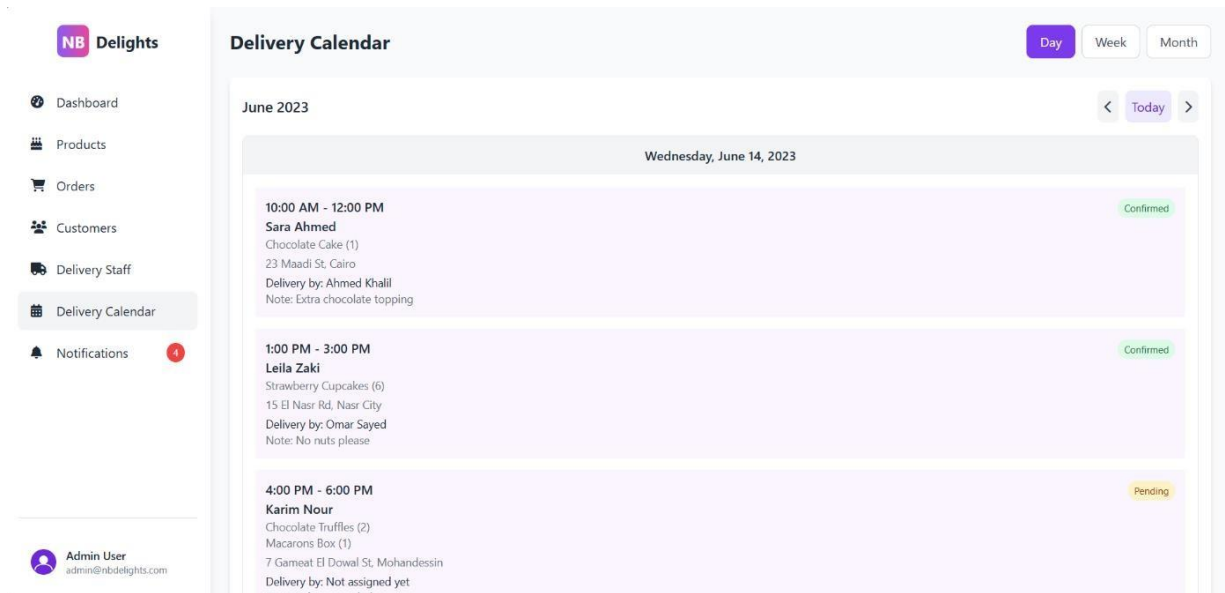


Figure 36: delivery calendar by day of NB-Delights

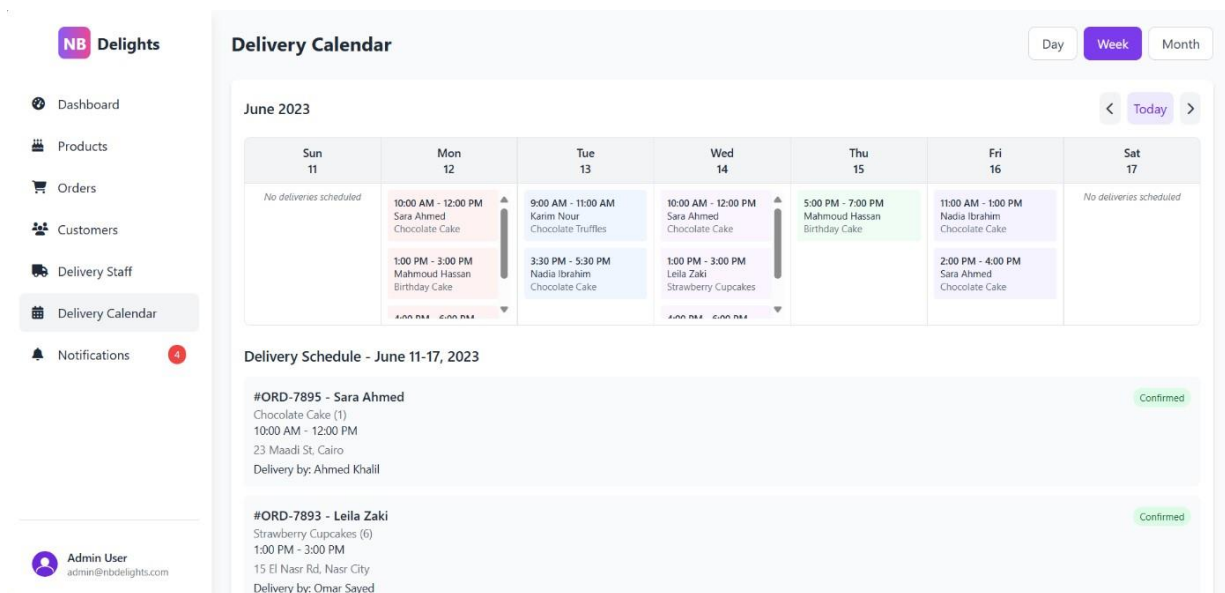


Figure 37: delivery calendar by week of NB-Delights

And Delivery calendar by month:

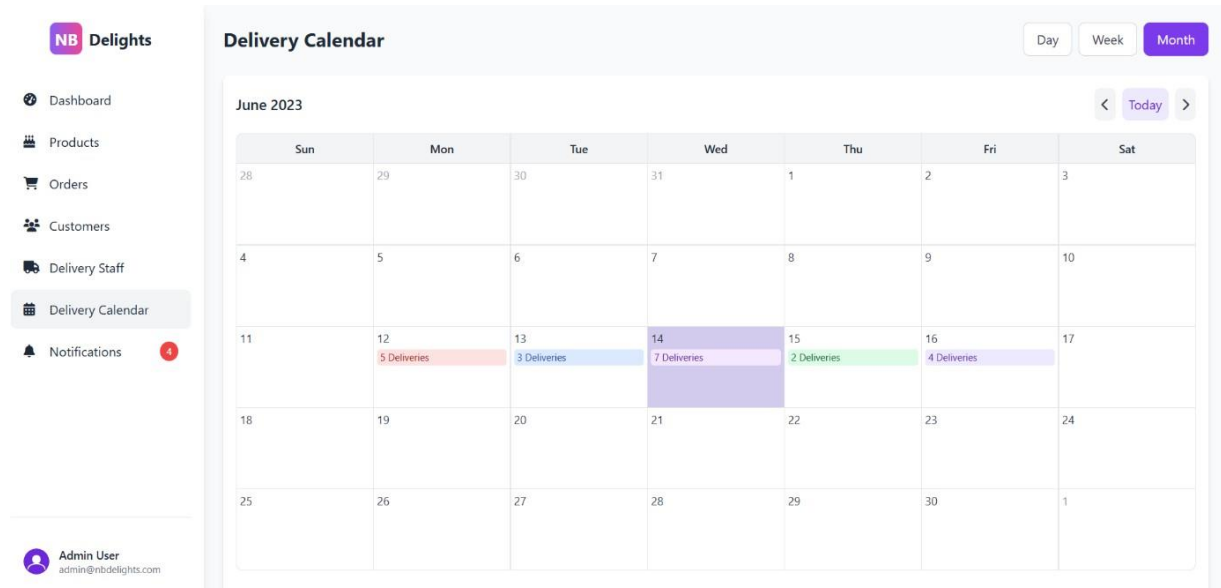


Figure 38: delivery calendar by month of NB-Delights

4.3.1.4.7 Notifications

Figure presents real-time system and user alerts (e.g., new orders, customer inquiries)

