

UNDERGRADUATE PROJECT REPORT

|  |  |
| --- | --- |
| **Project Title:** | An intelligence recommendation system for online food ordering platform based on Collaborative Filtering |
| **Surname:** | **Quan** |
| **First Name:** | **zhihao** |
| **Student Number:** | **202018010410** |
| **Supervisor Name:** | **Monty Mu** |
| **Module Code:** | **CHC 6096** |
| **Module Name:** | **Project** |
| **Date Submitted:** | **May 5, 2023** |

# **Declaration**

Here, students would sign a statement indicating that they adhered to appropriate academic conduct in carrying out their final project.

# **Acknowledgment**

Here, students are given the opportunity to thank those who have provided you with assistance and support.

# **Table of Contents**

[**Declaration** i](#_Toc164784318)

[**Acknowledgment** ii](#_Toc164784319)

[**Table of Contents** iii](#_Toc164784320)

[**Abstract** v](#_Toc164784321)

[**Abbreviations** vi](#_Toc164784322)

[**Glossary** vii](#_Toc164784323)

[**Chapter 1 Introduction** 1](#_Toc164784324)

[**1.1** **Background** 1](#_Toc164784325)

[**1.2** **Aim** 1](#_Toc164784326)

[**1.3** **Objectives** 1](#_Toc164784327)

[**1.4** **Project Overview** 2](#_Toc164784328)

[**1.4.1** **Scope** 2](#_Toc164784329)

[**1.4.2** **Audience** 3](#_Toc164784330)

[**Chapter 2 Background Review** 4](#_Toc164784331)

[**2.1** **Existing approach of Recommendation system:** 4](#_Toc164784332)

[**2.1.1** CB (Content Based Filtering) 5](#_Toc164784333)

[**2.1.2** **User CF (User-based collaborative filtering)** 6](#_Toc164784334)

[**2.1.3** **Limitation of Collaborative filter and Content-based recommendation:** 6](#_Toc164784335)

[**2.1.4** **Hybrid recommendation system:** 8](#_Toc164784336)

[**2.1.5** **Item-based popularity recommendation:** 9](#_Toc164784337)

[**2.2** **Analysis of existing food delivery platforms:** 9](#_Toc164784338)

[**Chapter 3 Methodology** 12](#_Toc164784339)

[**3.1** **Approach** 12](#_Toc164784340)

[**3.1.1** **Software development model:** 12](#_Toc164784341)

[**3.1.2** **Demand collection:** 13](#_Toc164784342)

[**3.1.3** **Overall system module architecture:** 13](#_Toc164784343)

[**3.1.4** **Use case:** 14](#_Toc164784344)

[**3.1.5** **Database design:** 16](#_Toc164784345)

[**3.1.6** **Recommendation system approach:** 16](#_Toc164784346)

[**3.2** **Technology** 21](#_Toc164784347)

[**3.3** **Project Version Management**. 21](#_Toc164784348)

[**Chapter 4 Implementation and Results** 22](#_Toc164784349)

[**Chapter 5 Professional Issues** 23](#_Toc164784350)

[**5.1** **Project Management** 23](#_Toc164784351)

[**5.1.1** **Activities** 23](#_Toc164784352)

[**5.1.2** **Schedule** 25](#_Toc164784353)

[**5.1.3** **Project Data Management** 25](#_Toc164784354)

[**5.1.4** **Project Deliverables** 26](#_Toc164784355)

[**5.2** **Risk Analysis** 27](#_Toc164784356)

[**5.3** **Professional Issues** 28](#_Toc164784357)

[1.1.2 Legal Issues 28](#_Toc164784358)

[1.1.3 Social Issues 29](#_Toc164784359)

[1.1.4 Ethical Issues 29](#_Toc164784360)

[1.1.5 Environmental Issues 29](#_Toc164784361)

[**Chapter 6 Conclusion** 30](#_Toc164784362)

[**References** 31](#_Toc164784363)

[**6.1** **Formatting Requirements** 33](#_Toc164784364)

[**6.2** **Written Presentation** 33](#_Toc164784365)

[**Appendices** 34](#_Toc164784366)

# **Abstract**

.  
Food delivery is becoming a new popular food trend in China, and the food delivery market has great potential for development in China. Therefore, this project aims to build an Intelligent online food delivery website with Python, Django and MySQL. For customers, it has features like registration, address management, favorites, shopping carts, and the ability to review and rate purchases.  For shop owners, the website offers features to manage their store and products, as well as order management features. In addition to this, the site uses Hybrid Recommender System (RS) to provide personalized recommendations to the users by using User-based Collaborative Filtering (User-CF) technology along with Content-Based Filtering (CB) technology to analyze the user's preferences as well as the similarity of the products.

This approach not only enhances the shopping experience but also boosts sales by intelligently matching products with customers' preferences. By integrating User-CF and CB, the website provides a tailored experience, whether you're a frequent shopper or new to the platform. The combination of these recommendation systems, along with the site's easy-to-use features and robust security measures, positions our food delivery website as a top choice for customers seeking a personalized shopping experience and for shop owners looking to connect with the right audience.

***Keywords: Recommendation system, Hybrid recommendation, Collaborative filtering, Content-based filtering, User-CF, Food delivery website***

# **Abbreviations**

CF: Collaborative Filtering

User-CF: User-based collaborative filtering

RS: Recommendation system

CB: Content-based filtering

# **Glossary**

**Recommendation System:**

Recommendation systems are information filtering systems providing a personalized item recommendation to a user in a service environment that can hold or collect various data. [1]

**Hybrid recommendation:**

A hybrid recommendation system is a method that combines multiple recommendation techniques to address the shortcomings of individual models. By integrating various methods, it can effectively generate recommendations that align more closely with users' preferences, thereby overcoming the limitations inherent in single-method systems [2].

**Content-based filtering:**

The content-based filtering approach makes recommendations by analyzing the characteristics of products that are of potential interest to the user. The advantage of this method is that it can show users new products [3].

**Collaborative Filtering:**

User-based Collaborative Filtering algorithms will recommend items of interest to the user based on the calculation of user or item similarity [4]. There are two common collaborative filtering methods: user collaborative filtering (User-CF) and Item-based collaborative filtering (Item-CF) [5].

**User-based Collaborative Filtering:**

Is an algorithm in collaborative filtering algorithm, which recommends items or product that similar users like for users by finding similar users for them.

**Cosine Similarity:**

Cosine similarity is a commonly used measure in computer science to determine the similarity between two vectors, often representing user or item preferences in a recommendation system. It calculates the cosine of the angle between two vectors, providing a similarity score typically ranging from 0 to 1. It helps determine the relationship between two related items, making it a common method for assessing the similarity of users or items based on their features or behaviors [6][7].

**Jaccard Index:** The Jaccard index, also known as the Jaccard similarity coefficient, is a statistical measure used to compare the similarity and diversity of sample sets, especially in recommender systems. The index assesses the similarity between sets by calculating the ratio of intersections to unions of sets, focusing only on the presence of items rather than the frequency of items in the set. In recommender systems, this index is used to compare users or items based on their ratings, thus helping to identify similar users or items with common appeal [8].

**Cold start:**

The Cold Start refers to the difficulty of generating accurate recommendations when a recommendation system is just getting started, due to a lack of sufficient user behavioral data and item data.

**Django:**

Django is a Python web framework that provides a set of methods to help developers to develop and maintain websites [9].

# **Introduction**

## **Background**

In the era of highly developed information technology and logistics industry, online take-out has become a new trend sweeping the world, especially in China, ordering take-out has become a new popular way of eating in China [10][11].

However, the increasing health consciousness of customers and their growing demand for healthy food options has become a major trend in the online food delivery market. As a result, online food delivery platforms need to respond by offering a wider range of healthy and sustainable food choices. In addition to this, online food selling platforms have been suffering from reduced information utilization and "information overload", where too many product categories can make it difficult for users to find their favorite information. Over time, users become less interested in the platform [10][12].

In order to solve the problem of information overload, the use of recommender systems has become an effective technical choice [10]. Therefore, in this project a hybrid recommendation system is implemented in which User-CF algorithm is integrated with CB algorithm to provide personalized recommendation of products to the customers through an intelligent recommender system so as to reduce the information overload problem faced by the users. Where CF is an algorithm that provides personalized recommendation of products by basing on the similarity of users, while CB provides recommendations to users by analyzing the content features of products. Integrating these two recommendation methods together is expected to provide users with more accurate and personalized recommendation services [2].

## **Aim**

The main goal of this project is to integrate a User-CF algorithm and CB into a hybrid recommendation system, so as to improve the user experience of the ordering platform by providing highly personalized and targeted food recommendations.

## **Objectives**

The object are as follows:

1. Research on the Existing Food Delivery Platform
2. Comparison between food delivery platforms
3. Research on recommendation system
4. Research on CF and Content-based filtering
5. Function requirements analysis
6. System design (divided into several modules)
7. Web Implementation (Implement the front-end and back-end)
8. Recommend function design
9. Develop recommend function
10. Test and evaluate website function
11. System testing & performance analysis

## **Project Overview**

### **Scope**

This project aims to develop an intelligent recommendation system for an online food ordering platform, designed to enhance user experience by offering personalized food and restaurant suggestions. The platform will operate through a web-based application developed using the Django framework, integrating a MySQL database for data storage, and employing Pandas, NumPy, and scikit-learn for data analysis and algorithm implementation. The core functionality will cater to two primary user types: customers and merchants.

For customers, the platform will offer features such as account registration, address management, favorites, shopping cart management, order management and viewing order status, payment features with the option to cancel payments, as well as the ability to post comments and rate stores and products after purchase. Additionally, users will be able to manage orders and search for products or stores. The recommendation system will employ User-Based Collaborative Filtering (User-CF) to filter for logged-in customers with a substantial order history, providing tailor-made suggestions by analyzing preferences of similar users. For new users or those lacking sufficient user history, the platform will implement a Content-based filtering approach, focusing on product attributes like average ratings, the number of times a product has been favorited, and sales volume to generate personalized recommendations.

