Software Requirements Specification

for

A Restaurant Recommendation System

Version 1.0 approved

Prepared by Chahak Garg, Laiba Shaikh, Shashank Tiwari, and Taha Cyclewala

SVIIT

<22/09/2023>

Table of Contents

1. Introduction

1.1 Purpose

The purpose of a restaurant recommendation system is to help individuals or users discover dining options that align with their preferences, tastes, and requirements. These systems use algorithms and data analysis techniques to suggest restaurants based on various factors, including:

User Preferences: Recommendation systems consider a user's past dining experiences, ratings, reviews, and interactions with the platform to understand their preferences. This personalization helps users discover restaurants that are likely to appeal to them.

Location: Recommender systems often take into account the user's current location or a specified location to recommend restaurants that are nearby or within a certain distance.

Cuisine and Food Type: Users can specify their preferred cuisine or type of food they're interested in (e.g., Italian, Mexican, vegetarian, seafood), and the system will suggest restaurants that serve those types of dishes.)

Budget: Users can set their budget preferences, allowing the system to recommend restaurants that match their price range, whether it's fine dining, casual dining, or budget-friendly options.

1.2 Document Conventions

Entity document should be justified.

- **Convention for main title:**

1. Front Face: Time New Roman
2. Front Style: Bold
3. Front Size: 18

- **Convention for subtitle:**

1. Front Face: Time New Roman
2. Front Style: Bold
3. Front Size: 14

- **Convention for body:**

1. Front Face: Time New Roman
2. Front Style: Normal
3. Front Size: 12

1.3 Intended Audience and Reading Suggestions

The intended audience for information on restaurant recommendation systems can vary widely, as it can be of interest to several groups of people and professionals. Here are some potential target audiences and reading suggestions for each:

General Consumers/Restaurant-Goers: If you're a food enthusiast looking to discover new dining experiences, you might enjoy articles and blog posts on food and travel websites that discuss restaurant recommendation apps and services like Yelp, TripAdvisor, or Zomato. These platforms often provide user-generated restaurant reviews and recommendations.
You can also explore food and travel magazines, websites, and TV shows that feature restaurant reviews and recommendations from food critics and travel experts.

Tech Enthusiasts: If you're interested in the technology behind restaurant recommendation systems, you can start with articles and research papers in the field of machine learning, natural language processing, and recommendation algorithms.
Look for academic journals, conferences, and books on topics related to recommendation systems, data mining, and artificial intelligence. "Recommender Systems Handbook" by Francesco Ricci, Lior Rokach, and Bracha Shapira is a comprehensive resource in this area.

Restaurant Owners and Managers: Restaurant owners and managers may benefit from articles and books on restaurant marketing and management that discuss the importance of online reviews and how to optimize their presence on review platforms.
They can also explore resources specific to restaurant technology and POS (Point of Sale) systems that integrate with recommendation features.

Data Scientists and Machine Learning Practitioners: For those interested in developing or working with recommendation systems, you can dive into research papers and books on machine learning and recommendation algorithms.
Explore online courses and tutorials on platforms like Coursera, edX, and Udacity, which offer courses on recommendation systems and data science.

Entrepreneurs and Startups: If you're considering creating a restaurant recommendation app or service, it's essential to understand the market and business aspects. Look for resources on startup strategy, market research, and app development.
Books like "Lean Startup" by Eric Ries and "Zero to One" by Peter Thiel offer valuable insights for aspiring entrepreneurs.

Academics and Researchers: Researchers in the field of recommendation systems may want to explore academic journals, conference proceedings, and research papers on the latest advancements in the area.
Attend relevant conferences and workshops such as the ACM RecSys Conference, which focuses on recommendation systems research.

Policy Makers and Regulators: Those involved in regulating or overseeing the restaurant and food service industry can benefit from reports and studies on the impact of restaurant recommendation systems on consumer behavior, privacy concerns, and market competition.

1.4 Product Scope

The product scope for a restaurant recommendation system can vary depending on the goals and objectives of the development team or organization. Here's an overview of the key components and features that can be included in the scope of such a system:

User Profiles:

- User Registration: Allow users to create accounts with their personal information and preferences.

- User Authentication: Implement secure login and authentication methods.

Restaurant Data:

- Restaurant Database: Maintain a comprehensive database of restaurants, including details such as name, location, cuisine, price range, opening hours, and contact information.
- Menu Information: Optionally, provide access to restaurant menus, including dishes, prices, and descriptions.

Recommendation Engine:

- Personalization: Develop recommendation algorithms that consider user preferences, location, past dining history, and reviews.
- Machine Learning Models: Implement machine learning models to improve recommendation accuracy.
- Real-time Updates: Ensure that recommendations are updated in real-time or at regular intervals.

User Interactions:

- Search Functionality: Allow users to search for restaurants based on criteria like cuisine type, location, price range, dietary preferences, and more.
- Ratings and Reviews: Enable users to rate and review restaurants, contributing to the recommendation system's data.
- Bookmarking/Favorites: Allow users to save and track their favorite restaurants.
- Check-ins: Implement a feature for users to check in when they visit a restaurant, sharing their experiences with friends or the community.

Location Services:

- GPS Integration: Utilize GPS or location data to provide location-based recommendations and directions to nearby restaurants.
- Geofencing: Offer location-based notifications or promotions when users are near specific restaurants.

1.5 References

"Matrix Factorization Techniques for Restaurant Recommendations" by Simon Foucher and George Karypis. This paper explores matrix factorization methods for restaurant recommendations.

"Location-Aware Restaurant Recommendation" by Yu Zheng, et al. It discusses location-based recommendation techniques for restaurants, which are important for personalized recommendations.