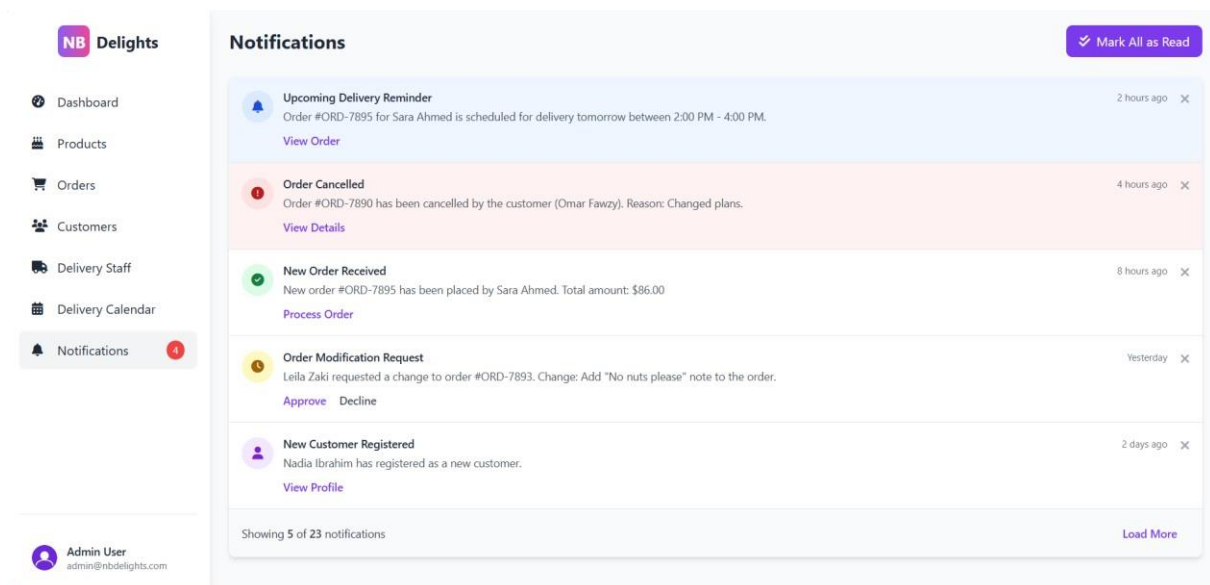


Figure 39: notifications of NB-Delights

4.3.1.5 Admin Profile

Located at the bottom left, the admin profile section displays the logged-in user details and email address (admin@nbdelights.com) for accountability and support access.

4.3.2 Client Mobile Application NB-Delights

NB Delights is an intelligent mobile application designed to revolutionize the personalized food catering experience for events such as birthdays, weddings, graduations, and more. Powered by cutting-edge AI technologies and built with Flutter and Firebase

4.3.2.1 Splash and Home screen

(figure 40) is the first screen users see when opening the app show NB-Delights logo and (figure 41) is second screen users see display:

- Search functionality connects to Flask + BERT + NLP (figure 42) (figure 43)
- Category filters (All, Pack, Sales, Sucre, Available)
- Product listings

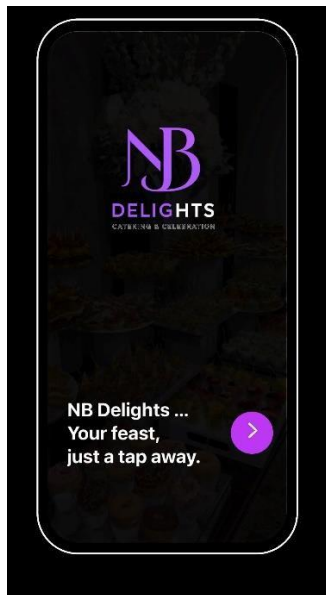


Figure 40: splash screen of NB-Delights

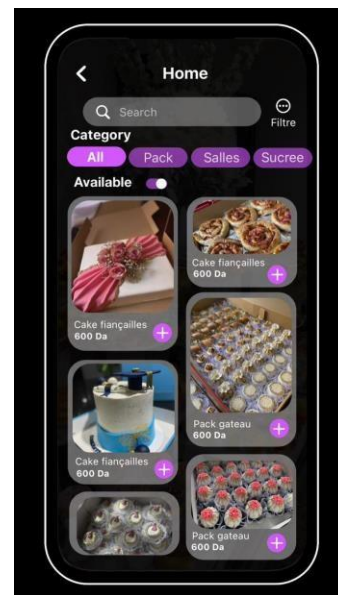


Figure 41: home screen of NB-Delights

Figures present the deference between simple search used local NLP method and smart search used Bert method and cosine similarity methods:

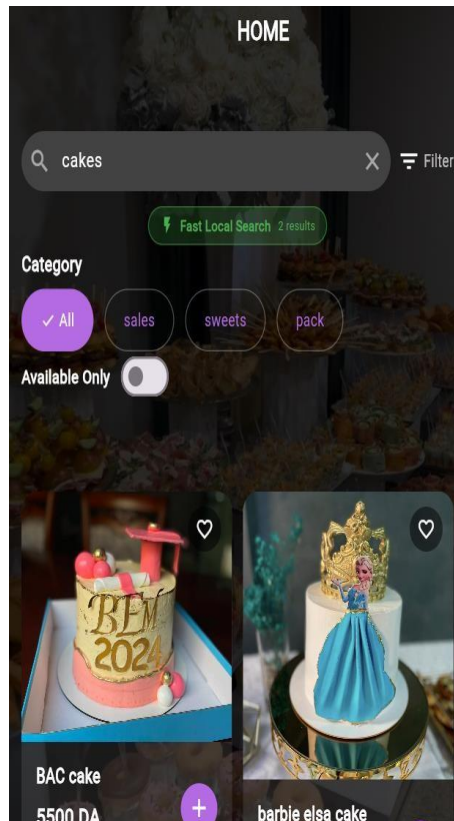


Figure 42: simple search

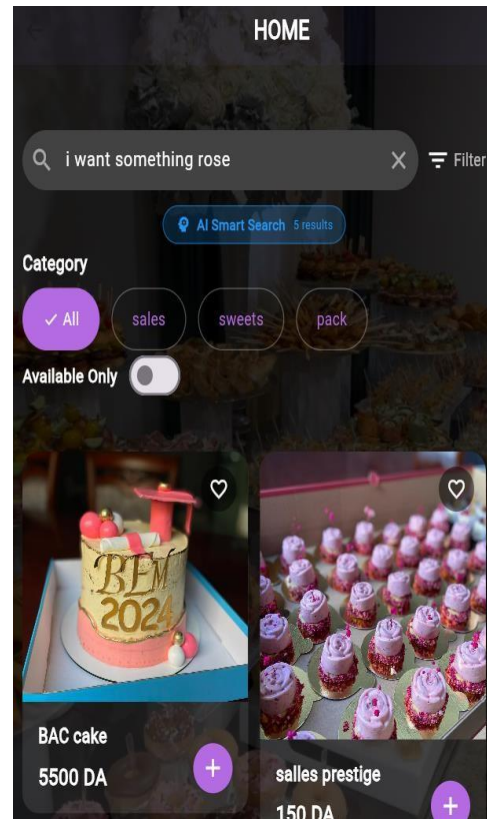


Figure 43: smart search

4.3.2.2 Filtering and Pack Generate Screens

(figure 44) users can filter products used collaborative filtering with cosine similarity method by:

- **Price range (minimum/maximum)**
- **Categories (all categories available)**
- **Event type**
- **Tags (e.g., chocolate, hazelnut, honey)**

(figure 45) Custom pack generation used association rule mining methods with:

- **Price range selection**
- **Category filtering**
- **Event type selection**
- **Number of persons**
- **Ingredient tags**

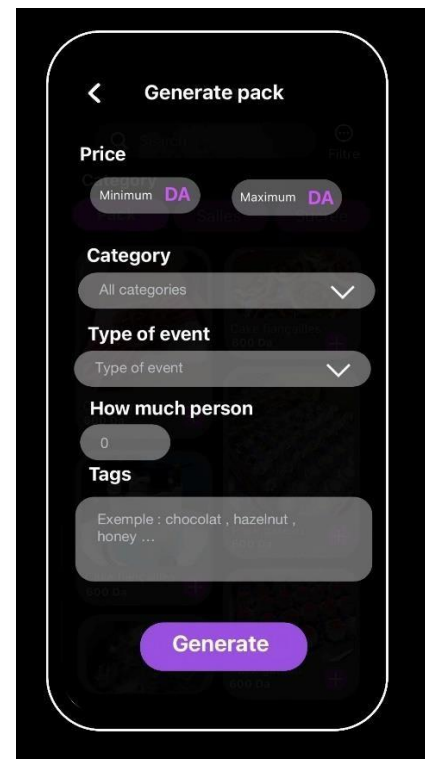
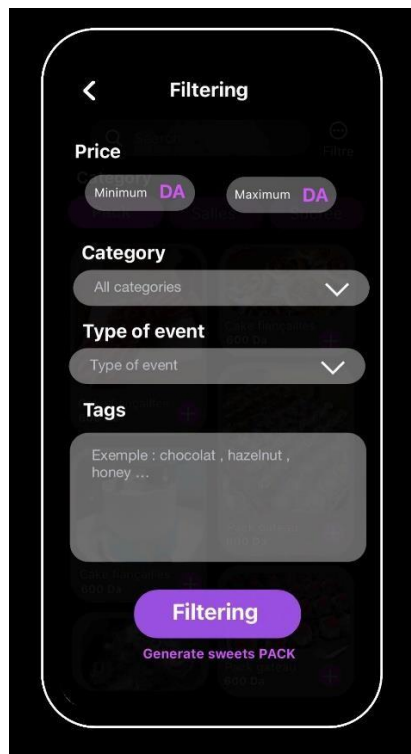


Figure 44: filtering screen of NB-Delights

Figure 45: pack generate screen of NB-Delights

After filtering the users will see display screen filtered with has preferences,
and after the user generate pack will see pack options (figure 41)

4.3.2.3 Product detail screen

Figure presents the details of product:

- **Hight-quality image**
- **Description and ingredients of product**

The user can't add to cart (panier) and order or add to favourite without login or sign up

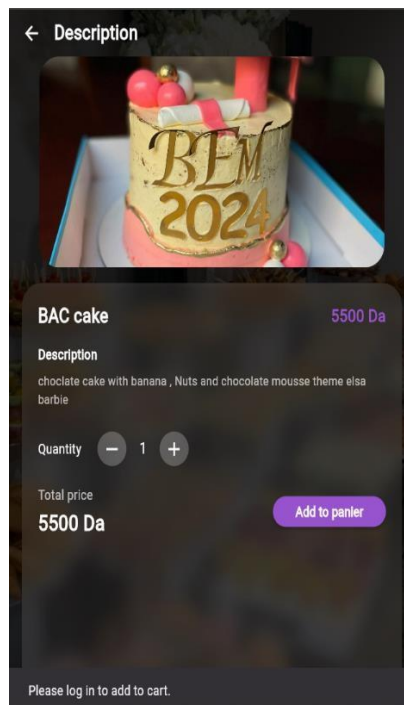


Figure 46: product details screen of NB-Delights

4.3.2.3 About NB-Delights Screen

Figure presents company information:

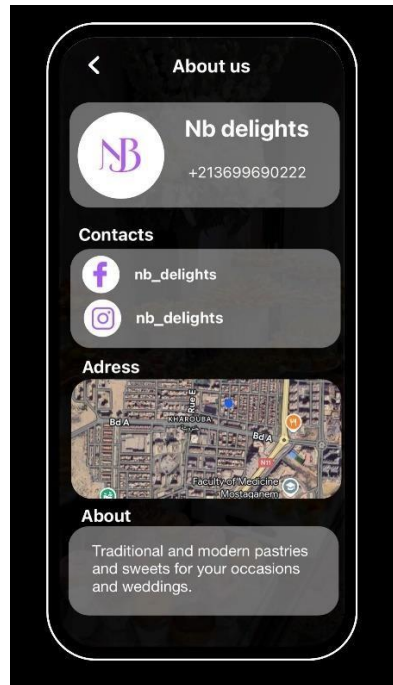


Figure 47: about NB-Delights screen

4.3.2.4 Authentication Flow Screens

With firebase authentication integration this figure present that the user should be login or signup to add orders