On the merchant side, the platform will allow for account and store registration, product management, order reception, and shipping management, along with access to store performance data.

The integration of this hybrid recommender system with the online ordering platform will significantly improve users' navigation and product discovery, ensuring a convenient, efficient and personalized ordering experience for non-users. By analyzing and leveraging user data and preferences, the platform aims to increase customer satisfaction and loyalty, ultimately leading to higher exposure and sales for merchants' stores.

### **Audience**

The primary audience of our platform includes the following:

1. Customers Seeking Personalized Food Recommendation: For users who need to seek personalized recommended food ordering experience, the personalized recommendation system of the website can be more efficient and more accurate to recommend food that may be of interest to them.
2. Restaurant Owners and Merchants: Our platform provides a platform for merchants to connect with customers, showcase their products, and greatly improve the opportunity for merchants to improve sales by leveraging the power of user data and preferences to personalize recommended products for each customer.
3. Busy office workers: For those with limited time and energy, our platform is a more convenient and intelligent solution. The site is able to provide users with fast, personalized meal recommendations based on their data and historical behavior.

# **Background Review**

## **Existing approach of Recommendation system:**

In this section, we will discuss and review the existing approach to recommendation systems.

Faced with the problem of information overload caused by the rapid development of the Internet. Therefore, recommendation systems are designed to solve this problem [13]. Recommendation systems are information filtering systems providing a personalized item recommendation to a user in a service environment that can hold or collect various data [1]. In fact, Recommendation systems are widely used in various fields, such as e-commerce and food delivery recommendation, etc., which not only brings benefits to the merchant, but also brings convenience to the users [10].

The current mainstream approaches to recommendation systems are illustrated in Figure 1. This figure outlines the hierarchy of mainstream sub-models of current recommendation system models into three main approaches: content-based filtering (using item features for recommendation), collaborative filtering and hybrid systems. Collaborative filtering methods are categorized into model-based and memory-based approaches based on the way they process the rating data of users and items [1]. Furthermore, memory-based collaborative filtering can be subdivided into two categories: user-based collaborative filtering and item-based collaborative filtering [14].

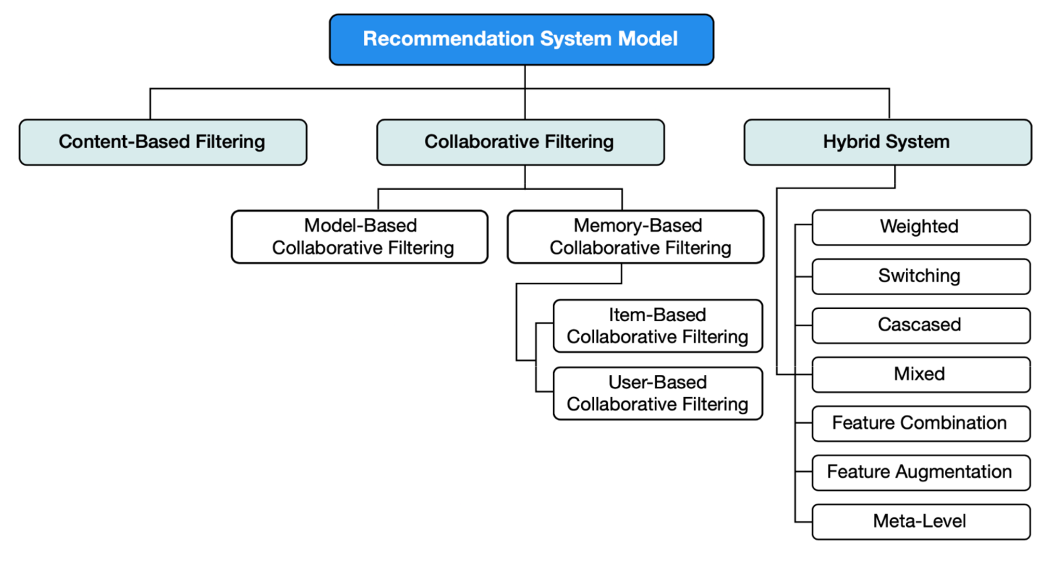


Figure 1: Overview of recommendation models [1].

### **CB (Content-based filtering)**

CB is an algorithm that recommend items to users based on the features of the items and the user's preferences. For example, recommending products to users based on their information about the product itself of interest [3]. For the CB recommendation algorithm, it is characterized by its ability to accurately extract the user's interests and does not require data from other users [14].

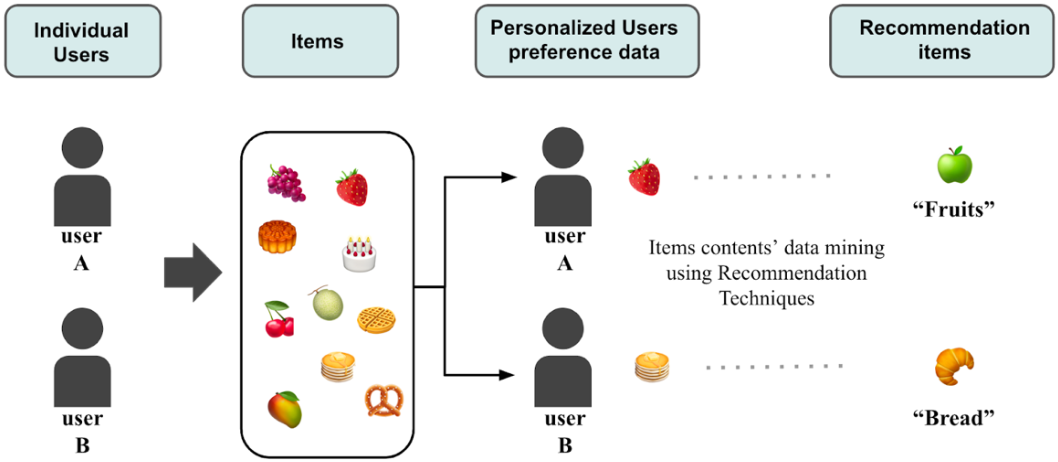
Figure 2 below outlines the principles of content-based filtering models in recommendation systems. As an example, it shows that for different users, in a collection of food items, these items are associated with personalized user preference data, indicating that user A likes strawberries and user B likes pancakes. By using recommendation techniques to analyze the content of the items, recommendations can be obtained for specific categories, such as "fruit" for User A and "bread" for User B.

Figure 2: Recommendation principle of Contents-Based Filtering Model [1].

### **User CF (User-based collaborative filtering)**

User-CF is based on calculating the similarity between users and then recommending products that are liked by other users similar to the target user [3]. For calculating similarity, Pearson correlation or cosine similarity is usually used to determine the similarity between users. In online delivery website can recommend similar good restaurants based on similar tastes between similar users [10][5].

User-CF is characterized by high recommendation accuracy without extracting the item's features and with a complete and rich dataset. In addition, User-CF can extract some potential correlations between recommended items and user preferences [14].

### **Limitation of Collaborative filter and Content-based filtering:**

* Limitation of User-CF:

Firstly, Li and Wang [14] discuss that as the number of users in the system increases, User-CF is confronted with the challenge of maintaining a large user similarity matrix. This not only increases the time required for recommendation computation, but also reduces the system's overall efficiency. Furthermore, User-CF does not involve the mining of item content, and therefore cannot address the cold-start problem for new items. In other words, the absence of necessary user behavior data prevents new items from being recommended to target users.

Second, Cacheda et al. [4]note that although User-CF is relatively simple to implement and provides fairly accurate results, it suffers from serious scalability problems. In order to make recommendations, the algorithm must process all the stored data, which is not feasible for large-scale online systems that require real-time recommendations.

Furthermore, Huang, Chen, and Zeng [15] highlight that since users typically rate only a few items and the majority of item data remains unrated, the challenge of recommending product diversity in this context is significant. Additionally, there is a paucity of behavioral history for new users to analyze, which further complicates the task of providing recommendations. Consequently, User-CF also encounters limitations in addressing common issues in recommender systems, such as the sparsity of the rating matrix and the cold-start problem.

* Limitation of CB:

Li and Wang [14] points out that CB may ignore objects that the user has not been exposed to before due to over-inclusion during the recommendation process. This situation can lead to monotony in the recommended content, depriving the user of the opportunity to explore new types of information, and thus a lack of novelty.

In addition, Cacheda et al. [4] explain that CB requires ITEMS that can be parsed by machine, which poses difficulties for the processing of multimedia information, because machine perception of content (e.g., color, texture, etc.) is different from human perception. Although manually labeled multimedia content solves this problem to some extent, content-based filtering still struggles to handle the vast amount of information available today. In addition, CB also can’t discover items that appear irrelevant but are actually interesting to the user's profile, it lacks the ability to make unexpected discoveries.

Finally, Salter and Antonopoulos [16] CB model only recommends data that is closely related to items that the user has previously evaluated, thus failing to provide access to diverse content. This suggests a clear limitation in the system's ability to recommend new items, thus reducing the user's exposure to new content.

* Conclusion of limitation:

Obviously, both User-CF and CB Recommendation have limitations, especially under conditions of data sparsity, which can lead to the cold start problem - difficulty in making accurate recommendations for new users or items without sufficient data [4]. This problem is particularly acute in the context of food delivery, where new restaurants or menu items are frequently added. These limitations can negatively impact the user experience by resulting in less accurate or less personalized recommendations.

### **Hybrid recommendation system:**

To address these challenges, hybrid recommender systems are introduced. A hybrid recommender system is a method that combines multiple recommendation techniques, which can combine the advantages of multiple models and mitigate the disadvantages of a single model. By integrating these approaches with additional data on item popularity and sales, our system can provide more robust and accurate recommendations even in the face of the typical cold-start problem associated with new items or users [10].

