University Of Mauritius

Faculty of Information, Communication and Digital Technologies

SOFTWARE REQUIREMENTS SPECIFICATION

FOR

FOOD DELIVERY MANAGEMENT SYSTEM

By team : QuickBite

Prepared by :

SOOBRAYEN Keshia 2412920

BEEDASSY Nirvana Luxmi 2413850

RAGHOONUNDUN Rishab 2412024

AUCKEL Ayush Manav 2412549

THUMMANAH Kentish 2411831

Course : BSc (Hons) Computer Science

Module : ICT 1208Y Software Engineering Principles

Lecturer : Dr. Mungur Avinash Utam

Date : 29th of September 2024

Table of Contents

Acknowledgement

In as much as it is our tradition that work be shared equally among team members this particular project was no exception. We stayed in touch with consistent video calls and in-person meetings at the library, and tried to keep working on the assignment continuously, so that we were able to proceed with the collaboration.

1. Introduction

1.1 Aims and Objectives

The **Food Delivery Management System** attempts to offer a holistic solution for being able to place an order for food, getting it delivered, and even managing restaurants. It shall enable customers to browse menus from several restaurants and order for meals with payments made through a simple command interface. This system shall enable restaurants to handle orders well and in a timely manner and also coordinate with the delivery people in real-time. Furthermore, delivering agents and the users themselves get updates of the real time tracking, making delivery processes more transparent to the customers. Finally, both users and restaurants are benefitted with efficient and integrated processes throughout the application.

The objectives are as follows:

1. Customer Interface: The platform shall have an interface that the customer shall perform a search of restaurants and menus as well as place orders effectively. The user interface shall also be comprehensive and integrate secure payments methods including; credit emergency payment, debit card and e-wallets for easy check out. Such things as previous orders and preferences regarding the items put forward also shall be treated as filters to be used in order to enable a fast navigation to interesting items.

2. Efficient Order Processing: Processing of orders within restaurants shall be effective according to the said system. It shall afford an opportunity to alert users when a new order has just been placed. This shall also allow restaurants to keep track of the menu, change the status and confirm orders from a simple restaurant managing system. It also shall facilitate coordination with the kitchen and delivery in order to ensure that food is prepared and dispatched well.

3. Delivery Operations: The core of the whole system is the control of the delivery services. Customers shall receive an e-notification of delivery times and the system shall also avail for route optimization for delivery agents and assign for delivery agents. The delivery agents shall be provided with a means by which they shall receive delivery request, update delivery status for instance Picked up or delivered and navigate to customer location.

4. Real-time Tracking and Notifications: Genres shall be able to monitor the status of their order from preparation stage to dispatch stage and the delivery stage. The user shall receive notifications at every major step: a receipt of something such as an order placed, picked or on its way to being delivered. Delivery agents shall receive a notification concerning new delivery order or an update on the status of existing orders.

It not only helps to improve users' satisfaction, as it provides the transparency of data and maintenance of actual time information, but it also contributes to the restaurants' and delivery agents' operational effectiveness, guaranteeing effective interaction of all the counterparts engaged in the process.

1.2 SRS Team members and assigned Subsystem

Customer Ordering Subsystem	Beedassy Nirvana Luxmi 2413850
Restaurant Management Subsystem	Soobrayen Keshia 2412920
Delivery Management Subsystem	Raghoonundun Rishab 2412024
Payment Gateway Subsystem	Thummanah Kentish 2411831
Admin Dashboard Subsystem	Auckel Ayush Manav 2412549