"A Survey of Recommender Systems in E-Commerce" by Paolo Cremonesi, et al. Although it covers a broader scope, it provides insights into recommender systems that can be applied to restaurant recommendation.

2. Overall Description

2.1 Product Perspective

The product perspective for a restaurant recommendation system involves considering its position and role within the broader context of the food and dining industry. Here are key aspects to consider from a product perspective:

User-Centric Approach: Focus on the user experience and how the system caters to the needs and preferences of diners. Prioritize user satisfaction and engagement.

Integration with Existing Platforms: Consider how the recommendation system can integrate with existing restaurant review and booking platforms, as well as food delivery services. Seamless integration enhances convenience for users.

Mobile and Web Compatibility: Ensure that the recommendation system is accessible through both mobile applications and web interfaces to cater to a broader audience.

Data Sources and Partnerships: Identify potential data sources for restaurant information, such as APIs from review platforms, and establish partnerships or data-sharing agreements where necessary.

Scalability and Performance: Plan for scalability to handle increased user traffic and data as the system gains popularity. Scalability ensures that the product remains efficient and responsive.

Data Privacy and Security: Implement robust data security measures to protect user information, especially if the system stores personal data or preferences.

Monetization Strategy: Define the product's monetization strategy, which could include advertising, premium subscriptions, or affiliate marketing with restaurants.

2.2 Product Functions

A restaurant recommendation system can encompass various functions to provide users with valuable recommendations and enhance their dining experiences. Here are the key product functions to consider:

Restaurant Data Management: Maintain a comprehensive database of restaurants, including details like restaurant names, locations, cuisines, price ranges, opening hours, and contact information.

Search and Filtering: Implement search functionality that allows users to search for restaurants based on criteria such as cuisine type, location, price range, dietary preferences, and user ratings.

Personalized Recommendations: Utilize recommendation algorithms to provide users with personalized restaurant suggestions based on their preferences, past dining history, and user behavior.

User Interactions: Enable users to rate and review restaurants, bookmark their favorite places, and check in when they visit a restaurant, sharing their experiences with the community.

Location-Based Services: Use GPS or location data to offer location-based recommendations and provide directions to nearby restaurants.

Data Flow Diagram

DFDs are used to depict graphically the data flow in a system [43]. It explains the processes involved in a system from the input to the report generation. It shows all possible paths from one entity to another of aa system. The detail of a data flow diagram can be represented in three different levels that are numbered 0, 1 and 2.

There are many types of notations to draw a data flow diagram among which YourdonCoad and Gane-Sarson method are popular. The DFDs depicted in this chapter uses the Gane-Sarson DFD notations.

Data Flow Diagram – Level 0

DFD level 0 is called a Context Diagram. It is a simple overview of the whole system being modeled. Fig 2.2.1 shows the DFD level 0 of the system.

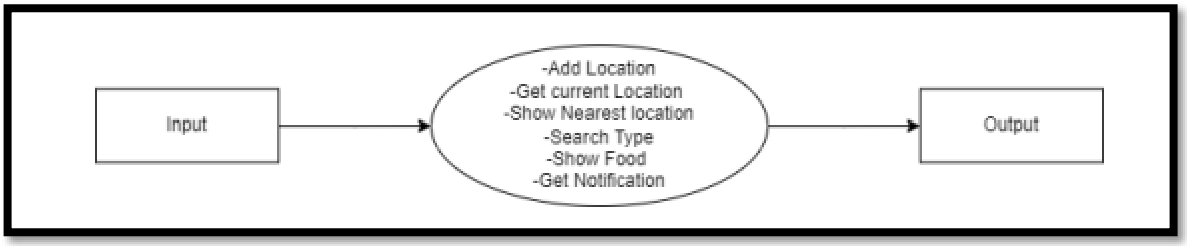

Fig 2.2.1: DFD - level 0

Data Flow Diagram – Level 1

DFD level 1 gives a more detailed explanation of the Context diagram. The high-levelprocess of the Context diagram is broken down into its subprocesses. The DFD level 1of the system is depicted

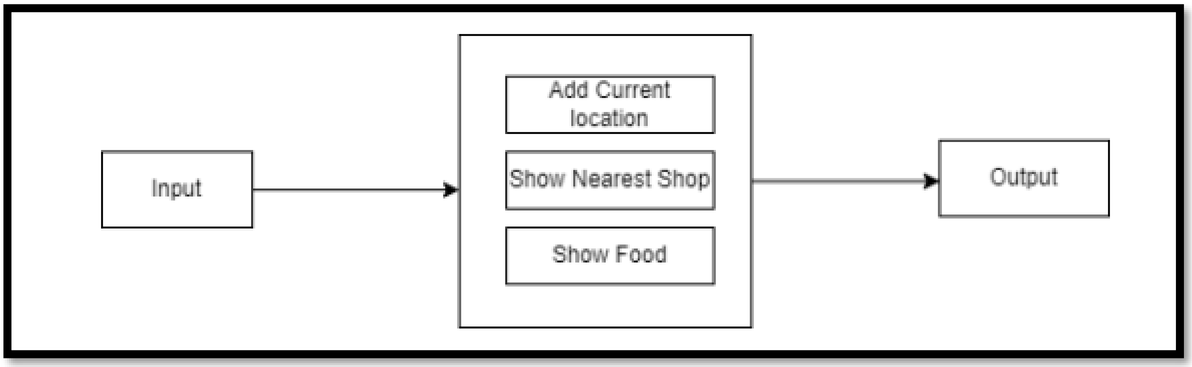

Fig 2.2.2: DFD - level 1

Data Flow Diagram – Level 2

DFD level 2 goes one more step deeper into the subprocesses of Level 1. Shows the DFD level 2 of the system. It might require more text to get into the necessary level of detail about the functioning of the system. The Level 2 gives a more detailed sight of the system by categorizing the processes involved in the system to three categories namely preprocessing, feature scaling and classification. It also graphically depicts each of these categories in detail and gives a complete idea of how the system works.