Bondevik et al. [17] describe a hybrid system is designed to leverage multiple sources of information, thereby improving the accuracy and personalization of recommendations provided to users. It represents a significant step forward in addressing the inherent limitations of traditional recommendation models, thereby significantly improving user satisfaction and engagement on the platform.

Song et al. [18] states that a hybrid recommendation method that combines a weighted combination of User-CF and Content-based recommendation methods based on existing models is able to capture user evaluations of food more accurately than a single traditional recommendation technique. It was found that the hybrid recommendation method performs more accurately in recognizing user evaluations and takes into account the similarities between users and food products compared to traditional recommendation methods.

Cacheda et al. [4] propose a solution to the shortcomings of cold start is proposed. This solution involves integrating multiple recommendation methods into a single recommendation system. The system utilizes the advantages of both user-CF and CB algorithms, and it also combines popularity and sales data to efficiently manage new items and users. This integration alleviates the cold start problem.

Pu and Hu [19] selected a hybrid recommendation algorithm for recommending movies in their study. The experimental results show that the hybrid recommendation algorithm achieves an accuracy of 81%, which is better than the traditional CB, Item-CF and User-CF algorithms.

Praditya, Permanasari, and Hidayah [20] propose that the limitations of existing models can be overcome by implementing a hybrid approach. In that article, three distinct models: User-CF, CB, and K-Nearest Neighbor Algorithm are combined to develop a method for recommending travel itineraries to users.

Shao and Xie [21] designed a hybrid algorithm utilizing user attributes and BP networks to solve the new user problem at cold start. The users are clustered according to their interests, and then the clustering results and the demographic information of the users are used to construct a decision tree, which associates the new users with the existing users, so as to realize the recommendation of new users. Experimental results show that the algorithm can improve the recommendation performance.

### **Item-based popularity recommendation:**

In addition, based on the discussion of the cold start problem above. In order to solve the cold start problem for new users, this project develops a recommendation function based on the popularity of the item, which is based on the extraction of the metadata of the item itself, and sorts the items according to their popularity (e.g., sales volume, number of favorites, ratings, etc.) so as to recommend them to new users. This avoids the situation where there is no product to recommend if the new user has no user data to extract.

## **Analysis of existing food delivery platforms:**

Online food ordering platforms have become increasingly popular in recent years, revolutionizing the way customers order food and receive delivery. Several companies, including Meituan, Ele. me and Uber eats, are commercial giants in the food delivery platform space, with Meituan and Ele.me being the most used delivery software in China [22]. These platforms have transformed the food industry, providing consumers with an easy and convenient way to order food online.

Therefore, in order to analyze and investigate the characteristics of Online food ordering platforms, Table 1 below shows functional comparisons of the different platforms. Through comparison, it is found that the basic functions of search, recommendation, purchase, shopping cart and order management are well implemented on the four platform.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Feature Comparison | MeiTuan food delivery [23] | Ele.ME [24] | Uber eats [25] | Just eat [26] |
| Customer register required Information | **Phone number** | **Phone number** | **Email &**  **Phone number** | **Email** |
| Become a merchant | **YES** | **YES** | **YES** | **YES** |
| Product Category function | **YES** | **YES** | **YES** | **YES** |
| Shopping Cart function | **YES** | **YES** | **YES** | **YES** |
| Search shop function | **YES** | **YES** | **YSE** | **YES** |
| Search product function | **YES** | **YES** | **NO** | **NO** |
| Collection shop function and management | **YES** | **YES** | **YES** | **NO** |
| Collection product function and management | **NO** | **NO** | **NO** | **NO** |
| Comment function | **YES** | **YES** | **NO** | **NO** |
| Delivery Method | **Delivery** | **Delivery** | **Delivery & Pickup** | **Delivery & Collection** |
| Recommended  Restaurants | **YES** | **YES** | **YES** | **YES** |
| Individual Recommended products | **NO** | **NO** | **NO** | **NO** |
| Address Management | **YES** | **YES** | **YES** | **YES** |
| Order status view and management | **YES** | **YES** | **YES** | **YES** |
| Shop rating function | **YES** | **YES** | **YES** | **YES** |
| Product rating function | **NO** | **NO** | **NO** | **NO** |

Table 1: Feature comparison between different platforms.

# **Methodology**

## **Approach**

The approach for a software development project should focus on the description of the software development methodology being used for the project.

### **Software development model:**

In the choice of software development model, the waterfall model is a common and classic development model, which shows the software development process in linear order. As shown in Figure 1 below, the software development process can be divided into seven parts in linear order: Requirement Analysis, System design, Model selection, Implement the system, system integration, test, system operation and maintenance.

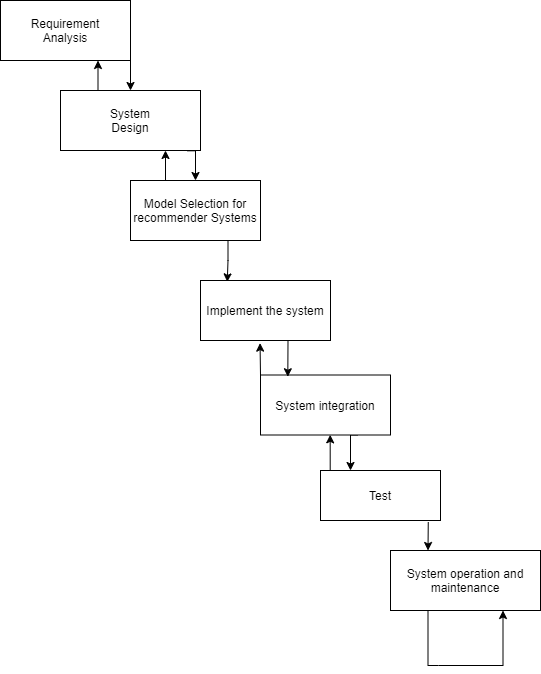


Figure 3: Waterfall model.

### **Demand collection:**

* Market research: Research on data analyze websites like statista to study the relevant market and competitors to understand the needs and preferences of users across the market. Relevant information can be obtained by investigating industry reports, data from market research agencies, and product analysis from competitors.
* User observation: Observe the behaviors and interactions of friends and classmates around when using the online takeout platform, so as to obtain the needs and problems of users in actual usage scenarios.
* Analysis for competitive food delivery website: Observe the popular online ordering software (Meituan, Ele.me, uber eats etc.) recorded to analyze which requirements are the most needed by users.

### **Overall system module architecture:**

This section describes the overall module of a food delivery website. It introduces and shows what modules make up the website, and in Figure 4 shows in detail what are the modules that make up the website.

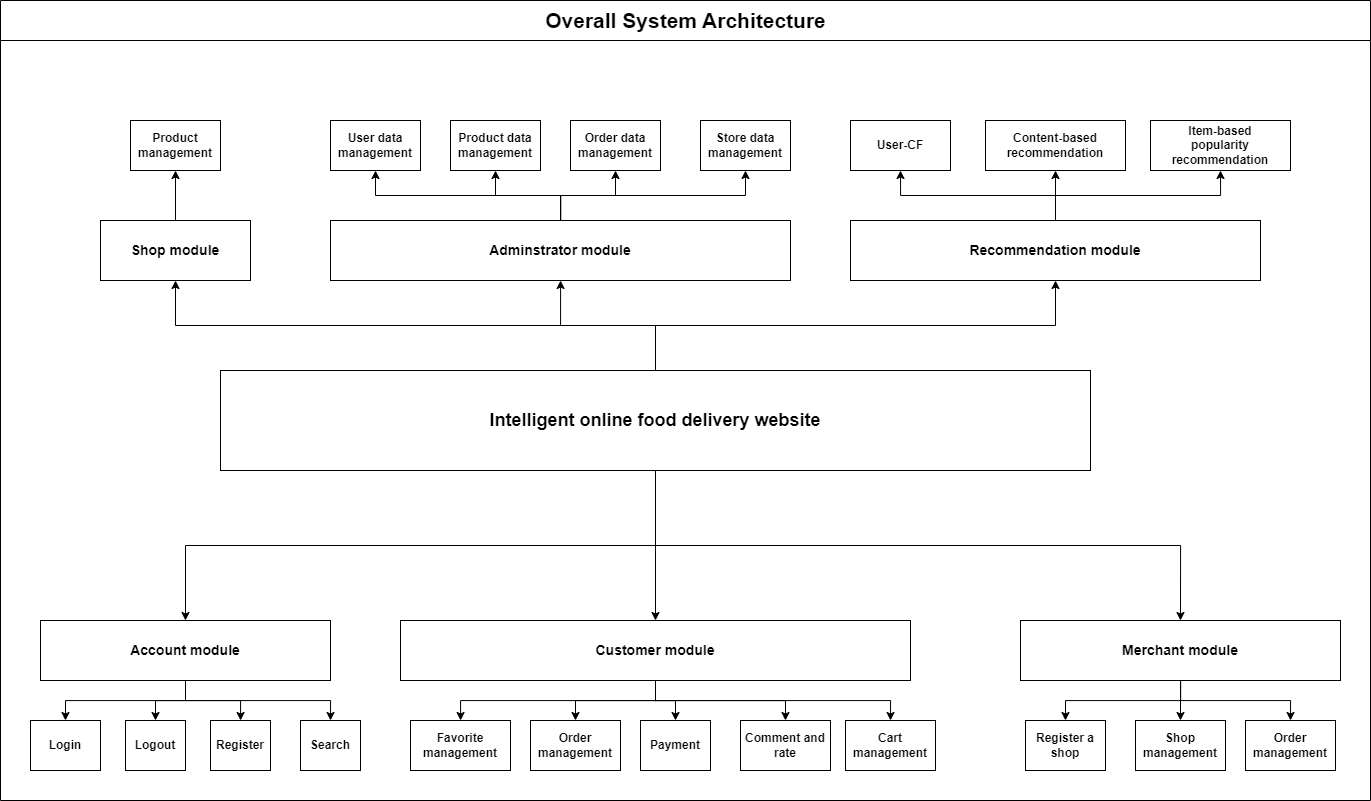


Figure 4: Overall system module architecture.