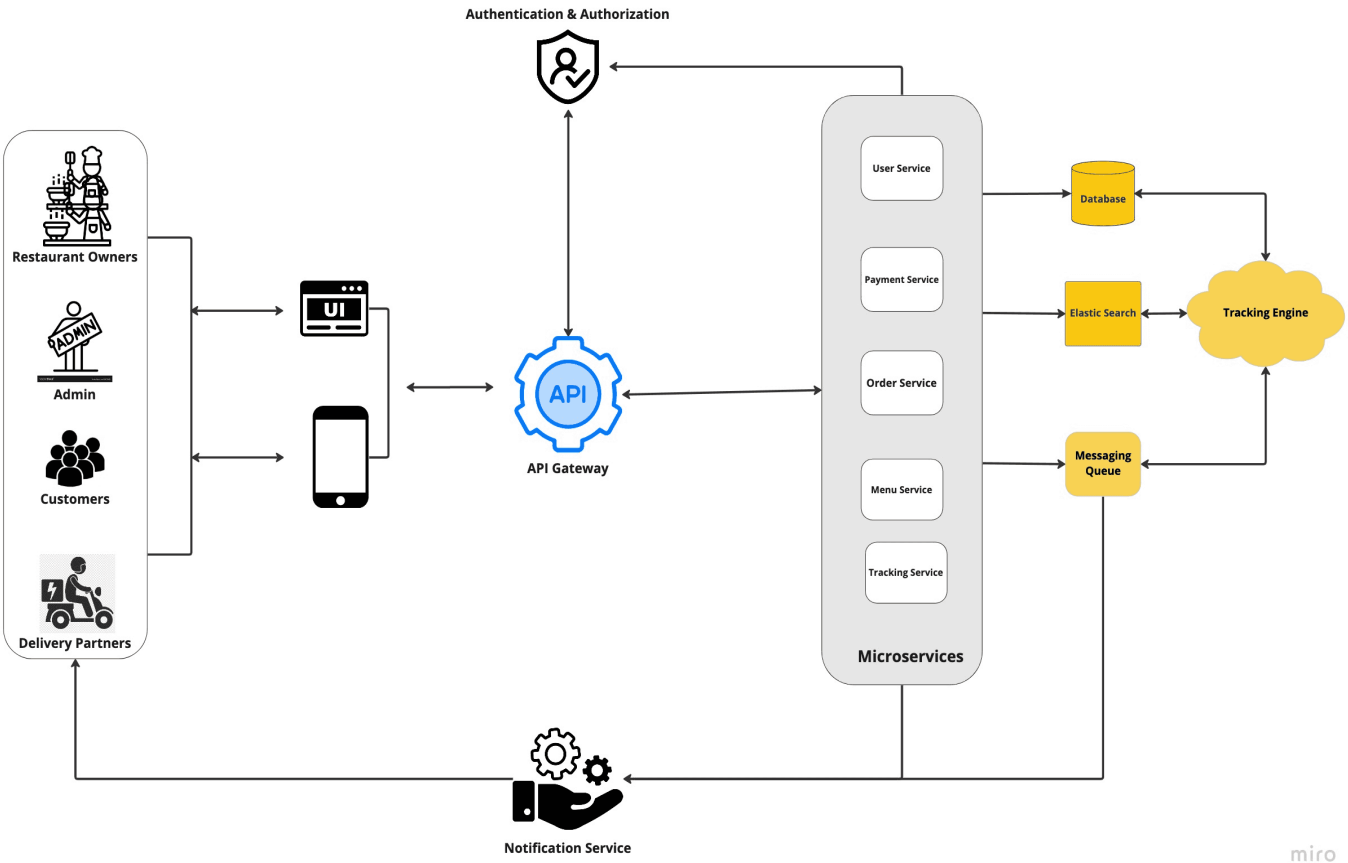
1.3 Recommended Software Process Model

The **Agile Development Process** is highly feasible when it comes to the development of the Food Delivery Management System is the flexibility that the system offers as well as the main concept of the process – iterative development. In Agile, much of the use is done in cycles of development and this means that during the detail cycles the feedback from the customers, restaurants and delivery agents shall be incorporated in real time. This makes it possible for the system to grow with changes in the needs of a new feature, change in design or improvement in performance. Agile also allows agile approaches to tendencies that shall be seen potentially problematic, to adjust, ensuring the next iteration leaves a system more refined and sophisticated to meet users' likes and the market's demands.