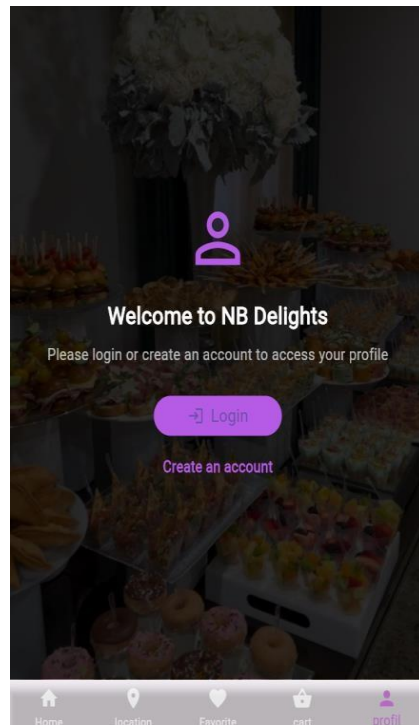


Figure 48: profile screen

4.3.2.4.1 Login and Signup Screens

Figures present creates account through sign up and login screens

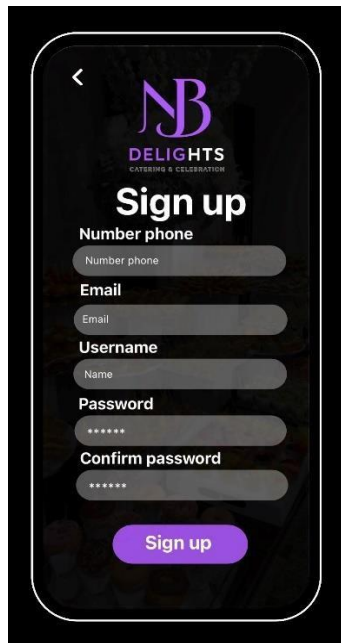


Figure 49: sign up screen of NB-Delights



Figure 50: login screen of NB-Delights

After sign up the user should be confirmed her phone with secret code (figure 51), and the user that already has account can update password (figure 52) or if password is forgotten, can recover through password recovery flow (figure 53):



Figure 51: confirmation
Phone number screen

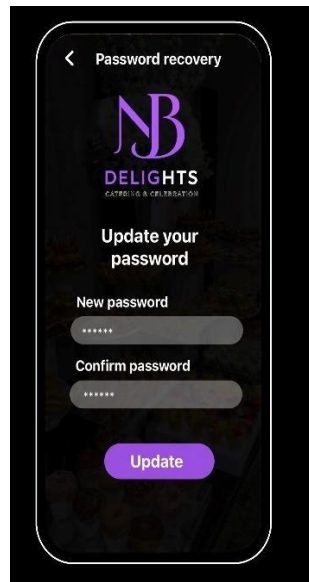


Figure 52: update
password screen

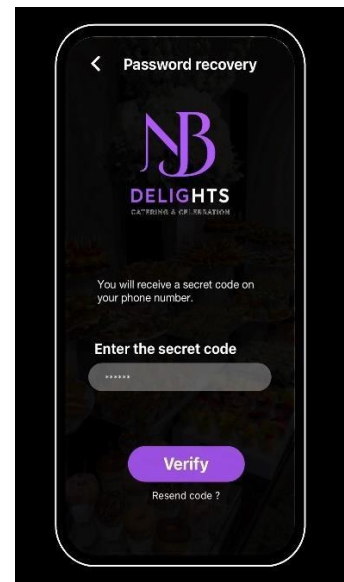


Figure 53: password
recovery screen

4.3.2.4.2 User Profile Screens

Figures present user profile display user information and can view order history and manage profile through profile screens



Figure 54: profile screen

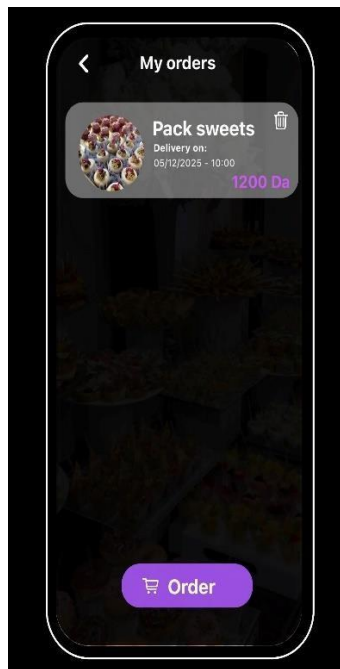


Figure 55: order history screen

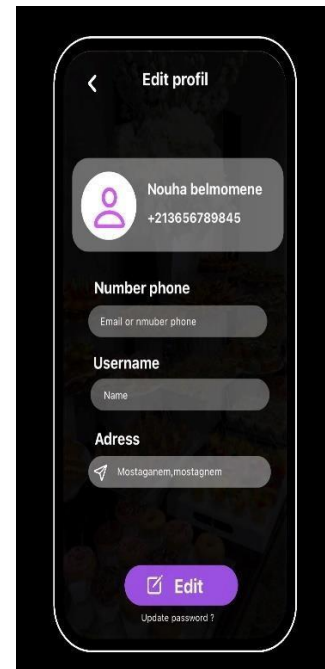


Figure 56: edit profile screen

4.3.2.5 Favourite Screen

Figure presents list view of liked products



Figure 57: favourite screen of NB-Delights

4.3.2.5 Cart screen

Figure presents Displays selected items with prices:

- Base products (e.g., Cake fiançailles - 600Da)
- Customizations (figure 59): Users can customize products with:
 - Special requests (e.g., more chocolate)
 - Color preferences (e.g., purple)
 - Personalized messages (e.g., "happy birthday Adem")
- Order summary with subtotal, delivery fee (400Da), and total
- Event type selection
- Event date and time selection
- Delivery address options (e.g., Mostaganem)
- Delivery date selection
- Final "Order" button