### **Use case:**

This section presents a User case diagram of the three user types from the different user perspectives supported by the website.

The User case diagram visualizes the functional requirements of the system, understands what the system is supposed to do, and how users can interact with the system.

* Customer Use case:

The interaction between a customer and this online food delivery website is shown in Figure5. It includes basic account registration, login and restaurant search functionality, and various use cases for shopping cart management, payment functions, product favorites, store favorites, posting comments and ratings.

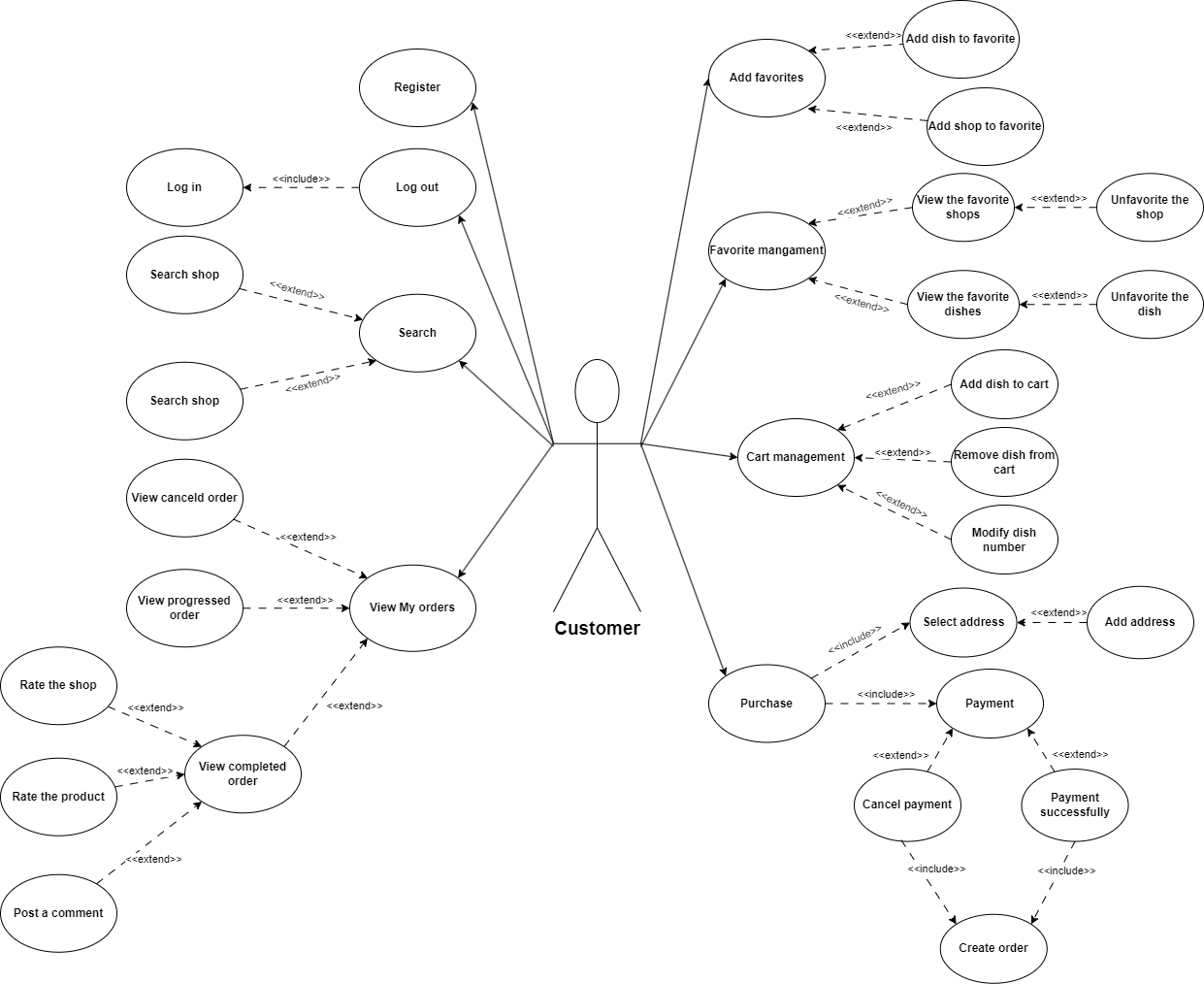


Figure 5: Customer Use case diagram.

* Merchant Use case:

The functions available to restaurant merchants using the platform are illustrated in Figure 6. These include store registration and management, processing orders and updating order status, in addition to generic functions such as logging in, registering for an account and searching and viewing stores.

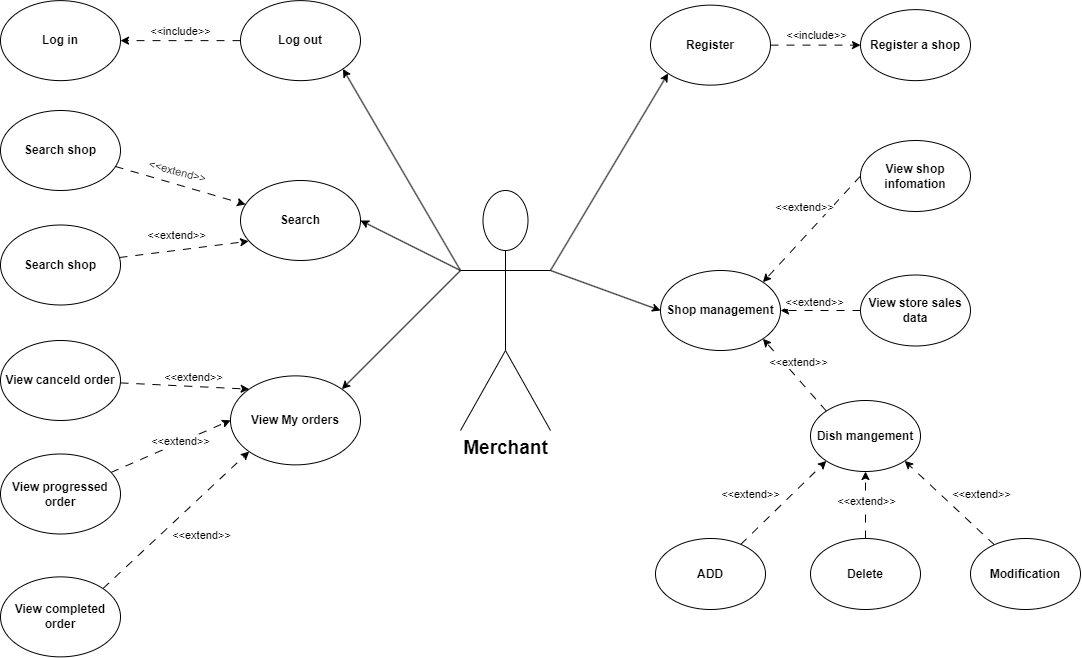


Figure 6: Merchant User case diagram.

* Administrator Use case:

Figure 7 shows a detailed User case for the administrator of the food delivery system, which includes viewing and managing (e.g., adding, deleting, and modifying data) data (e.g., users, stores, products, etc.) in the website.

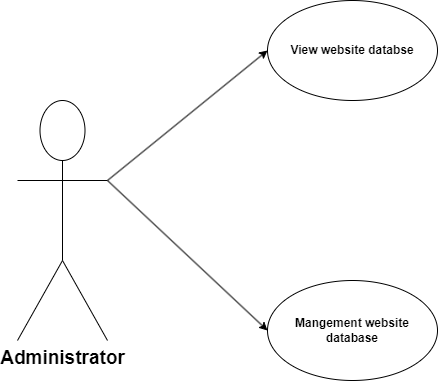


Figure 7: Administrator User case diagram.

### **Database design:**

This section shows the design of the online food delivery website database.

For this online food delivery system, I have designed a database for storing the data to be used in this website (such as user, order, store, product data etc.) The detailed structure of this database is shown in Figure 8. After implementing the database through Mysql, I used Mysql workbench to manage each table in the database.