2. Overall Description

2.1 System Architecture Diagram

This diagram here shall show the relationships between subsystems like Customer Interface, Restaurant Management, Delivery Management, Payment Processing, and Admin Dashboard.



Admin Panel: Responsible for onboarding of Restaurants Owners and Delivery Partners.

Restaurant Owners: Responsible for adding menus and managing orders and payments.

Delivery Partners: Responsible for picking an order from the Restaurant Owners and delivering them to the customers.

Customers: They are the actual client. They shall be consuming the application. The customer shall make an order from the system.

2.2 Subsystem Description

Customer Ordering Subsystem

This subsystem enables the users to search for any type of restaurants and makes available for perusal. When it comes to the offered restaurants, clients shall navigate through their menus, and see descriptions of certain items as well as their prices. Customers shall select items, specify requirements like meat preference or even put their orders on hold for delivery or pickup. After orders, they shall be able to monitor the status of their order in real-time, right from confirmation to delivery.

Restaurant Management Subsystem

This shall be used by the restaurants in managing their profile, changing menus, as a sign of availability of a particular item, and to command orders. It also offers the openness to make contact with the delivery people to ensure the order is being compiled for delivery. Restaurants shall also control offers, timings, and response to customers.

Delivery Management Subsystem

It organizes the task of placing the agents that bring about timely delivery through a subsystem. Real-time tracking of location enables it to track delivery and to rearrange within the delivery to consider factors like traffic congestion, nearness etc. Delivery agents receive information about the new assignments or the status of delivery and acquire customer location for easy navigation.

Payment Gateway Subsystem

The Payment Gateway subsystem enables secure payments, therefore, users shall pay via credit/debit cards, digital wallets and bank transfers. It shall include dependable third party payment systems which are capable of performing secure third party transactions concerning the client’s details. This shall also process refund and raise invoices.

Admin Dashboard Subsystem

The Admin Dashboard has full control on the entire system. It shall manage user accounts (customers, restaurants, delivery agents), resolve disputes, and create detailed reports on the performance of the system, transactions, history, and user activity. The dashboard also provides tools to check system health, manage escalations, and perform checks on general system security and compliance.

3. User and System Requirements

3.1 Functional Requirements

3.1.1 Customer Ordering Subsystem

The **Customer Ordering Subsystem** uses a mobile app which provides users with an intuitive and efficient way to browse, select, and order food from a variety of restaurants.

The subsystem shall be able to:

1. Browse restaurants based on location

Consumers shall be required to scroll through a selected list of restaurants in their region, utilising GPS to display restaurants close to a given user. Some filters include the cuisine type; ratings; delivery time and so on help with browsing. What this does is that users shall browse the various menus they find most appropriate depending on the geographical location.

2. View detailed menus

The users shall be shown the menu of the selected restaurant and be provided with the following options; names of the foods, descriptions, prices, photos of foods, portion sizes, options to add or remove things like cheese, vegetarian or non-vegetarian meals. Users shall also be able to read reviews on some meals which shall help them make a decision.

3. Order Customization and Placement

It also encompasses provisions for change in an order by including some ingredients, deleting them, changing portion sizes, and combination prices. Upon the completion of the items' placements in the cart, it shall enable users to check on the order to make some adjustments before payment. The allowed modes of payment are Credit/ Debit card, Wallets and COD.

4. Order History and Re-ordering

This feature shall be more convenient for the user since customers get the access to the list of their orders placed before. Such details consist of name of the restaurant, date, menu, quantity and total price. Customers are able to choose previous orders – single items or full orders – with a single tap, eliminating the need to look for them again in the menu. To the frequent users, the function alleviates the burden, enhances

ordering efficiency and overall customer satisfaction particularly to the most frequent order takers.