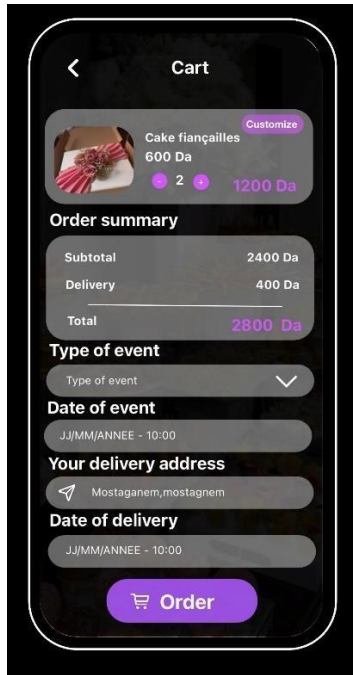


Figure 58: cart screen of NB-Delights

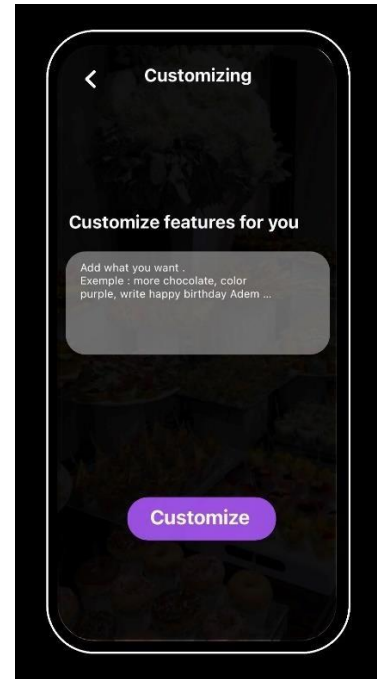


Figure 59: customizing screen of NB-Delights

4.4 Conclusion

This chapter covered the implementation of the NB Delights system, detailing both the client and provider sides, as well as the integration of AI and Firebase technologies. The use of modern frameworks like Flutter and Firebase enabled rapid development, scalability, and data security. Additionally, the integration of BERT for semantic understanding and cosine similarity for matching significantly enhanced the search functionality. This AI-powered approach allowed the system to deliver more accurate, personalized, and context-aware product recommendations, improving user satisfaction and overall system performance.

GENERAL CONCLUSION

In conclusion, AI represents a powerful tool for modernizing the catering sector, as demonstrated by the success of NB-Delights. The implemented methods have enabled the creation of an intelligent platform capable of adapting to individual needs and anticipating trends.

Among the AI methods used in the NB-Delights application, **Natural Language Processing (NLP)** with **BERT** plays a central role in understanding user queries and providing semantically relevant search results. **Sentiment Analysis**, powered by cosine similarity techniques, helps detect user preferences and emotional responses, improving recommendation accuracy. **Collaborative Filtering** allows the system to suggest products based on user behavior and similarities with others, while **Association Rule Mining** (such as Apriori or FP-Growth) uncovers frequent item patterns to optimize bundle offers. Each of these methods contributes to creating a smarter, more personalized experience.

The use of these AI methods provides significant advantages for NB-Delights. They enable **real-time personalization**, helping clients discover products that truly match their tastes. AI also improves the **efficiency of decision-making**, by automating suggestions and adapting dynamically to user behavior. Additionally, it enhances **customer satisfaction** through faster, more accurate recommendations, while allowing the business to identify emerging trends and respond proactively to customer needs.

However, challenges remain, particularly regarding computational resources and data utilization. Future prospects include algorithm optimization, integration of new technologies like GPT models, and expanded functionalities to enhance customer satisfaction. Thus, NB-Delights emerges as an innovative solution, paving the way for a new era of personalized and intelligent food services.

Bibliography

- [1] M. K. Nguyen and T. D. Huynh, "Personalized Recommendation Systems: An Overview and Future Challenges," Master's thesis, Nanyang Technological University, 2020.
- [2] "Understanding Recommendation Systems," Towards Data Science, 2021. [Online]. Available: <https://towardsdatascience.com/understanding-recommendation-systems-12345678> [Accessed: Oct. 20, 2021].
- [3] F. Ricci, L. Rokach, and B. Shapira, Recommender Systems Handbook, 2nd ed. Boston, MA, USA: Springer, 2015.
- [4] Y. Koren, R. Bell, and C. Volinsky, "Matrix Factorization Techniques for Recommender Systems," in Proceedings of the 2009 IEEE International Conference on Data Mining, 2009, pp. 424-433.
- [5] "Hype Cycle for Artificial Intelligence," Gartner, 2022. [Online]. Available: <https://www.gartner.com/en/documents/1234567> [Accessed: Oct. 20, 2022].
- [6] "Artificial Intelligence," Encyclopedia Britannica, 2023. [Online]. Available: <https://www.britannica.com/technology/artificial-intelligence> [Accessed: Oct. 20, 2023].
- [7] "The History of Artificial Intelligence," Encyclopedia Britannica. [Online]. Available:
- [8] "History of AI," AI Research Institute, 2020. [Online].
- [9] "AI Winter: The Period of Regression," Machine Learning Journal. [Online].
- [10] "Deep Blue vs. Garry Kasparov: A Historic Match," Chess History Journal. [Online].
- [11] IBM Cloud Education, "AI model lifecycle: How to develop, deploy and manage AI," IBM, 2020. [Online]. Available: <https://www.ibm.com/cloud/learn/ai-lifecycle>
- [12] M. Abadi et al., "TensorFlow: A system for large-scale machine learning," in Proc. 12th USENIX Symp. Operating Systems Design and Implementation (OSDI), Savannah, GA, USA, Nov. 2016, pp. 265–283.
- [13] I. Goodfellow, Y. Bengio, and A. Courville, Deep Learning, Cambridge, MA, USA: MIT Press, 2016.
- [14] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in Proc. 3rd Int. Conf. Learning Representations (ICLR), San Diego, CA, USA, 2015.