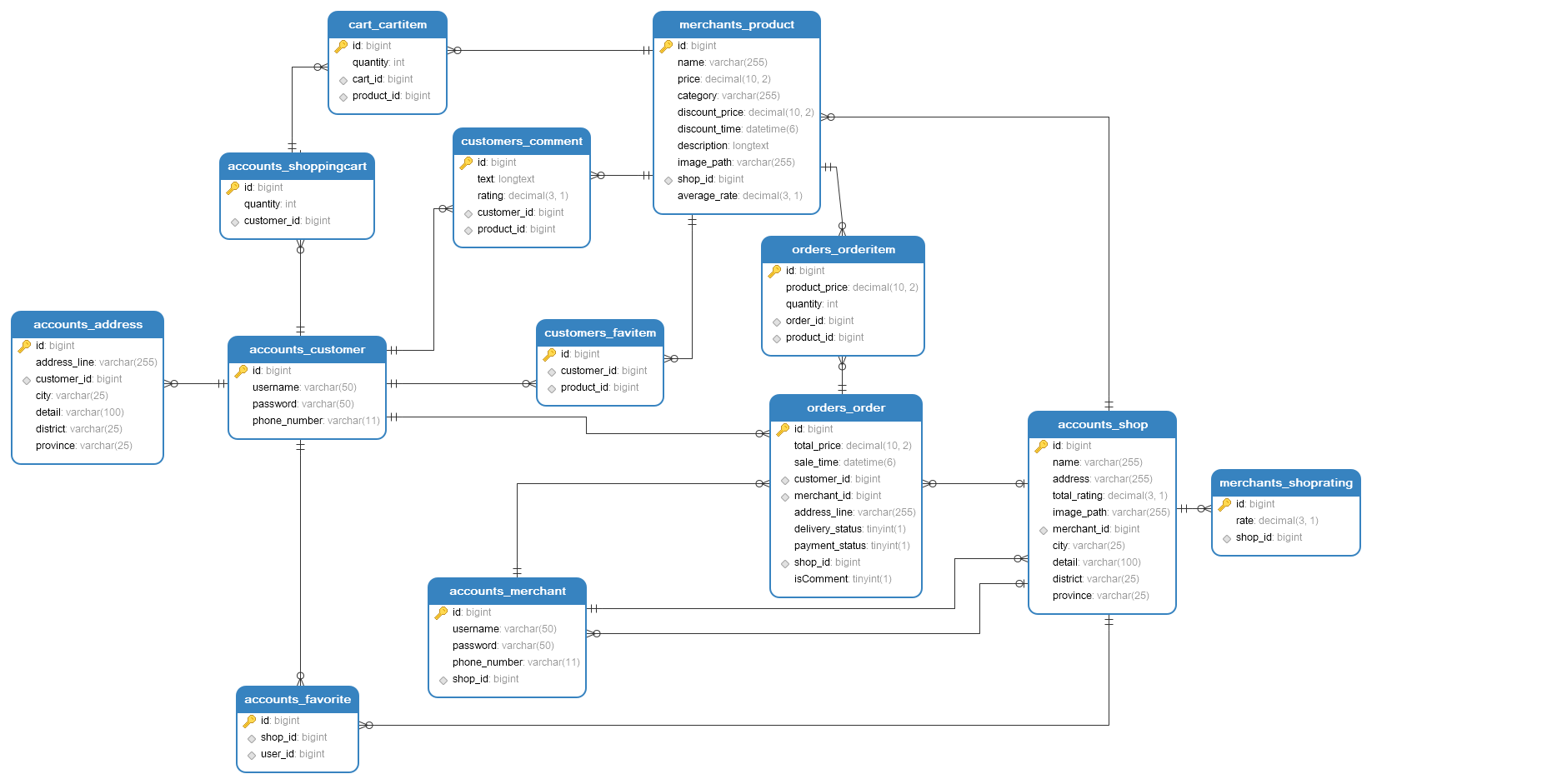


Figure 8: Database structure diagram.

### **Recommendation system approach:**

This section shows in detail the composition of the Hybrid recommendation system implemented in the project, the operation flow and the principle of the recommendation algorithm.

#### Composition of recommendation system

The recommender system in this project consists of the following three recommendation methodologies, namely User-CF, CB and Item-based popularity recommendation. The composition of these methodologies is illustrated in detail in Figure 9.

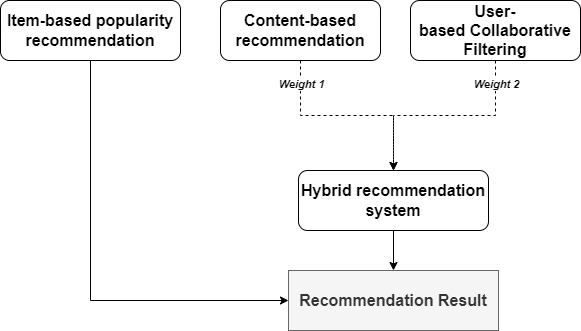


Figure 9: Composition of hybrid recommendation system.

#### Operation flow of recommendation:

Figure 10 depicts the complete operational flow of the recommendation system for making recommendations for a user. The recommended sequence of steps is as follows:

* Step 1: Checks if the user is logged in and what type of user they are:

This step to verifies the user's login status and user type. Only users who are logged in and whose user type is customer have historical behavioral data that can be utilized to analyze and subsequently personalize the recommendation service for them.

* Step 2: Check if the user's historical behavior data is sufficient:

This step checks whether the user's historical behavioral data (such as the number of purchase orders and the number of favorite products, etc.) is sufficiently large, and if it is not sufficiently large, then the use of Item-based popularity recommendation is to avoid the use of Hybrid recommendation to avoid the use of less historical behavioral data to produce a relatively large bias

* Step 3: Hybrid recommendation:

In this step, the system employs two recommendation algorithms (User-CF and CB) to generate recommendations, which are then merged using a weighting method.

* Step 4: Filter out low-rated product:

The objective of this step is to filter the items in the generated recommendation results in order to avoid the recommendation results that have items rated as low by users but are still recommended.

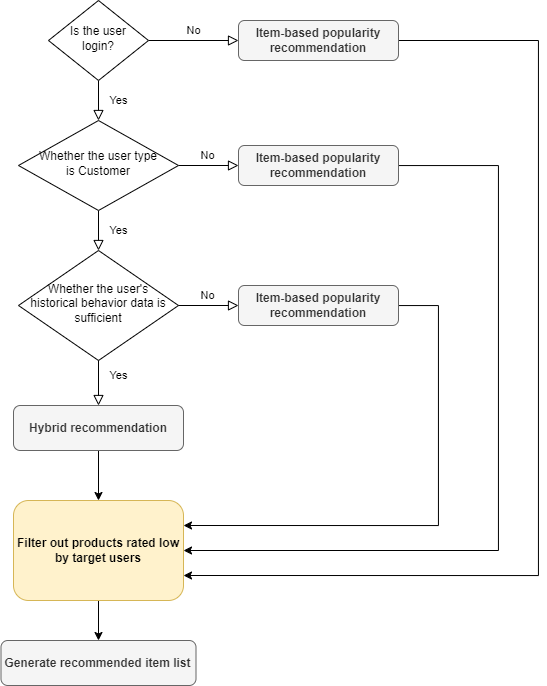


Figure 10: The operation flow of recommendation.

#### Item-based popularity recommendation:

In item-based popularity recommendation, the recommendation score is calculated by extracting the metadata of the item itself. This allows the most popular items to be recommended to the user. In this recommendation function, two operations are used to calculate the score:

* Min-Max Scaling Normalize:

The formula (1) is used for normalizing data values within a specified range. This transformation redirects the features to a fixed range (0,1), which ensures that each feature has an equal impact on the model, thus avoiding that certain features may have a disproportionate impact on the prediction results due to their large range of values.

where x represents an original value, min(x) is the minimum value in the data set, max(x) is the maximum value, and x' is the normalized value.

* Recommend score:

Formula (2) defines how metadata is used in the item-based popularity recommendation function to calculate the recommendation scores of different items according to their weights.

The variables are as follows: ‘w’ represents weight, ‘avgRate’ represents the average rating of the item, ‘favorite’ represents the number of favorites, and ‘sales’ represents the total number of sales of the item.

After the calculation of (1) and (2) above, the products are sorted in descending order according to the value of the recommendation score. Subsequently, the products with high scores are recommended to the users. The advantage of such an operation is that it does not require the user's historical behavioral data, which effectively solves the cold-start problem of the recommender system.

#### User-CF:

For the implementation of User-based Collaborative Filtering recommendation function, we first create item-user table to store the item and the corresponding user who likes the item, then use Adjusted Cosine Similarity to calculate the similarity between the customer users, build the user similarity matrix, and then recommend the item according to the similar user's favorite item to recommend the item according to the similar user's favorite item.

* Item-User table:

As shown in Table 2, the item-user table stores the item id and the corresponding ids of all users interested in the item. the item-user table is implemented to facilitate the later calculation of user similarity and recommendation of items of interest to similar users.

|  |  |
| --- | --- |
| Item id | Set of user ids |
| 1 | {5,12,30} |
| 2 | {2,7} |
| 3 | {3} |

Table 2: example of item-user table

* Adjusted Cosine Similarity:

Traditional recommendation algorithms frequently rely on explicit user feedback (e.g., ratings or reviews) to form "nearest neighbor sets" that predict user preferences. However, these methods can be influenced by arbitrary ratings or incomplete feedback from users that represent their implicit behavior [27]. Consequently, explicit feedback alone does not fully capture user preferences. To address this issue, we propose replacing the traditional cosine similarity with adjust cosine similarity, a formula that incorporates implicit user feedback to enhance prediction accuracy.

As show as in formula (3), Adjusted Cosine Similarity is primarily used in collaborative filtering to measure similarity between two users based on their ratings, adjusting for the average rating of each user to avoid bias from varying user rating scales.

denotes the set of users who rated both items and . where and are the ratings given by user to items and , respectively, and is the average rating of user . This metric compensates for user bias, making it effective for personalized recommendations.

* User similarity matrix:

The user similarity matrix represents the most important component of the User-CF system. Its structure is shown in Table 3, User similarity matrix stores the similarity between different users.

|  |  |  |  |
| --- | --- | --- | --- |
| User id | 1 | 2 | 3 |
| 1 | 0 | 0.346 | 0.157 |
| 2 | 0.346 | 0 | 0.561 |
| 3 | 0.157 | 0.561 | 0 |

Table 3: Example of User similarity matrix

The recommender system can identify other users who are most similar to the target user through the User similarity matrix. This allows the system to obtain the products that these similar users are interested in through the Item-user table.

#### Content-based filtering algorithm:

In the context of content-based recommendation (CBR), the user's behavior is analyzed in order to identify products that may be of interest to them. Metadata associated with these products is then used to calculate the degree of similarity between them, with the aim of identifying products that the user may be interested in. The feature matrix is employed to record the feature matrix of the products, while the Jaccard index is utilized to calculate the similarity between them.

* Feature Matrix:

The feature matrix for content-based filtering is constructed by extracting and combining various item attributes into a structured format. The key features include:

* *Category*: This feature is converted to numeric format by hot encoding. The purpose of adding categories to the feature matrix is to take into account the categories of the items that the user likes.
* *Average Rating*: This feature reflects the overall approval of users of the item.
* *Favorites Count*: As a numerical indicator of the popularity of an item.
* *Total Sales*: This feature is one of the indicators of the popularity of the product.
* *Number of Ratings*: This feature is used to measure the degree of user engagement and feedback on the product.