3.1.2 Restaurant Management Subsystem

The **Restaurant Management Subsystem** assists restaurants with keeping the orders, their menus, and overall management via the platform. It ensures uninterrupted contact with clients and organizes workflow processes to the highest performance, satisfaction of the customers, and growing the business.

The subsystem shall be able to:

1. Menu Management

Restaurants shall be able to create, update and manage their menus in the system which includes creating categories & items along with their descriptions, pricing and dietary tags (vegetarian or gluten-free options). There shall be the option to include special offers, allow for discounts and display a star dish. Moreover, the restaurants shall feature menu items that automatically update in real time and become unavailable on a customer-facing online order form whenever the kitchen runs out of a product, so customers only see what shall be ordered.

2. Order Management and Processing

Restaurants shall get new order notification and order acceptance/rejection based on restaurant's capacity and availability. They shall be provided with an in depth breakdown of each order- what the customer has ordered and any adaptations to that specification to help with preparation. Also, restaurants shall update the order status (“preparing,” “ready for pickup,” “delivered”), so the customer knows in real time what is happening. Further, the system shall keep a record of all the orders placed and processed for quick resolution shall there be customer complaints in the future.

3. Inventory Tracking & Management

The system shall monitor key ingredient stock levels of low or out of ingredients and notify managers to avoid longer waiting times. It shall enable the restaurant to trigger automatic reorders from vendors when the stock falls below a certain level. On top of that, menu items that rely on out-of-stock ingredients shall be automatically flagged as "unavailable" in real-time so customers shall not have the bad experience of getting excited about a certain plate and finding out it is actually off the table.

4. Performance & Feedback analytics

Restaurants shall be able to see detailed sales reports, broken down into revenue, daily order counts, top items sold and busiest times. It collects customer reviews and feedback at an order level, enabling management to view a summary for service improvement purposes. It also offers insights into customer favourites, allowing restaurants to determine which items shall be publicized or changed based on feedback or popularity. Further, the reports about delivery time and food score shall help restaurants improve their operational efficiency.

3.1.3 Delivery Management Subsystem

A **Delivery Management subsystem** shall efficiently manage the deliveries, optimize route building, driver assignment, and real-time updates. This shall handle orders, automate dispatching, customer notifications, and collect proof of delivery. The aim is to speed up the delivery process and reduce the cost by ensuring that customers stay satisfied with timely and correct information.

The subsystem shall be able to:

1. Route Optimisation

A route optimization in a delivery management subsystem shall be capable of effective route planning and adjustment to minimize the travel time and, hence, the cost. It shall include factors like traffic flow, time slots in deliveries, weather condition, and ensure that drivers follow the fastest and most fuel-efficient paths . The system shall also adapt routes in real-time when unexpected events like road closures or accidents occur. This shall also group the deliveries logically to reduce the number of trips. This shall offer improved fleet utilization and on-time delivery.

2. Customer Notification

Customer notifications shall keep recipients informed throughout the delivery process, enhancing their experience and building trust. This shall be automated through SMS, email, or in-app notifications about the critical stages of the process-order confirmation, dispatch, and estimated time of arrival. Also, notifications customers to any changes, such as delays or route adjustments, shall be issued along with a reschedule delivery service wherever necessary. Information given in a timely and relevant manner to customers enhances the system by providing better transparency

and reducing anxiety, thus encouraging positive engagements that lead to better customer satisfaction.

3. Driver Management and Assignment

The system shall lighten the task of drivers' delivery assignments by taking into consideration factors of proximity, vehicle capacity, and driver availability. It shall allow the dispatcher to assign deliveries efficiently in real time, maintaining workload balance and utilizing drivers optimally. Besides that, it shall monitor driver performance, record delivery times, and provide feedback for improvements based on the same for better efficiency. The system shall also enable communication between the drivers and dispatchers for updating or changing any delivery plan with speed. This capability not only enhances operational efficiencies but also supports route management for drivers to allow for better on-time deliveries and increased customer satisfaction.