- [15] T. Fawcett, "An introduction to ROC analysis," *Pattern Recognit. Lett.*, vol. 27, no. 8, pp. 861–874, Jun. 2006.
- [16] D. Jurafsky and J. H. Martin, *Speech and Language Processing*, 3rd ed. Draft, 2023.
[Online]. Available: <https://web.stanford.edu/~jurafsky/slp3/>
- [17] "BERT: Pre-training of Deep Bidirectional Transformers," Devlin et al. (2018)
- [18] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient Estimation of Word Representations in Vector Space," *arXiv preprint arXiv:1301.3781*, 2013.
- [19] Sanjay Rathee, Manohar Kaul, and Arti Kashyap. 2015. R-Apriori: An Efficient Apriori based Algorithm on Spark. In *Proceedings of the 8th Workshop on Ph.D. Workshop in Information and Knowledge Management (PIKM '15)*. Association for Computing Machinery, New York, NY, USA, 27–34. DOI: <https://doi.org/10.1145/2809890.2809893>
- [20] "Fast Algorithms for Mining Association Rules," Agrawal et al. (1994).
- [21] B. Pang, L. Lee, and S. Vaithyanathan, "Thumbs up? Sentiment Classification using Machine Learning Techniques," in *Proc. EMNLP*, 2002, pp. 79–86.
- [22] A. L. Maas et al., "Learning Word Vectors for Sentiment Analysis," in *Proc. of ACL*, 2011, pp. 142–150.
- [23] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.
- [24] G. Adomavicius and A. Tuzhilin, "Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions," *IEEE Trans. on Knowl. and Data Eng.*, vol. 17, no. 6, pp. 734–749, Jun. 2005.
- [25] Y. Koren, R. Bell, and C. Volinsky, "Matrix factorization techniques for recommender systems," *Computer*, vol. 42, no. 8, pp. 30–37, Aug. 2009.
- [26] B. Sarwar et al., "Item-based collaborative filtering recommendation algorithms," in *Proc. 10th Int. Conf. on World Wide Web (WWW)*, 2001, pp. 285–295.
- [27] M. J. Pazzani and D. Billsus, "Content-based recommendation systems," in *The Adaptive Web*, Springer, 2007, pp. 325–341.
- [28] X. Su and T. M. Khoshgoftaar, "A survey of collaborative filtering techniques," *Advances in Artificial Intelligence*, vol. 2009, Article ID 421425, 2009.

- [29] J. MacQueen, “Some methods for classification and analysis of multivariate observations,” in Proc. of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, vol. 1, 1967, pp. 281–297.
- [30] M. Ester, H. Kriegel, J. Sander, and X. Xu, “A density-based algorithm for discovering clusters in large spatial databases with noise,” in Proc. of the 2nd Int. Conf. on Knowledge Discovery and Data Mining (KDD), 1996, pp. 226–231.
- [31] Y. Zhang, D. Yang, “A Survey on Deep Learning for Recommender Systems,” IEEE Trans. on Emerging Topics in Computational Intelligence, vol. 2, no. 3, pp. 228–239, Jun. 2018.
- [32] J. Liu et al., “DeepStyle: Learning user preferences for visual recommendation,” in Proc. 40th Int. ACM SIGIR Conf., 2017, pp. 841–844.
- [33] Q. Liu, S. Wu, and L. Wang, “Sequential prediction with recurrent neural networks for recommender systems,” in Proc. 2016 IEEE Int. Conf. on Data Mining (ICDM), pp. 1053–1058.
- [34] R. Zhao, X. Xia, and D. King, “Deep reinforcement learning for list-wise recommendations,” in Proc. of the 27th ACM Int. Conf. on Information and Knowledge Management (CIKM), 2018, pp. 1313–1322.
- [35] V. Mnih et al., “Human-level control through deep reinforcement learning,” Nature, vol. 518, pp. 529–533, 2015.
- [36] R. S. Sutton, D. McAllester, S. Singh, and Y. Mansour, “Policy gradient methods for reinforcement learning with function approximation,” in Proc. of NIPS, 2000, pp. 1057–1063.
- [37] S. Rendle, “Factorization Machines,” in Proc. of the 2010 IEEE Int. Conf. on Data Mining, pp. 995–1000.
- [38] Y. Koren, R. Bell, and C. Volinsky, “Matrix Factorization Techniques for Recommender Systems,” Computer, vol. 42, no. 8, pp. 30–37, Aug. 2009.
- [39] Y. Juan, Y. Zhuang, W. Chin, and C. Lin, “Field-aware factorization machines for CTR prediction,” in Proc. of the 10th ACM Recommender Systems Conf., 2016, pp. 43–50.

- [40] W. Ying et al., “Graph-based recommendation systems: A review,” in *IEEE Access*, vol. 8, pp. 216515–216531, 2020.
- [41] W. Hamilton, Z. Ying, and J. Leskovec, “Inductive representation learning on large graphs,” in *Proc. of Advances in Neural Information Processing Systems (NeurIPS)*, 2017, pp. 1024–1034.
- [42] F. Jiang et al., “Time-aware recommendation systems: A comprehensive survey,” in *IEEE Access*, vol. 7, pp. 142998–143020, 2019.
- [43] G. E. P. Box, G. M. Jenkins, G. C. Reinsel, and G. M. Ljung, *Time Series Analysis: Forecasting and Control*, 5th ed., Wiley, 2015.
- [44] Pang, B., & Lee, L. (2008). *Opinion Mining and Sentiment Analysis*. *Foundations and Trends in Information Retrieval*, 2(1–2), 1-135.
- [45] Mnih, V., Silver, D., Sutton, R. S., & Horgan, D. (2015). Human-level control through deep reinforcement learning. *Nature*, 518(7540), 529-533.
- [46] Schafer, J. B., Konstan, J. A., & Riedl, J. (2007). *Recommender Systems in E-Commerce*. In *Proceedings of the 1st ACM Conference on Electronic Commerce* (pp. 158-166).
- [47] Hyndman, R. J., & Athanasopoulos, G. (2018). *Forecasting: Principles and Practice*. OTexts. Retrieved from *Forecasting: Principles and Practice*
- [48] Gonzalez-Jimenez, H., Ayala, C., & Ruiz, R. (2020). The role of social media in food preferences and health recommendations. *Journal of Consumer Behaviour*, 19(5), 1014-1023.
- [49] Liu, X., Chen, P., & Zhang, X. (2021). Understanding customer loyalty in the context of personalization: The role of trust and satisfaction. *Journal of Retailing and Consumer Services*, 58, 102-215.
- [50] *Expert Systems with Applications* 237 (2024) 121541
- [51] [SPIE Conference Proceedings](#)
- [52] Xu, K., Zhou, H., Zheng, H., Zhu, M., & Xin, Q. (2024). Intelligent classification and personalized recommendation of e-commerce products based on machine learning. *arXiv preprint arXiv:2403.19345*.