These features are then integrated into a comprehensive feature matrix, which serves as the foundation for calculating similarities between items.

* User preference vector:

A user preference vector is a mathematical representation of a user’s preferences, constructed from the aggregation of features of items that the user has interacted with. This vector essentially summarizes the user's tastes and preferences in a form that can be easily compared and analyzed against item features.

* Jaccard Index:

The Jaccard Index is a statistical measure used to assess the similarity and diversity of sample sets. As shown in formula (4), the similarity between two sets can be quantified by calculating the ratio of the size of the intersection set to the size of the merged set. A larger ratio indicates a greater degree of similarity between the two sets.

where and are sets. (Intersection) is the number of elements common to both sets, and (Union) is the total number of elements in either set. This index is particularly useful in comparing the similarity and diversity of sample populations.

## **Technology**

The hardware used in the project are as follows:

|  |  |  |  |
| --- | --- | --- | --- |
| **Computer** | **CPU** | **GPU** | **Random Access Memory** |
| ASUS Tianxuan | R7-4800H | GTX 1650ti 4GB | 16GB |

Table 4: Hardware environment of project

The software used in the project are as follows:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Language** | **OS** | **Database** | **Framework** | **Front-end** | **Back-end Third library** | **Testing tools** |
| Python | Windows 10 | Mysql 8.0,  Mysql workbench | Django | Javascript,  jQuery,  Bootstrap,  Feather icon | PyMysql,  Pandas,  Numpy,  scikit-learn | Pytest, Pytest-django |

Table 5: software environment of project

**Image Data set:**

I used the Food image dataset from Kaggle.

Data set Url: <https://www.kaggle.com/datasets/kmader/food41>

## **Project Version Management**.

In this project, I used Github desktop as a project version management software tool and created a repository on the Github platform. This repository stores all the code for the project, the files along with their versions and upload times.

Here is Github repository url: <https://github.com/Blueblue22two/202018010410_project_recommendation-system>

# **Implementation and Results**

Here students are to provide detailed descriptions and documentation of results and testing. Critical evaluation and discussion of results, issues encountered constraints, limitations, and originality.

The subsection layouts of this section mostly depends on the type of project that the student is carrying out. Students can introduce subsections that will help the readability of their work.

For instance, students doing software development-based projects should provide the detailed use of their software in this chapter. Screenshots (images) of their graphical user interfaces can be depicted in this chapter. Other relevant details about the testing and evaluation of their software can be stated here as well.

## **Testing plan:**

### **Unit testing:**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Test id | Test unit | Test Objective | Input case | Except Output | Actual Output | Status |
| 1 | Customer Register | Test registering a customer with valid data | Username=’newcustomer’,  Password=’securepass123,’  Phone=’12345678901’ | Status 200, Response: Register successfully. | Status 200, Response: Register successfully. | **Pass** |
| 2 | Customer Register | Test registering a customer with duplicate username | Username=’ duplicateuser’,  Password=’securepass123,’  Phone=’12345678901’ | Status 409, Response: Already have a same username. | Status 409, Response: Already have a same username. | **Pass** |
| 3 | Merchant Register | Test registering a customer with valid data | Username=’newmerchant’,  Password=’securepass123,’  Phone=’ 98765432101’ | Status 200, Response: Register successfully. | Status 200, Response: Register successfully. | **Pass** |
| 4 | Merchant Register | Test registering a merchant with invalid data | Username = Null,  Password = ’securepass123,’  Phone = ‘notaphone’ | Status 400, Response: Register successfully. | Status 400, Response: Register successfully. | **Pass** |
| 5 | Customer Login | Test login with correct data | Username = 'testuser', password = 'testpass'  userType = ‘1’ | Status 200, Response: testuser login successfully. | Status 200, Response: testuser login successfully. | **Pass** |
| 6 | Customer Login | Test login with wrong password | Username = 'testuser', password = 'wrongpass'  userType = ‘1’ | Status 400, Response: Invalid username or password. | Status 400, Response: Invalid username or password. | **Pass** |
| 7 | Customer  Log out | Test logged-in users for logging out | Username = ‘testuser’ | Status 302, Response: Logout successfully. | Status 302, Response: Logout successfully. | **Pass** |
| 8 | Add product to Cart | Test adding a product to the shopping cart | Instance of customer = ’testuser’  Instance of product=’Test Product1’ | Status 200, Response: Product added to cart successfully. | Status 200, Response: Product added to cart successfully. | **Pass** |
| 9 | Modify cart item | Test modifying the quantity of a cart item | Customer ID,  Product ID,  Quantity of product | Status 200, Response: Quantity updated successfully. | Status 200, Response: Quantity updated successfully. | **Pass** |
| 10 | Add Address | Test adding a new address for customer | Province='TestProvince', City='TestCity', District='TestDistrict', Detail='123 My Street' | Status 302, Redirect to cart page. | Status 302, Redirect to cart page. | **Pass** |
| 11 | Add Favorite Product | Test adding a product to favorites | Product ID of 'Test Product12' | Status 200, Response: Product added to favorites successfully! | Status 200, Response: Product added to favorites successfully! | **Pass** |
| 12 | Cancel Favorite Product | Test canceling a favorite product | Product ID of 'Test Product12' already favorited | Status 200, Response: Product favorite successfully cancelled | Status 200, Response: Product favorite successfully cancelled | **Pass** |
| 13 | Favorite a product repeatedly | Test adding a product to favorites twice | Product ID of 'Test Product12' | Status 200, Response: 'You have already added this product to favorites.' | Status 200, Response: 'You have already added this product to favorites.' | **Pass** |
| 14 | Cancel Non-existent Shop Favorite | Test canceling a favorite for a non-existent shop | Shop name = 'Definitely Not a Real Shop' | Status 404, Response: 'Shop does not exist.' | Status 404, Response: 'Shop does not exist.' | **Pass** |
| 15 | New merchant user create shop | Test creating a new shop | Username='TestMerchant2', Password='password123456', Phone='22345678901', Shop Data including name, address, and image file | Status 302, Response: Store created successfully. | Status 302, Response: Store created successfully. | **Pass** |
| 16 | Add a product | Test adding a new product by merchant user | Username='TestMerchant1', Shop=Shop object, Product details including name, price, category, and image file | Status 200, Response: add Product successfully. | Status 200, Response: add Product successfully. | **Pass** |
| 17 | Modify product data | Test modifying a product name and price | Username='TestMerchant1', Product=Product object, Updated product name and price | Status 200, Response: Product updated successfully. | Status 200, Response: Product updated successfully. | **Pass** |
| 18 | Delete product | Test deleting a product | Username='TestMerchant1', Product=Product object | Status 200, Response: Product removed successfully. | Status 200, Response: Product removed successfully. | **Pass** |
| 19 | Delete a non-existing product | Test deleting a non-existing product | Username='TestMerchant1', Invalid Product ID = 9999 | Status 404, Response: Product not found | Status 404, Response: Product not found | **Pass** |
| 20 | Generate Payment and Order | Test generating a payment and creating an order | Username='TestUser1', Password='password123', Phone='12345678901', ProductId=product.id, Quantity=2, TotalPrice=42.0, AddressId=address.id | Status 200, Redirect to payment view, Order and OrderItem created in database | Status 200, Redirect to payment view, Order and OrderItem created in database | **Pass** |
| 21 | Post comment and rating |  |  |  |  | **Pass** |
| 22 |  |  |  |  |  | **Pass** |
| 23 |  |  |  |  |  | **Pass** |
| 24 |  |  |  |  |  | **Pass** |
| 25 |  |  |  |  |  | **Pass** |
| 26 |  |  |  |  |  | **Pass** |

### **Functional testing:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test id | Test function | Test scenario | Except Result | Status |
| 1 | User Registration | User fills in valid information and submits registration form. | After the user submits the form and successfully registers, the corresponding user session is created and the account is automatically logged in and redirected to the main page. | **Pass** |
| 2 | User Login and Validation | Registered users select the correct user type to log in, then enter the correct username and password to log in. | Returns a successful login response and redirects the page to the main page. | **Pass** |
| 3 | User Logout | Logged-in users choose to log out of their accounts. | Log out successfully and redirect to the main page, then delete the logged-in user information stored in the session. | **Pass** |
| 4 | Add product to Shopping Cart | Customer adds a product to the shopping cart and checks if it's successfully added. | The product is successfully added to the shopping cart. | **Pass** |
| 5 | Modify the number of products in cart | User modifies the quantity of a product in the cart | The cart successfully updates with the new quantity. | **Pass** |
| 6 | Add New Address | Customer selects the address details and submits the form to add a new address. | The address was saved successfully and redirected to the shopping cart page. | **Pass** |
| 7 | Add Favorite Product | Customer chooses a product to add to their favorites | The product is added to favorites, and the customer receives a success message. | **Pass** |
| 8 | Cancel Favorite Product | Customer removes a product from their favorites | The favorite product is successfully removed, and a confirmation message is displayed. | **Pass** |
| 9 | Create new shop | Merchant user submits a new shop registration information form. | After submitting and validating the form, the user is redirected to this new shop page. | **Pass** |
| 10 | Add product | Merchant users add new products by filling out the new product information and uploading the form to the product management page. | The product is successfully added to the database. The successful addition response is returned. | **Pass** |
| 11 | Modify Product | Merchant updates information about existing products on the Product Management page. | The corresponding product details in the database will be updated and the product update success response will be returned. | **Pass** |
| 12 | Delete product | Merchant deletes an existing product. | Product data is successfully removed from the database and a response is returned that the product was successfully deleted. | **Pass** |
| 13 | Generate Payment and Order | Customer clicks on the payment function on the shopping cart page to generate the appropriate payment. | The corresponding order data is generated in the database and redirected to the payment page. | **Pass** |
| 14 | Post comment and rating |  |  | **Pass** |
| 15 |  |  |  | **Pass** |
| 16 |  |  |  | **Pass** |