4. Proof of Delivery (POD)

Proof of delivery in any delivery management system is verification that an item has reached the intended recipient. This sub-system shall collect evidence like a digital signature, photo, or GPS location upon completion. This helps in making people accountable for verifying the time, place, and person delivering the service to reduce fraud or disputes. Secondly, POD data provides lots of evidence when it comes to record keeping, which shall be useful when resolving some customer inquiries or claims about something either not delivered or damaged. It enhances transparency and trust between business and customers.

3.1.4 Payment Gateway Subsystem

The **Payment Gateway Subsystem** is really important for handling payments safely in the Food Delivery Management System. It allows customers to pay in different ways while keeping their personal details secure. This part of the system makes payments out easy and helps follow security protocols, which improves the overall experience for users.

The system shall be able to :

1. Payment Processing

The system shall allow users to pay via different types of payment methods such as credit/debit card, digit wallets or bank transfers. This allows user to put their banking or card details to complete orders and transactions.

2.Transaction Security

The system shall securely process all transactions to protect sensitive user data/information. All payment information shall be encrypted using industry-standard protocols(E.g Transport Layer Security/Secure Socket Layer) during transactions to prevent unauthorized access.

3. Refund Processing

The system shall allow for the process of refunds for cancelled orders or payment errors, ensuring that users receive their funds back. Users shall be given access to a simple form to submit refund requests, including reasons for cancellation or error. System shall reverse the transactions made by user back to original payment method.

4.Transaction History Management

The system shall provide users with access to their transaction history, allowing them to view details of post orders and payments. The transaction history shall include details such as order date and time, items ordered, total amount charged, payment method used and status of the transaction.

5. User Notification

The system shall send notifications to users regarding the status of payment. Users shall be notified when transactions have been confirmed, failed or still pending. Notifications shall include essential details such as Transaction ID or order number,

amount charged, payment method used and a note message for smooth user experience.

3.1.5 Admin Dashboard Subsystem

The **food delivery system Admin Dashboard** is an important tool of management where the administrators take over, commanding and optimizing the whole process of the system with efficiency. It is also intuitive to work swiftly through different functions and reach the most critical data by admins. What it means in its essence is that the dashboard is offering, in fact, real-time analytics that shall enable administrators to see the live status of each and every order in real time-whether pending, in process, or complete. Further, performance metrics on total sales, average order value, and delivery times shall yield the general efficiency of operations and define further scope for improvement.

The subsystem shall be able to :

1. User management module

This shall be helpful if one needs to manage user accounts comprehensively: one shall create new user profiles, edit the ones that already exist, and manage user statuses-whether they are active, suspended, or deactivated. It gives a very comprehensive overview of user activity, inclusive of the history of orders with feedback, hence helping an admin reach a better understanding of customer behaviour and preference. The module offers full search and filtering capabilities whereby administrators shall easily find users according to various criteria, including frequency of ordering or account status. Meanwhile, it offers direct contact channels to inform users about updates in accord with their accounts using system messages that may concern issues, promotions, or policy updates. By communicating proactively, users shall be more satisfied and attached.

2. Restaurant management module

The Restaurant Management Module puts administrators at the helm of managing the restaurants' life cycles on the platform. The admins shall look through all new restaurant applications and approve only the qualified ones, hence assurance of inclusivity within the service. Besides, it provides an easy way of updating the restaurant profile with actual details relating to menu items, structure of pricing, and

operational hours, hence keeping the information updated. Performance tracking within this module also allows the admins to follow some key metrics: Sales figures, Customers' ratings, and Order volumes. These are the means whereby the admin shall filter out restaurants which are performing very well and support those that are not going that well, thereby allowing mutual help in improvement on the restaurants' part.