- [53] Xu, W., Xiao, J., & Chen, J. (2024, October). Leveraging large language models to enhance personalized recommendations in e-commerce. In *2024 International Conference on Electrical, Communication and Computer Engineering (ICECCE)* (pp. 1-6). IEEE.
- [54] Wang, Z., Zhu, Y., Li, Z., Wang, Z., Qin, H., & Liu, X. (2024). Graph neural network recommendation system for football formation. *Applied Science and Biotechnology Journal for Advanced Research*, 3(3), 33-39.
- [55] Visual Paradigm, <https://visual-paradigm.com/usecase-diagram>
- [56] GeeksforGeeks, https://www.geeksforgeeks.org/sequence_diagram
- [57] GeeksforGeeks, <https://www.geeksforgeeks.org/use-case-diagrams>
- [58] Visual Studio Code, <https://code.visualstudio.com>
- [59] Microsoft PowerPoint, <https://www.microsoft.com/en-us/microsoft-365/powerpoint>
- [60] Microsoft Word, <https://www.microsoft.com/en-us/microsoft-365/word>
- [61] Adobe Illustrator, <https://www.adobe.com/products/illustrator.html>
- [62] Draw.io, <https://www.draw.io>
- [63] Dart Programming Language, <https://dart.dev>
- [64] Python Programming Language, <https://www.python.org>
- [65] MDN Web Docs – JavaScript, <https://developer.mozilla.org/en-US/docs/Web/JavaScript>
- [66] MDN Web Docs – HTML, <https://developer.mozilla.org/en-US/docs/Web/HTML>
- [67] Flutter, <https://flutter.dev>
- [68] Flask Framework, <https://flask.palletsprojects.com>
- [69] Firebase, <https://firebase.google.com>
- [70] Firebase Admin SDK Setup, <https://firebase.google.com/docs/admin/setup>
- [71] Hugging Face Transformers, <https://huggingface.co/docs/transformers/index>
- [72] PyTorch, <https://pytorch.org>
- [73] Scikit-learn, <https://scikit-learn.org>
- [74] Firebase Admin SDK for Python, <https://firebase.google.com/docs/reference/admin/python>
- [75] NumPy, <https://numpy.org>
- [76] Pluralizer (PyPI), <https://pypi.org/project/pluralizer>

- [77] Flask-CORS, <https://flask-cors.readthedocs.io/en/latest>
- [78] Firebase Auth (pub.dev), https://pub.dev/packages/firebase_auth
- [79] Cloud Firestore (pub.dev), https://pub.dev/packages/cloud_firestore
- [80] HTTP (pub.dev), <https://pub.dev/packages/http>
- [81] Sentence Transformers – Hugging Face, <https://huggingface.co/sentence-transformers>

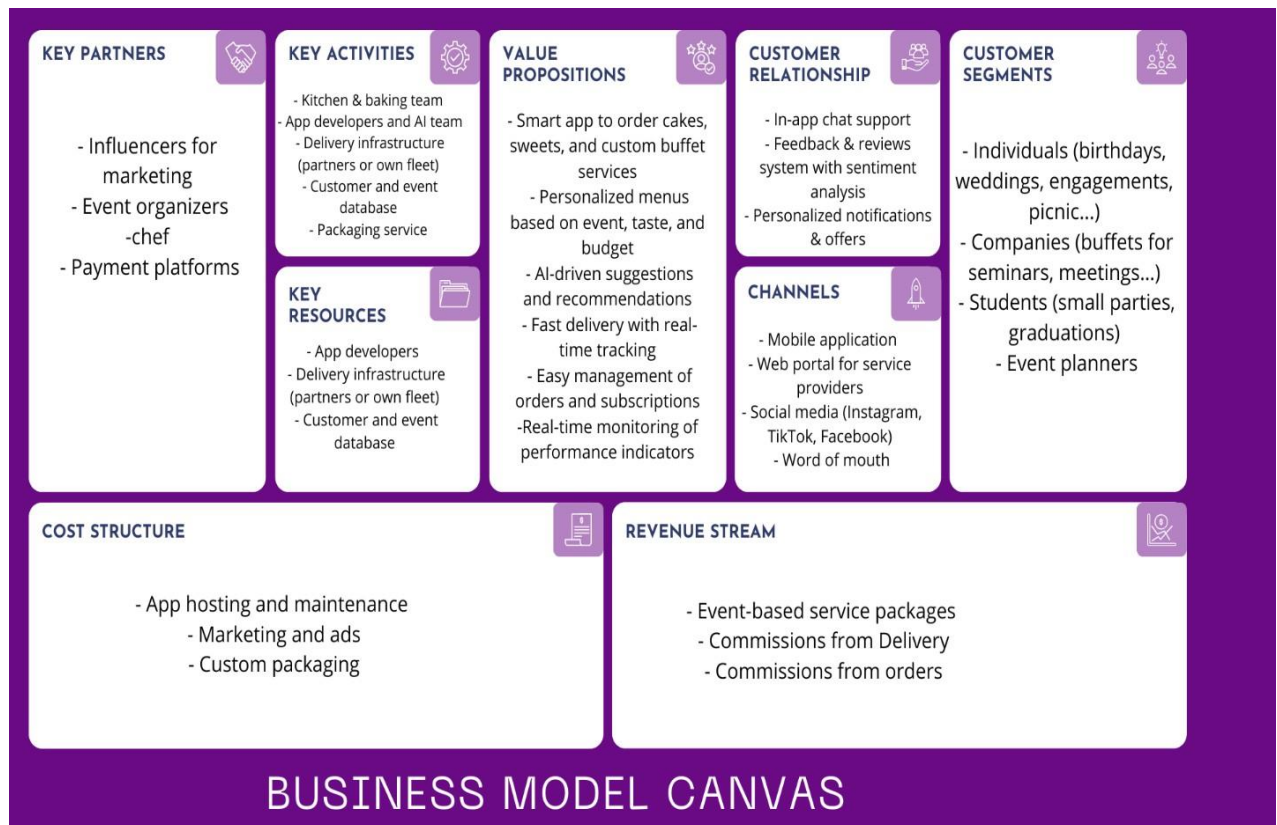
Annex BMC, Business Plan

Project technical sheet:

Title of your project	NB-Delights
phone number	+213 657252249
email address	NBdelights.ai@gmail.com
Your city or town of activity	Mostaganem, algeria

Nature of project:

Service



1. Business Description

● **Overview:** NB Delights is an intelligent mobile and web-based catering platform that specializes in preparing and delivering custom cakes, desserts, and buffet services for various events (e.g., birthdays, weddings, corporate meetings, and small parties). Powered by AI, the application enables customers to easily place orders, receive personalized recommendations, and track their deliveries in real-time. The project is based in Mostaganem and aims to scale across Algeria.

2. Customer Segments

Who are our most important customers? Who are we creating value for?

Phase 1: Local Market (Mostaganem)

- Individuals (birthdays, weddings, engagements, picnics)
- Students (graduations, small events)
- Local event organizers
- Small local businesses (buffets for seminars)

Phase 2: National Expansion

- Event planning agencies
- Companies and corporations (conferences, business meetings)
- Catering partners and food providers looking for smart logistics

3. Customer Relationships

How do we attract, retain, and support our customers?