### **Performance testing:**

### **Security testing:**

## **Graphical user interfaces of the website:**

## **Recommendation system performance analysis and evaluation:**

# **Professional Issues**

## **Project Management**

### **Activities**

|  |  |  |  |
| --- | --- | --- | --- |
| Objective | Detail | Start data | End date |
| 1. Investigation on the Existing Food Delivery Platform | Researched prominent food delivery platforms (Meituan, Ele.me, Uber Eats). | 2023/10/16 | 2023/10/24 |
| 1. Comparison between food delivery platforms | Compared features, market positions, and user feedback of selected platforms. | 2023/10/20 | 2023/10/28 |
| 1. Research on recommendation system | Study and analyzes different types of recommendation systems and explores the application of these methods in takeout platforms. | 2023/10/24 | 2023/10/29 |
| 1. Write Project proposal | Developed and submitted a comprehensive project proposal. | 2023/10/25 | 2023/11/3 |
| 1. Research on CF and Content-based filtering | Investigated the workings of Collaborative Filtering (CF) and Content-based filtering algorithms | 2023/10/29 | 2023/11/10 |
| 1. Function requirements analysis | Analyze the website and different types of users and record their software requirements. | 2023/11/1 | 2023/11/18 |
| 1. System design | Design the website architecture by breaking it down into the following modules:   1. Customer module 2. Merchant module 3. Store module 4. Product module 5. Order module 6. Recommendation module | 2023/11/14 | 2023/11/30 |
| 1. Database design | 1. Based on Function requirements analysis design the framework of database mode. 2. Implement the database by Mysql. 3. Connect database with Django. | 2023/11/20 | 2023/11/26 |
| 1. Web Implementation (Implement the front-end and back-end) | 1. Web UI design 2. Completed the front-end user interface for customers, merchants, and administrators. 3. Implemented back-end function. | 2023/12/1 | 2023/2/10 |
| 1. Progress report | Implement the progress report. | 2023/12/10 | 2023/12/27 |
| 1. Recommend function design | Design the architecture of the recommendation system, including the use of user-based collaborative filtering recommendation and content-based filtering. | 2024/1/2 | 2024/1/10 |
| 1. Develop recommend function | 1. Developed and implemented the two main components of the recommendation system: the user collaborative filtering algorithm and the content-based filtering algorithm.  2. Integrated these algorithms into the website's backend, ensuring proper data circulation and updates. | 2024/1/11 | 2024/3/1 |
| 1. Test and evaluate website function | Planned testing and evaluation for the collaborative filtering recommendation. | 2024/3/1 | 2024/3/10 |
| 1. System testing & performance analysis | Conduct comprehensive system testing of the website, including functional testing, user interface testing and performance testing. | 2024/3/11 | 2024/3/15 |
| 1. Write Final report | Implement the final report | 2024/3/18 | 2024/4/6 |
| 1. Create Poster | Will design and create a poster for project presentation. | 2024/4/6 | 2024/4/15 |

Table 6: Activity table

### **Schedule**



Figure 11: Gantt chart of activity

### **Project Data Management**

In this section, students must describe how they have used resources such as Baidu drive, Gitee, etc., to manage project logs, reports, literature, etc.

* For Weekly reports, I upload them to the Weekly report folder every week.
* Upload all the reports to the Reports folder.
* Upload all the Reference files to the References folder.
* Upload all the code to a code file and do version management.
* Upload other files (such as charts) to the others folder.
* Upload the files related to Presentation to the Presentation folder.
* Upload UI-related files to the UI folder.

The figure 4 below shows the file format in the repository in detail.

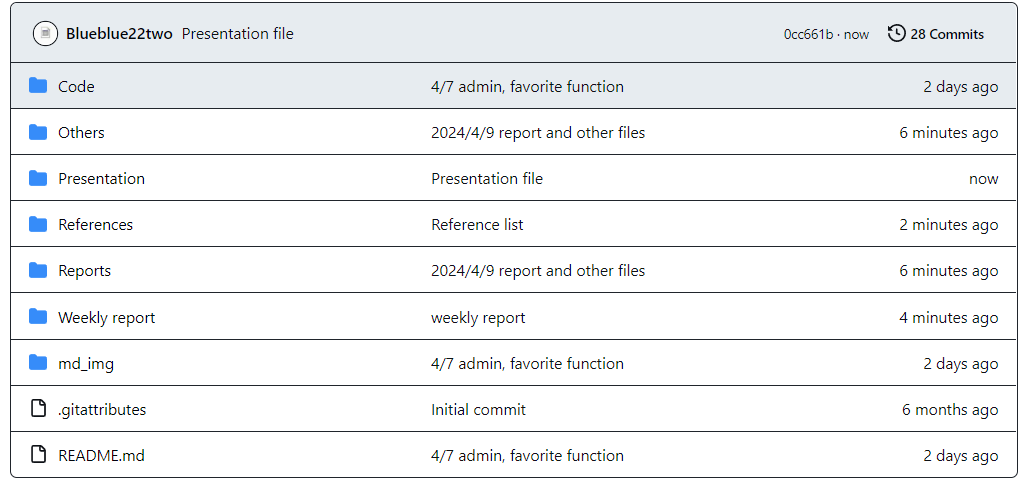


Figure 12: Github project repository

### **Project Deliverables**

|  |  |
| --- | --- |
| Type | Content |
| **Document** | Project proposal |
| Progress report |
| Final report |
| Weekly report |
| Ethical from |
| Presentation & poster |
| Reference files |
| **Code** | Database sql file |
| Front-end file and code |
| Back-end code |

Table 7: Project deliverable

## **Risk Analysis**

Risk analysis as informed by the current project progress; Resolved risks and the success of the mitigation strategy; Changes to the project plan as a result of risks; Future risks.

This section shows a risk analysis and mitigation strategies for the project.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Risk ID | Potential Risk | Cause ID | Potential Causes | Severity | Likelihood | Risk | Mitigation ID | Mitigation |
| R1.1 | User dissatisfaction with recommendation system | C1.1.1 | Poor algorithm performance | 3 | 3 | 9 | M1.1.1 | Regularly evaluate and update the recommendation algorithm based on user feedback and usage patterns. |
| C1.1.2 | Inadequate user data collection | 2 | 2 | 4 | M2.1.2 | Implement robust data collection mechanisms and ensure transparency in data usage policies to gain user trust. |
| R1.2 | Low user engagement with the platform | C1.2.1 | Limited variety of restaurants and cuisines | 3 | 2 | 6 | M1.2.1 | Expand the range of available restaurants and cuisines through partnerships and collaborations. |
| C1.2.2 | Poor user interface design | 2 | 3 | 6 | M1.2.2 | Invest in user experience design and conduct usability testing to enhance the platform's ease of use. |
| R1.3 | Technical infrastructure issues | C1.3.1 | Unanticipated scalability challenges | 4 | 3 | 12 | M1.3.1 | Conduct thorough scalability testing and implement scalable architecture from the beginning. |
| C1.3.2 | Third-party service failures | 3 | 2 | 6 | M1.3.2 | Have backup plans and redundancies for critical third-party services. |
| R1.4 | Security vulnerabilities | C1.4.1 | Poor data encryption practices | 4 | 3 | 12 | M1.4.1 | Implement strong encryption methods for sensitive user data. |
| C1.4.2 | Insufficient user authentication measures | 3 | 2 | 6 | M1.4.2 | Enhance user authentication protocols and incorporate multi-factor authentication. |
| R1.5 | Legal and regulatory compliance issues | C1.5.1 | Inadequate understanding of local food safety regulations | 3 | 3 | 9 | M1.5.1 | Engage legal experts to ensure compliance with local food safety regulations and standards. |
| C1.5.2 | Privacy concerns and data protection laws | 4 | 2 | 8 | M1.5.2 | Implement robust privacy policies and obtain explicit user consent for data processing. |

Table 8: Risk table

## **Professional Issues**

Identification and discussion of relevant legal, social, ethical, and environmental issues in the context of the project. Refer to professional codes of conduct, e.g. BCS, ACM.

### Legal Issues

Identification:

In the context of our online food delivery platform project, legal issues primarily revolve around Intellectual Property Rights (IPR), Health & Safety (H&S), and compliance with data protection laws, including GDPR.

Discussion:

IPR: Careful attention will be given to the use of proprietary information and software. Unauthorized use may lead to legal consequences, necessitating thorough validation of research documents, source code, and external libraries. Compliance with licenses such as MIT License and copyright laws is crucial for lawful use.

Data Protection (GDPR): Adherence to GDPR regulations is paramount, ensuring the secure handling of user data, privacy, and informed consent.

### Social Issues

Identification:

Social issues in our project involve considerations of social responsibility and the prevention of unintended biases and discriminatory behavior.

Discussion:

Social Responsibility: The application's design and development will adhere to the Web Accessibility Initiative, ensuring accessibility, user-friendliness, and inclusivity for all users.

### Ethical Issues

Identification:

Ethical considerations focus on the responsible use of research and developments, particularly emphasizing educational and research purposes and avoiding risks associated with malfunctioning Machine Learning (ML) models.

Discussion:

Educational and Research Purpose: Emphasis on the strictly educational and research-oriented nature of the project to mitigate risks associated with ML models.

Avoidance of Commercial Use: Clear communication that the product is not for commercial use and should not influence or affect financial or investment decisions.

### Environmental Issues

Identification:

While not explicitly stated in the given material, environmental issues may arise from the potential environmental impact of the project.