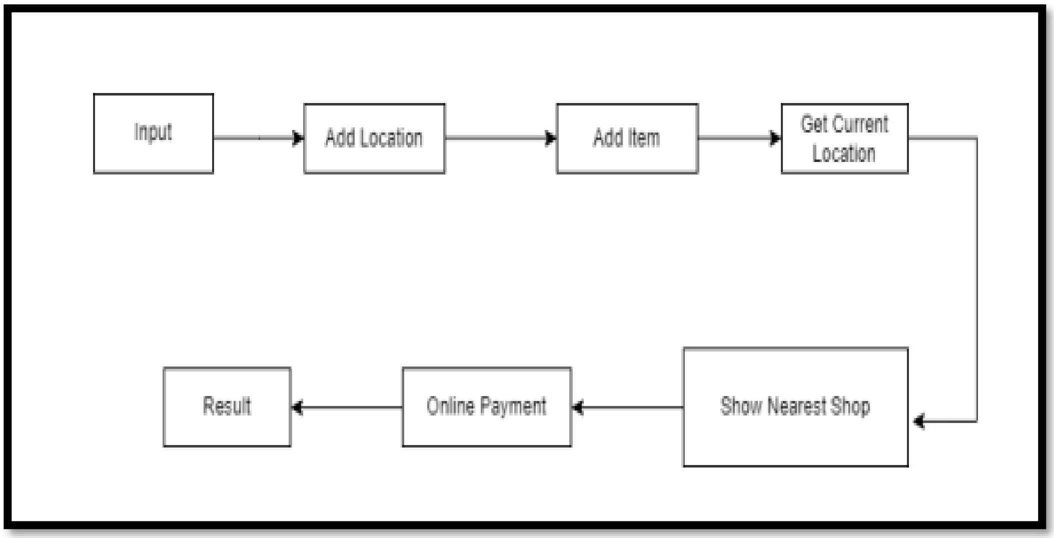

Fig 2.2.3: DFD - level 2

2.3 User Classes and Characteristics

User classes and their characteristics for a restaurant recommendation system can help define the target audience and tailor the system's features to meet the diverse needs and preferences of different users. Here are some user classes and their characteristics:

General Diners:

- Characteristics: These users are looking for dining options based on their general preferences and cravings. They may prioritize factors like cuisine type, price range, and location.
- Needs: Recommendations for everyday dining, discovering new local restaurants, and occasional special dining experiences.

Food Enthusiasts:

- Characteristics: These users are passionate about food and are interested in exploring unique and authentic culinary experiences. They may have specific preferences for gourmet or ethnic cuisines.
- Needs: Recommendations for niche or specialty restaurants, access to detailed menu information, and the ability to discover hidden gems.

Vegetarians and Vegans:

- Characteristics: Users with dietary restrictions or preferences, such as vegetarians and vegans, require recommendations that align with their dietary choices.
- Needs: Recommendations for restaurants that offer vegetarian or vegan options, including clear labeling and menu information.

Health-Conscious Diners:

- Characteristics: Health-conscious users prioritize nutritious and low-calorie options. They may look for restaurants that offer healthy choices, salads, or specific diet plans.
- Needs: Recommendations for restaurants with healthy menu items, nutritional information, and calorie counts.

Budget-Conscious Diners:

- Characteristics: Users on a budget seek dining options that fit within their financial constraints. They may prioritize affordability and discounts.
- Needs: Recommendations for budget-friendly restaurants, daily specials, and access to deals and promotions.

2.4 Operating Environment

Operating Environment defines the A technical overview of our service specifications, together with the way the sub-components inter-operate.

Software

- Operating Systems: Windows 11, Windows Server 2012 or greater
- Python 3.1 or above
- Python libraries NumPy 3.9, Pandas3.9, Skit-learn1.2.2
- Flask framework 2.2.3,

Hardware

- Processor: 1 GHz or greater
- Hard Disk Space*: 16GB available disk space
- Memory: 2 GB (32-bit), 4 GB (64-bit)
- Graphics: Support for DirectX 9 graphics with minimum 128MB RAM. 1.6.2

2.5 Design and Implementation Constraints

Designing and implementing a restaurant recommendation system can be a complex endeavor with various constraints to consider. These constraints can include technical, ethical, legal, and practical limitations. Here are some common design and implementation constraints for a restaurant recommendation system:

Data Availability: Availability and quality of restaurant data, including restaurant details, user reviews, and menu information, can be limited and may require data scraping, data cleaning, or partnerships with third-party data providers.

Scalability: As the user base and restaurant database grow, the system must be scalable to handle increased traffic and data without compromising performance.

Data Privacy and Security: The system must comply with data privacy regulations (e.g., GDPR) and ensure the security of user information and preferences.

Algorithmic Complexity: Developing and maintaining recommendation algorithms can be computationally intensive, and optimizing these algorithms for real-time recommendations can be challenging.

User Engagement: Encouraging users to actively use the recommendation system and contribute reviews and ratings can be difficult and requires thoughtful design and incentives.

Cold Start Problem: Recommending restaurants to new users with limited interaction history (the cold start problem) can be challenging. The system must employ strategies to address this issue.