- In-app chat support for real-time customer assistance
- Feedback & review system powered by AI sentiment analysis
- Personalized offers and notifications through the app
- Engaging social media campaigns with influencers
- Loyalty programs and referral bonuses to reward regular customers

4. Value Proposition

What value do we deliver to the customer? How do we solve their problems?

- Smart app for ordering cakes, sweets, and custom buffet packages
- AI-based personalized recommendations based on taste, event type, and budget
- Real-time delivery tracking and performance monitoring
- Easy subscription and order management
- Personalized menus tailored to customer preferences and events
- Time-saving and hassle-free event catering with quality assurance

5. Key Activities

What are the most critical operations for delivering our value?

- Kitchen operations and baking team management
- Mobile app and AI algorithm development
- Packaging and presentation of food items
- Real-time order tracking and logistics management
- Customer data and event preference management
- Digital marketing and social media engagement
- Coordination with delivery partners or internal fleet

6. Key Partners

Who are our essential collaborators and suppliers?

- Influencers and social media content creators
- Professional chefs and bakers
- Event organizers and venues
- Online payment gateways
- Packaging suppliers
- Delivery service providers or company-owned fleet

7. Key Partners

How do we reach and communicate with customers?

- Mobile application for Android and iOS
- Web portal for service providers and clients
- Social media platforms (Instagram, TikTok, Facebook)
- Word of mouth from happy clients
- Partnerships with event venues and local businesses
- Participation in local fairs and expos

8. Revenue Streams

How do we generate revenue?

- Sale of event-based food packages
- Commission from orders made through the platform
- Commission from delivery services
- Premium subscriptions for custom menus or priority delivery
- Extra charges for urgent or last-minute orders

9. Cost Structure

What are the main expenses involved in running the business?

- Mobile app development, hosting, and maintenance
- Digital marketing, influencer partnerships, and paid ads
- Custom packaging and branding materials
- Delivery costs (fuel, vehicle maintenance, driver salaries)
- Security (cameras, alarms for kitchen and warehouse)
- Staff wages and management costs

10. Key Resources

What resources are critical to our operations?

- App developers and AI engineers
- Kitchen equipment and ingredients
- Delivery fleet and tracking systems
- Product photography gear (camera, lighting)
- Security surveillance systems
- Hosting servers and databases
- Customer and event preference data

10.1 Material resources:

The main sections provide a provisional budget, including estimated revenues, fixed and variable operating costs, and anticipated profits. Additionally, a financing plan outlines funding sources—such as personal contributions, subsidies, and bank loans—clearly illustrating the financial structure required to launch and sustain the business. The document also contains a breakdown of key expenses (equipment, ingredients, salaries, delivery logistics, and app development), aligning with the startup’s goal of combining AI-based

recommendation systems with catering food service. These well-organized financial tables and forecasts are critical in demonstrating the project's feasibility, scalability, and readiness for incubation.

<u>TABLEAU DU BUDGET DE LANCEMENT</u>	
<u>DESIGNATION</u>	<u>COUT ESTIME</u>
DEVELOPPEMENT APPLICATION MOBILE OU WEB	1 200 000,00 DA
MATERIELS DE CUISINE	800 000,00 DA
VEHICULE DE LIVRAISON	800 000,00 DA
STOCKS INITIAL OU STOCK DE DEPART	300 000,00 DA
MARKETING DE LANCEMENT	200 000,00 DA
DIVERS DEPENCES (FRAIS CREATION ENTREPRISE	100 000,00 DA
FOND DE ROULEMENT	400 000,00 DA
<u>TOTAL</u>	3 800 000,00 DA
<u>CHARGE MENSUELLES</u>	
<u>DESIGNATION</u>	<u>COUT ESTIME</u>
SALAIRE	240 000,00 DA
COMMUNITY MANAGER	40 000,00 DA
INGREDIENTS ET FOURNITURE	150 000,00 DA
ENTRETIEN MAINTENANCE&CARBURANT	50 000,00 DA
HEBERGEMENT APPLICATION ET MAINTENANCE	30 000,00 DA
RESEAU SOCIAUX ET PUBLICITE	70 000,00 DA
EMBALLAGE	30 000,00 DA
LOYER CUISINE	50 000,00 DA
<u>TOTAL CHARGE MENSUELLE</u>	660 000,00 DA
<u>CHARGE ANNUEL</u>	7 920 000,00 DA

bilan			
actif	montant	passif	montant
immobilisations	2 800 000,00 DA	fond propre	400 000,00 DA
stocks	300 000,00 DA	dettes fournisseur	2 700 000,00 DA
tresorerie	1 530 000,00 DA	resultat net	1 530 000,00 DA
total actif	4 630 000,00 DA	total pasif	4 630 000,00 DA

الجمهورية الجزائرية الديمقراطية الشعبية

وزارة التعليم العالي والبحث العلمي

جامعة عبد الحميد بن باديس-مستغانم- كلية العلوم
الدقيقة والإعلام الآلي

تصريح شرفي خاص بالالتزام بقواعد النزاهة العلمية لإنجاز البحث

أنا الممضي أدناه، الطالب(ة): بلمومن نهى رقم التسجيل الجامعي: 37030129
الحامل لبطاقة التعريف الوطنية رقم: 110030970002760002
والصادرة بتاريخ: 2021/12/12

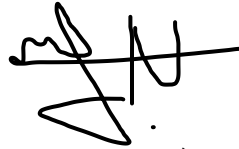
عن بلدية مستغانم
المسجل ب: كلية العلوم الدقيقة والإعلام الآلي / قسم: رياضيات و إعلام الآلي
شعبة: اعلام الآلي / التخصص: ذكاء اصطناعي و انترنت الاشياء
والمكلف بإنجاز مذكرة ماستر بعنوان:

Smart application for personal food catering service _ NB- Delights _

أصرح بشرفي أنني ألتزم بمراعاة المعايير العلمية والمنهجية ومعايير الأخلاقيات العلمية والنزاهة الأكاديمية المطلوبة في إنجاز البحث ، وأتحمل المسؤولية الشخصية عن كل المحتوى المتضمن في البحث المذكور أعلاه .

التاريخ: 2025/06/24

امضاء المعني :



* ملحق القرار الوزاري رقم 933 المؤرخ في 28 جويلية 2016 الذي يحدد القواعد المتعلقة بالوقاية من السرقة العلمية ومكافحتها.