Discussion:

Environmental Impact: Consideration of the project's environmental impact, such as energy consumption and resource usage, to minimize any negative effects.

# **Conclusion**

Summary of what was achieved and potential future work.

# **References**

[1] H. Ko, S. Lee, Y. Park, and A. Choi, ‘A Survey of Recommendation Systems: Recommendation Models, Techniques, and Application Fields’, *Electronics*, vol. 11, no. 1, Art. no. 1, Jan. 2022, doi: 10.3390/electronics11010141.

[2] M. E. B. H. Kbaier, H. Masri, and S. Krichen, ‘A Personalized Hybrid Tourism Recommender System’, in *2017 IEEE/ACS 14th International Conference on Computer Systems and Applications (AICCSA)*, Hammamet: IEEE, Oct. 2017, pp. 244–250. doi: 10.1109/AICCSA.2017.12.

[3] R. S. Mulyana, A. Id Hadiana, and E. Ramadhan, ‘Recommendation System Of Product Sales Ideas For MSMEs Using Content-based Filtering and Collaborative Filtering Methods’, in *2023 International Conference on Computer Science, Information Technology and Engineering (ICCoSITE)*, Jakarta, Indonesia: IEEE, Feb. 2023, pp. 252–256. doi: 10.1109/ICCoSITE57641.2023.10127844.

[4] F. Cacheda, V. Carneiro, D. Fernández, and V. Formoso, ‘Comparison of collaborative filtering algorithms: Limitations of current techniques and proposals for scalable, high-performance recommender systems’, *TWEB*, vol. 5, p. 2, Jan. 2011.

[5] H. Liu, ‘Implementation and Effectiveness Evaluation of Four Common Algorithms of Recommendation Systems - User Collaboration Filter, Item-based Collaborative Filtering, Matrix Factorization and Neural Collaborative Filtering’, in *2022 International Conference on Cloud Computing, Big Data Applications and Software Engineering (CBASE)*, Suzhou, China: IEEE, Sep. 2022, pp. 224–227. doi: 10.1109/CBASE57816.2022.00049.

[6] Y. Kawada, ‘Cosine similarity and the Borda rule’, *Soc Choice Welf*, vol. 51, no. 1, pp. 1–11, Jun. 2018, doi: 10.1007/s00355-017-1104-2.

[7] M. Schwarz, M. Lobur, and Y. Stekh, ‘Analysis of the effectiveness of similarity measures for recommender systems’, in *2017 14th International Conference The Experience of Designing and Application of CAD Systems in Microelectronics (CADSM)*, Lviv - Polyana, Ukraine: IEEE, 2017, pp. 275–277. doi: 10.1109/CADSM.2017.7916133.

[8] M. Rezvani and S. M. Hashemi, ‘Enhancing Accuracy of Topic Sensitive PageRank Using Jaccard Index and Cosine Similarity’, in *2012 IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology*, Macau, China: IEEE, Dec. 2012, pp. 620–624. doi: 10.1109/WI-IAT.2012.166.

[9] ‘Django documentation’. Django Software Foundation, 2022. [Online]. Available: https://docs.djangoproject.com/en/5.0/

[10] L. Wang and B. Yi, ‘Research on O2O take-away restaurant recommendation system: taking ele.me APP as an example’, *Cluster Comput*, vol. 22, no. S3, pp. 6069–6077, May 2019, doi: 10.1007/s10586-018-1814-y.

[11] Statista Market Insights, ‘Online Food Delivery: market data & analysis’, Statista Market Insights, Mar. 2023. Accessed: Oct. 30, 2023. [Online]. Available: https://www.statista.com/study/40457/food-delivery/

[12] X. Li, ‘Research on the Application of Collaborative Filtering Algorithm in Mobile E-Commerce Recommendation System’, in *2021 IEEE Asia-Pacific Conference on Image Processing, Electronics and Computers (IPEC)*, Dalian, China: IEEE, Apr. 2021, pp. 924–926. doi: 10.1109/IPEC51340.2021.9421092.

[13] C. N. Sánchez, J. Domínguez-Soberanes, A. Arreola, and M. Graff, ‘Recommendation System for a Delivery Food Application Based on Number of Orders’, *Applied Sciences*, vol. 13, no. 4, p. 2299, Feb. 2023, doi: 10.3390/app13042299.

[14] M. Li and L. Wang, ‘A Survey on Personalized News Recommendation Technology’, *IEEE Access*, vol. 7, pp. 145861–145879, 2019, doi: 10.1109/ACCESS.2019.2944927.

[15] Z. Huang, H. Chen, and D. Zeng, ‘Applying associative retrieval techniques to alleviate the sparsity problem in collaborative filtering’, *ACM Trans. Inf. Syst.*, vol. 22, no. 1, pp. 116–142, Jan. 2004, doi: 10.1145/963770.963775.

[16] J. Salter and N. Antonopoulos, ‘CinemaScreen Recommender Agent: Combining Collaborative and Content-Based Filtering’, *IEEE Intell. Syst.*, vol. 21, no. 1, pp. 35–41, Jan. 2006, doi: 10.1109/MIS.2006.4.

[17] J. N. Bondevik, K. E. Bennin, Ö. Babur, and C. Ersch, ‘A systematic review on food recommender systems’, *Expert Systems with Applications*, vol. 238, p. 122166, Mar. 2024, doi: 10.1016/j.eswa.2023.122166.

[18] C. Song, Q. Yu, E. Jose, J. Zhuang, and H. Geng, ‘A Hybrid Recommendation Approach for Viral Food Based on Online Reviews’, *Foods*, vol. 10, p. 1801, Aug. 2021, doi: 10.3390/foods10081801.

[19] Q. Pu and B. Hu, ‘Intelligent Movie Recommendation System Based on Hybrid Recommendation Algorithms’, in *2023 International Conference on Ambient Intelligence, Knowledge Informatics and Industrial Electronics (AIKIIE)*, Ballari, India: IEEE, Nov. 2023, pp. 1–5. doi: 10.1109/AIKIIE60097.2023.10389982.

[20] N. Wayan Priscila Yuni Praditya, A. Erna Permanasari, and I. Hidayah, ‘Designing a tourism recommendation system using a hybrid method (Collaborative Filtering and Content-Based Filtering)’, in *2021 IEEE International Conference on Communication, Networks and Satellite (COMNETSAT)*, Purwokerto, Indonesia: IEEE, Jul. 2021, pp. 298–305. doi: 10.1109/COMNETSAT53002.2021.9530823.

[21] Y. Shao and Y. Xie, ‘Research on Cold-Start Problem of Collaborative Filtering Algorithm’, in *Proceedings of the 2019 3rd International Conference on Big Data Research*, Cergy-Pontoise France: ACM, Nov. 2019, pp. 67–71. doi: 10.1145/3372454.3372470.

[22] Statista, ‘Catering-services-in-china’, Statista. Accessed: Nov. 01, 2023. [Online]. Available: https://www.statista.com/study/63164/catering-services-in-china/

[23] ‘Meituan Waimai’. Meituan. Accessed: Oct. 31, 2023. [Online]. Available: https://waimai.meituan.com/

[24] ‘Ele.me’. Ele.me. Accessed: Oct. 31, 2023. [Online]. Available: https://www.ele.me

[25] ‘Uber Eats’. Uber Eats. Accessed: Oct. 31, 2023. [Online]. Available: https://www.ubereats.com

[26] Just Eat, ‘Just Eat’. Just Eat. [Online]. Available: https://www.just-eat.ie/

[27] R. Ji, Y. Tian, and M. Ma, ‘Collaborative Filtering Recommendation Algorithm Based on User Characteristics’, in *2020 5th International Conference on Control, Robotics and Cybernetics (CRC)*, Wuhan, China: IEEE, Oct. 2020, pp. 56–60. doi: 10.1109/CRC51253.2020.9253466.

* The layout above is a suggestion of how to present your Final Project Report. Whenever appropriate, introduce sections that will help the readability of your work.
* The Length of the final report should be **8000 – 10000 words**.
* All sections and subsections should be numbered for cross-referencing purposes.
* Regarding citations and references, students must adhere to the University guidelines or IEEE referencing style. **Students doing software development-based projects can cite related websites, web applications, developer documentation, etc. They can cite related articles to their projects, but it is not required. Students doing research-oriented projects should focus on citing research articles. They can also cite appropriate websites whenever necessary. Students are advised to use appropriate reference management software such as Mendeley Reference Manager or Zotero to ensure the correctness of all references.**

## **Formatting Requirements**

Your written report must be presented in the following format:

* All main sections/chapters should begin on a new page. The Declaration page, Tables of Contents pages, Acknowledgment, Abstract, Abbreviation, Glossary, Project Chapters (Chapters 1 to 6), and Appendices should all start on a new page.
* It must be word-processed in 11-point Arial font.
* It must be black text on a white or ivory background
* All pages must be numbered. Follow the appropriate page numbering format specified in the template.
* Margins must be as follows: Top: 1 inch, Bottom: 1 inch (2.5 cm), Left: 1.25 inches, Right:
* 1.25 inches (3.2 cm)
* Use a line spacing of 1.5
* Numbers and captions to figures and tables should be at the bottom of the figure or table. If the figure or table is mounted sideways into the report, then its bottom is on the right-hand side of the report. **All tables and figures must be labeled**.
* Normally, the report should not contain more than 80 tables/figures.

## **Written Presentation**

* The final project report must have a concise written presentation and referencing style.
* It should also have a clear & logical presentation.

**NOTE:**

1. **All the text in red colour are basic guidelines and must be DELETED after using this guide.**
2. **Finally, update the “Table of Contents” appropriately to display the correct section titles and corresponding page numbers.**

# **Appendices**

This section can have the essential information/data that are necessary to be included within the report but would disrupt the flow of the main argument. This section is not marked. Examples include links to data and software repositories, questionnaires, raw survey results, and wireframes.

**总字数8000-10000**