2.6 User Documentation

User documentation for a restaurant recommendation system is crucial to help users understand how to use the system effectively and make the most of its features. Here's a basic outline of what user documentation for a restaurant recommendation system could include:

Introduction: Briefly explain the purpose and benefits of the restaurant recommendation system.
Mention the target audience for the documentation.

Getting Started: Provide instructions for creating a user account if applicable.
Explain how to access the system, whether through a website, mobile app, or other platforms.

User Profile: Describe how users can complete and edit their profiles.
Highlight the importance of accurate profile information for personalized recommendations.

Search and Browse: Explain how to search for restaurants based on criteria such as cuisine type, location, price range, and dietary preferences.

Provide guidance on using filters and sorting options.

Restaurant Recommendations: Describe how the recommendation algorithm works, considering factors like user preferences and location.
Explain how to view and interact with recommended restaurants.

Interacting with Restaurants: Show users how to view restaurant details, including photos, menus, and user reviews.
Explain how to rate and review restaurants or add them to favorites/bookmarks.

Location Services: If the system uses location-based services, explain how users can enable or disable them.
Provide guidance on how location is used for recommendations and directions to nearby restaurants.

User Support: Provide contact information for customer support or a help center.
Explain how users can report issues or provide feedback.

Privacy and Security: Highlight the system's data privacy and security measures.
Explain how user data is protected and how user information is used.

FAQs (Frequently Asked Questions): Compile a list of frequently asked questions and their answers to address common user queries.

2.7 Assumptions and Dependencies

When designing and developing a restaurant recommendation system, it's essential to make certain assumptions and identify dependencies to ensure a clear understanding of what the system requires and how it operates. Here are some typical assumptions and dependencies for a restaurant recommendation system:

Assumptions:

- User Data Accuracy: The system assumes that user-provided information, such as location, dietary preferences, and restaurant ratings, is accurate and up-to-date.

- Internet Connectivity: Users are assumed to have reliable internet access to use the recommendation system, whether through a website or mobile app.

- Location Accuracy: Location-based recommendations assume that the user's device provides accurate location data, and the system can access this data.

- Data Privacy and Consent: Users are assumed to provide informed consent for the collection and use of their data for recommendations, adhering to data privacy regulations.

- Restaurant Data Availability: The system assumes access to a comprehensive and regularly updated database of restaurant information, including details like menus, prices, and operating hours.

Dependencies:

- Data Sources: The system depends on access to reliable data sources for restaurant information, including APIs, web scraping, or partnerships with data providers.

- Location Services: For location-based recommendations, the system relies on access to the user's location data through GPS or other means.

- User Engagement: The quality of recommendations depends on user engagement and their willingness to provide feedback, ratings, and reviews.

- Algorithm Development: The recommendation algorithms are a critical dependency, and their development and optimization may require significant time and resources.

- Privacy and Security Measures: The system depends on robust privacy and security measures to protect user data and comply with legal requirements.

3. External Interface Requirements

3.1 User Interfaces

Designing user interfaces for a restaurant recommendation system is crucial to ensure an intuitive and engaging user experience. Depending on the platform (web, mobile app, or other), the design may vary, but the key principles of usability and aesthetics remain consistent. Here are some essential user interfaces (UI) and components to consider:

Homepage: The homepage serves as the entry point for users. It typically includes a search bar, recommendations, and navigation options.
Features:

- Search bar for location or cuisine-based searches.
- Eye-catching visuals showcasing top-rated restaurants.
- Quick links to popular cuisines or trending restaurants.
- Navigation menu for accessing various sections of the app.

Search Results Page: This page displays search results based on user input, filters, or recommendations.
Features:

- List or grid view of restaurants with images, names, ratings, and brief descriptions.
- Filters for refining results (e.g., cuisine type, price range, dietary preferences).
- Sorting options (e.g., by rating, distance, or price).
- Map view with restaurant markers for location-based searches.

Restaurant Details Page: When a user selects a restaurant, they are directed to a detailed page with comprehensive information.
Features:

- High-quality images of the restaurant, including the interior and dishes.
- Restaurant name, address, contact details, and operating hours.
- Detailed menu with prices, descriptions, and dietary labels.
- User-generated reviews and ratings.
- Reservation or booking options if available.

3.2 Hardware Interfaces

The restaurant recommendation system does not require any specific hardware interfaces, but it should be designed to be compatible with a variety of hardware components, including desktop and laptop computers, mobile devices, web browsers, and internet connections. This ensures that users can access and use the website on different platforms and devices without any issues.

3.3 Software Interfaces

The restaurant recommendation system needs to interact with various software components to provide its functionality, including web servers, databases, email clients, and APIs. It should be designed to be compatible with popular software components to ensure seamless integration and optimal performance.

3.4 Communications Interfaces

The restaurant recommendation system uses various communication interfaces to exchange data and information with external systems and services, such as HTTP/HTTPS, SMTP, APIs, FTP/SFTP, and WebSocket. These interfaces should be designed to ensure secure and reliable communication and optimal performance.

4. System Features

A restaurant recommendation system can offer a wide range of features to enhance user experiences and provide valuable dining suggestions. Here are some key system features for a restaurant recommendation system:

- Search and Filters:
- Recommendation Engine:
- Restaurant Listings:
- User Reviews and Ratings:
- Menu Access:
- Location Services:

5. Other Nonfunctional Requirements

5.1 Performance Requirements

Performance requirements for a restaurant recommendation system are critical to ensure that the system operates efficiently, responds quickly, and provides a seamless user experience. These requirements help set benchmarks for system performance and ensure that it can handle various aspects of user interactions and data processing. Here are some performance requirements for a restaurant recommendation system:

Response Time: The system should respond to user queries, searches, and interactions within a reasonable time frame. For example, recommendations should be generated and displayed to users within a few seconds.

Scalability: The system should be scalable to accommodate increased user traffic and data as the user base grows. It should handle higher loads during peak usage without a significant decrease in performance.

Concurrent User Support: Specify the number of concurrent users the system should be able to handle without performance degradation. This requirement is particularly important for systems with a large user base.

Database Performance: Define the maximum acceptable response times for database queries, especially when retrieving restaurant information, reviews, and user data.

Recommendation Generation Time: Set a maximum time limit for generating personalized restaurant recommendations. Users should receive recommendations promptly, even if they have complex preferences.

Search and Filtering Speed: Specify the acceptable response times for restaurant search queries and filtering options. Users should experience minimal delays when applying filters or conducting searches.

Location-Based Services: Ensure that location-based services, including real-time location detection and mapping features, operate with minimal latency.

5.2 Safety Requirements

Safety requirements for a restaurant recommendation system are essential to ensure the protection of user data, maintain system reliability, and address potential risks or vulnerabilities. Here are some safety requirements for such a system:

Data Privacy and Protection: Implement robust data protection measures to safeguard user information, including personal data, location data, and dining preferences.
Comply with relevant data privacy regulations (e.g., GDPR) and clearly communicate the system's data handling practices to users.

User Authentication and Authorization: Enforce secure user authentication mechanisms to prevent unauthorized access to user accounts and data.
Implement role-based access control to ensure that only authorized personnel can modify critical system settings.

Secure Data Transmission: Use encryption protocols (e.g., HTTPS) to secure data transmission between users' devices and the system's servers.
Ensure that sensitive user data is not exposed during data exchange.

Account Security: Require strong and unique passwords for user accounts and implement mechanisms for password recovery and account lockout after multiple failed login attempts.
Encourage two-factor authentication (2FA) for enhanced security.

Secure APIs: If the system offers APIs for third-party integration, secure those APIs with authentication and authorization mechanisms.
Implement rate limiting and access controls to prevent abuse.

5.3 Security Requirements

Security is a critical aspect of a restaurant recommendation system to protect user data, prevent unauthorized access, and ensure system integrity. Here are essential security requirements for such a system:

User Authentication:
Strong Password Policies: Enforce password complexity requirements, and encourage users to create strong, unique passwords.
Two-Factor Authentication (2FA): Offer 2FA as an option to enhance account security.

User Authorization:
Implement role-based access control (RBAC) to ensure that users have appropriate permissions based on their roles within the system.

Data Encryption:
Use encryption protocols (e.g., HTTPS/TLS) to secure data transmission between users' devices and the system's servers.
Encrypt sensitive data at rest, including user credentials and sensitive user information in the database.

Secure APIs:
Secure all APIs with proper authentication and authorization mechanisms.
Implement rate limiting and access controls to prevent abuse of APIs.

Input Validation:
Employ input validation techniques to prevent common vulnerabilities such as SQL injection, cross-site scripting (XSS), and cross-site request forgery (CSRF).

5.4 Software Quality Attributes

Software quality attributes, also known as non-functional requirements, are essential for ensuring the overall quality and performance of a restaurant recommendation system. These attributes describe how well the system performs its functions and how it behaves in various situations. Here are key software quality attributes for such a system:

Usability:

- User-Friendly Interface: Ensure that the user interface is intuitive, easy to navigate, and visually appealing.
- Efficient Workflow: Optimize user workflows to minimize user effort and clicks required to achieve tasks.
- Accessibility: Make the system accessible to users with disabilities, adhering to accessibility standards (e.g., WCAG).

Performance:

- Responsiveness: The system should respond promptly to user interactions, queries, and requests.
- Scalability: Ensure that the system can handle increased user loads and data volume without significant performance degradation.
- Throughput: Measure and optimize the number of transactions or requests the system can handle per unit of time.

Reliability:

- Availability: The system should be available and operational most of the time, with minimal downtime for maintenance or unexpected failures.
- Fault Tolerance: Implement mechanisms to recover gracefully from failures and errors, ensuring uninterrupted service.
- Data Integrity: Safeguard user data against loss, corruption, or unauthorized access.

Security:

- Data Protection: Secure user data, both in transit and at rest, using encryption and access controls.
- Authentication and Authorization: Ensure proper user authentication and authorization mechanisms are in place.
- Vulnerability Mitigation: Regularly scan for and address security vulnerabilities, following secure coding practices.

Scalability:

- Horizontal and Vertical Scalability: Design the system to scale horizontally (adding more servers) and vertically (upgrading hardware) as needed to handle increased loads.

Maintainability:

- Modularity: Use a modular architecture that allows for easy updates, enhancements, and maintenance of individual components.
- Code Quality: Maintain clean, well-documented, and well-structured code to facilitate ongoing development.
- Version Control: Use version control systems to manage code changes and collaborate effectively.

6. Other Requirements

To develop a restaurant recommendation system other requirement are:

Integration with POS Systems: If the system offers reservations or online ordering, it should seamlessly integrate with restaurant Point of Sale (POS) systems for order processing.

Geofencing and Location Services: Implement geofencing features to notify users of nearby restaurants and promotions when they enter specific geographic areas.

Offline Access: Offer limited offline access to allow users to view saved restaurant information and recommendations when they have no internet connection.

Multi-Language Support: Provide multilingual support to cater to a diverse user base, including localized content and translations.

Dietary and Allergen Information: Include comprehensive dietary and allergen information for dishes on restaurant menus to accommodate users with specific dietary requirements.

Appendix A: Glossary

To properly interpret an SRS (Software Requirements Specification), the following terms, acronyms, and abbreviations may be necessary:

SRS: Software Requirements Specification. It is a document that describes the functional and non-functional requirements of a software system.

Functional requirements: These are the requirements that describe what the software system should do. They specify the behavior and functionality of the system.\

Non-functional requirements: These are the requirements that describe how the software system should behave. They specify the performance, usability, reliability, and other quality attributes of the system.

User interface (UI): The user interface is the part of the software system that the user interacts with. It includes all the screens, buttons, forms, and other elements that the user can see and interact with.

API: An Application Programming Interface is a set of protocols, routines, and tools for building software applications. APIs specify how software components should interact with each other.

Database: A database is a structured collection of data that is organized and stored in a computer system. It is used to store and manage information for the software system.

GUI: Graphical User Interface is a type of user interface that allows users to interact with the software system using graphical elements such as icons, buttons, and menus.

Entity: An entity is a data object in a database. It represents a real-world object, such as a customer or an order.

Attribute: An attribute is a characteristic or property of an entity. It represents a specific piece of data about the entity.

Module: A module is a self-contained unit of code that performs a specific function or set of functions.

ERD: Entity-Relationship Diagram is a graphical representation of entities and their relationships to each other in a database.

Appendix B: Analysis Models

Data Flow Diagram

DFDs are used to depict graphically the data flow in a system [43]. It explains the processes involved in a system from the input to the report generation. It shows all possible paths from one entity to another of aa system. The detail of a data flow diagram can be represented in three different levels that are numbered 0, 1 and 2.
There are many types of notations to draw a data flow diagram among which YourdonCoad and Gane-Sarson method are popular. The DFDs depicted in this chapter uses the Gane-Sarson DFD notations.

Appendix B.1: Data Flow Diagram – Level 0

DFD level 0 is called a Context Diagram. It is a simple overview of the whole system being modeled. Fig 4.2 shows the DFD level 0 of the system.

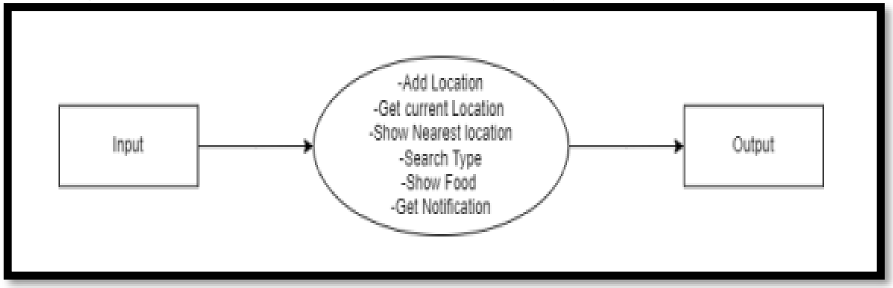

Figure Appendix B.2: DFD level 0

Appendix B.2: Data Flow Diagram – Level 1

DFD level 1 gives a more detailed explanation of the Context diagram. The high-level process of the Context diagram is broken down into its subprocesses. The DFD level 1of the system is depicted

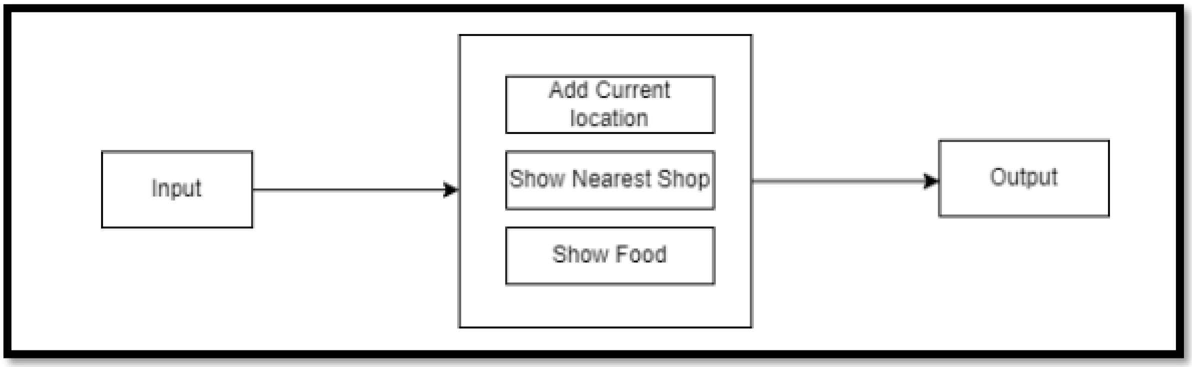

Figure Appendix B.2: DFD - level 1

Appendix B.3: Data Flow Diagram – Level 2

DFD level 2 goes one more step deeper into the subprocesses of Level 1. Shows the DFD level 2 of the system. It might require more text to get into the necessary level of detail about the functioning of the system. The Level 2 gives a more detailed sight of the system by categorizing the processes involved in the system to three categories namely preprocessing, feature scaling and classification. It also graphically depicts each of these categories in detail and gives a complete idea of how the system works.

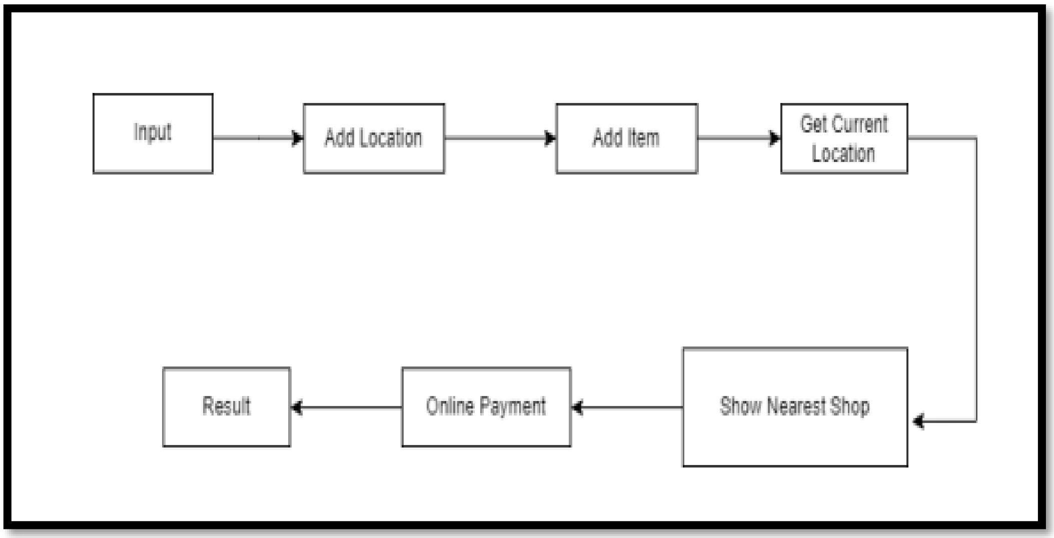

Figure Appendix B.3: DFD - level 2

Appendix B.4: Class Diagram

A class diagram is a type of UML (Unified Modeling Language) diagram that represents the static view of a system, i.e., the system's structure, classes, and their relationships. In software engineering, a class diagram is used to describe the classes and objects in a system and their relationships to one another. It illustrates the attributes, methods, and relationships of classes and their instances. Class diagrams help to visualize and document the design of a system, making it easier for developers to understand and modify the system. They are widely used in Object-Oriented Programming (OOP) and other software development methodologies.

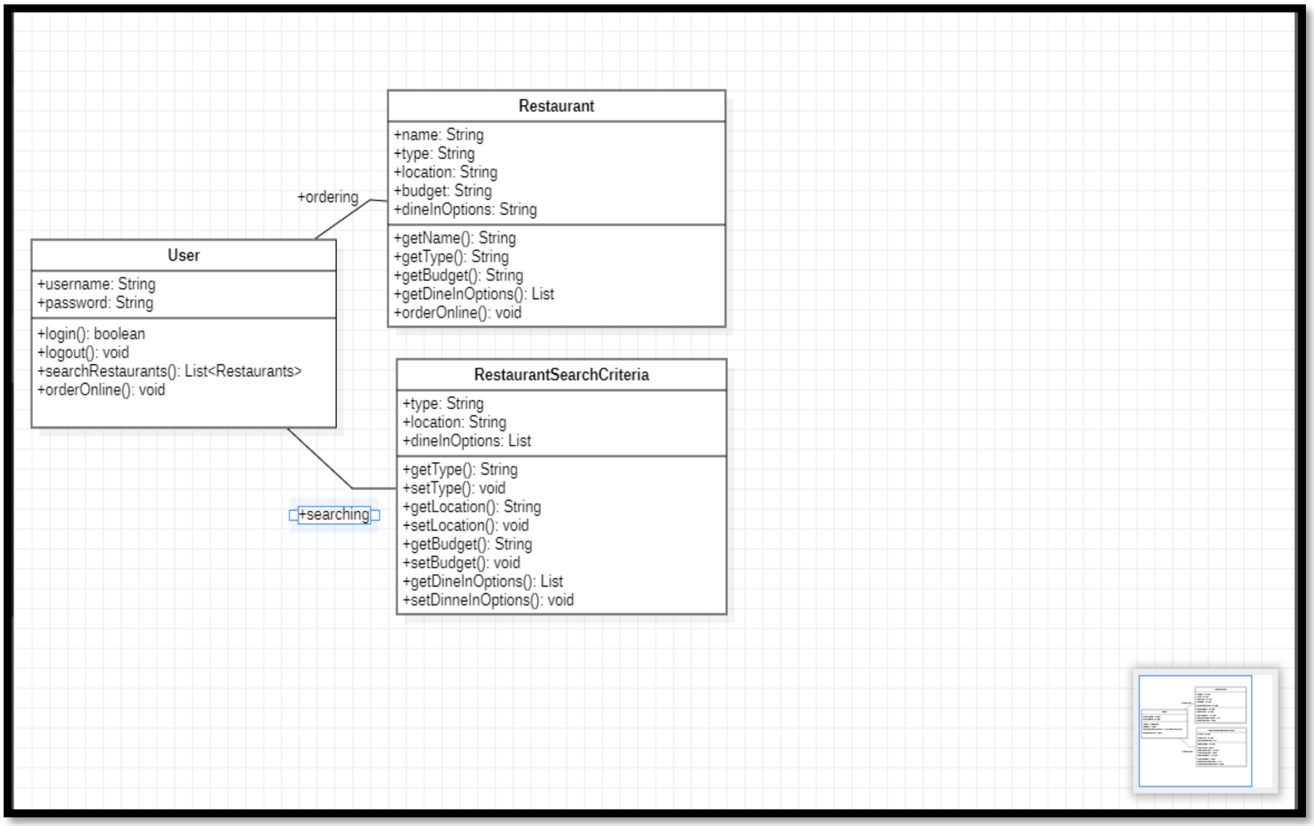

Figure Appendix B.4: Class Diagram

Appendix B.5: Entity Relationship Diagram

An ER diagram (Entity-Relationship diagram) is a type of diagram used in software engineering and database design to represent entities, attributes, and relationships between them. An ER diagram represents a conceptual view of the database and helps to visualize the data schema in a structured format. It consists of three main components: entities, attributes, and relationships. Entities are objects or concepts that have distinct attributes, and relationships are the associations between the entities. Attributes are the properties or characteristics of the entities. ER diagrams are useful in database design and can help developers to understand the data requirements of a system and design a database schema accordingly. They are also used to communicate the database design to stakeholders, including clients and other team members.

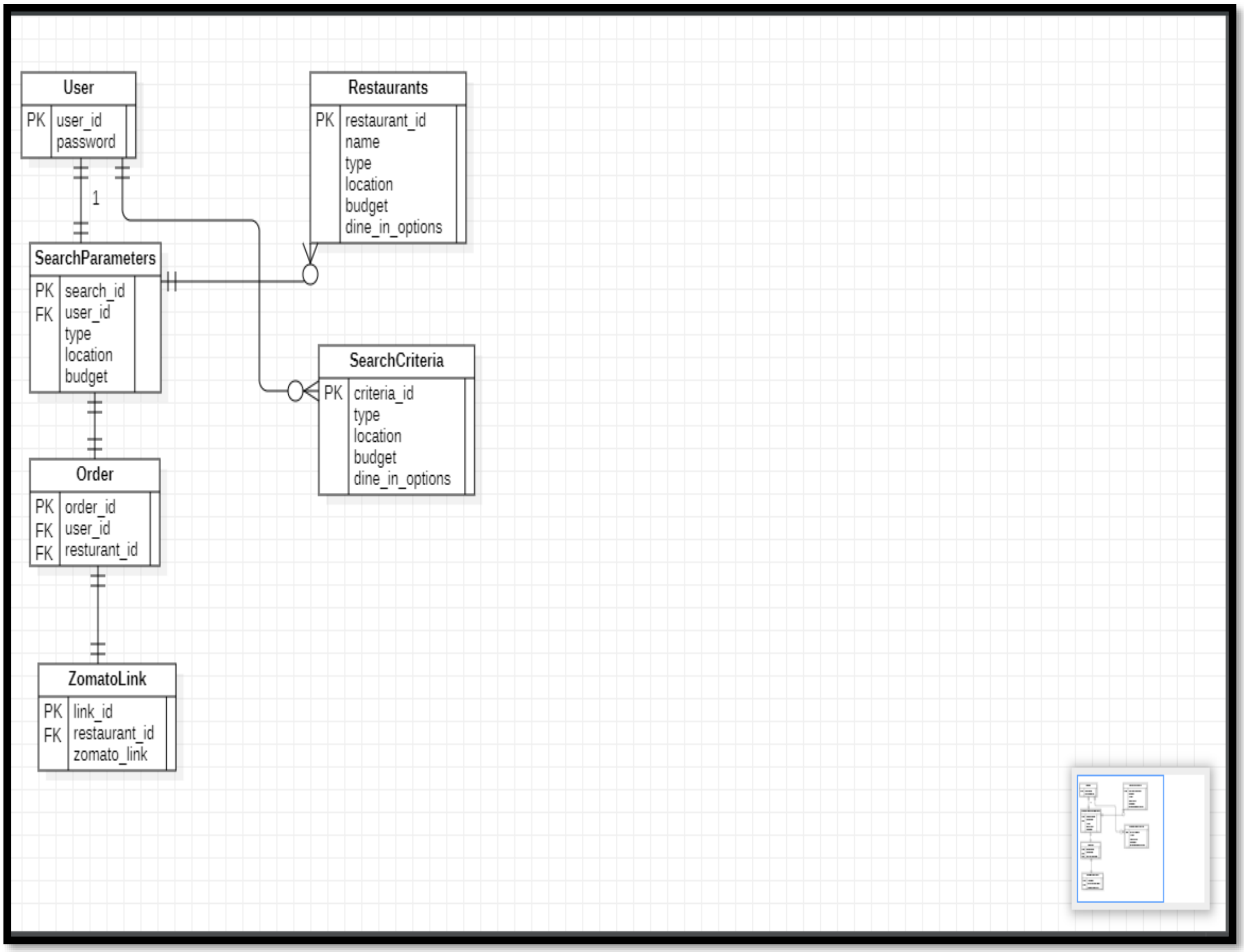

Figure Appendix B.5: ER Diagram

Appendix C: To Be Determined List

Items we can consider including in a "to be determined" list for a restaurant recommendation system:

Advanced AI and Machine Learning Techniques: Specify which advanced AI and ML techniques will be used for recommendation and how they will be integrated.

User-Generated Content Policies: Define detailed policies for user-generated content, including moderation procedures and guidelines.

Marketing and User Acquisition Strategies: Determine the specific marketing channels, strategies, and budgets for user acquisition and retention.

Data Storage and Scalability Architecture: Decide on the database technology, data storage solutions, and architectural choices for scaling the system as it grows.

Monetization Models: Outline the detailed monetization strategies, pricing models, and revenue generation mechanisms.

Mobile App Platforms: Specify whether the system will have dedicated mobile apps for different platforms (iOS, Android) and the development roadmap.

Legal and Compliance Considerations: Identify and address legal and compliance requirements specific to the restaurant recommendation industry, including licenses and permits.

User Onboarding and Tutorials: Plan the onboarding process for new users, including tutorials, tooltips, and guided experiences.