3. Order management module

The Order Management Module makes sure that tracking of every single order is effectively done, so that the same are managed in real time. Any pending order, an order in progress, or a completed order shall be marked by the system's administrator. With this, full visibility shall help take quick action wherever things might go wrong-such as issues pertaining to delay of delivery or with the accuracy of order. It shall further provide complete report capability through the detailed analysis it shall be performing on order trends, peak ordering times, and popular menu items. In view of this, administrators shall take the proper staffing, inventory, and promotional strategies to serve their customers even better.

4. Notification and module alerts

In this respect, the Notification and Alert Module is of great instrumental value in keeping everybody aware. It shall, therefore, enable the administrators to send notifications to users and restaurants selectively for any important update, special promotion, or changes in their rules of service. Meanwhile, automated alerts could be set up to notify admins about critical issues, such as dispute frequencies spiking or large changes in volume. This furthers a shared sense of community, wherein the users and restaurants alike are all on the same page. Greater satisfaction and loyalty from the users toward the service then come out as a result of the dynamics between all parties involved.

5. Security and access control module

In fact, security shall always be the first priority on any digital platform. The Security and Access Control Module makes sure that data are secure. This module shall be able to provide, in this module, role-based access controls so the system administrators shall outline what type of permission every system role shall receive or not. Thus, only the rightful people shall be able to contact sensitive data or critical functions, therefore diminishing the chances of data breaches. Also, all the

administrative activities in the dashboard are logged to audit trails in detail. This shall contribute more to accountability through the clear history presented of the actions taken in such a way that any action or illegality in the system shall be readily identifiable.

3.2 Non-Functional Requirements

3.2.1 Security in the Mobile App

Security is highly valued given that many users trust the applications with personal details, payment information inclusive. Said techniques of data protection include TLS/SSL; the mobile app shall use it in protecting transfer of data between the mobile app and the server.

1. Payment Information

The information to be entered for payment shall however be safe; say through encryption such as AES to safely protect credit card and transaction details. The application shall refrain from storing any private data on the local server and in turn, use third party secured payment gateways compatible with relevant prevailing security standards like PCI-DSS.

2. User Authentication

Two-factor authentication (2FA) and biometric verification such as fingerprint or face recognition shall be implemented to secure user accounts, ensuring that only authorized users shall access their accounts and payment methods.

3. Compliance with Data Protection Regulation

It means that the application shall be GDPR-compliant and any other privacy laws meaning that personal data shall not be collected where the user has not given consent. It shall allow the user to have control of data, choices to also view, edit or delete the data as may be needed under legal requirements.

4. Secure Storage

Data of the users stored on the server shall be encrypted. Access controls shall be sensitive and undue access shall not be encouraged. Passwords shall be hashed to ensure security and they shall use brypts in hashing.

3.2.2 System Reliability

System reliability includes such aspects as fault tolerance, error correction, data protection, and ability to regain its normal working state in case of failure. High reliability means that users do not have to worry about the system behaving abnormally with unexpected outages and loss of data ensuring having more trusts and satisfaction from the users.

1. Error Handling

The system shall also embed effective error management system within the platform aimed at preventing, mitigating and correcting errors. This includes displaying useful feedback to users, explaining what the problem is and what shall be done about it when such events occur. Moreover, the system also records such events in a log for future reference to help developers address issues that are likely to arise repeatedly and enhance system performance overall.

2. Data Backup

The system shall schedule the automated backup of all critical data, including restaurant and customer information on a frequent basis, thus protecting itself from data losses. All data in the system shall be backed up at certain intervals and dry copies, stored in safe offsite locations to avoid possibility of loss through hardware failures or catastrophes.

3. Performance Monitoring

The system shall equip itself with sophisticated performance monitoring capabilities that measure key metrics including response time values, system availability, user loads, special events and status. By means of performance management, management shall control the systems in the load and in the abnormality and somehow prevent this influence on the user from the very beginning, ensuring that the systems do not ‘fall apart’ under loads and operate normally and reliably in various abnormal situations.

3. Failover Mechanisms

The system shall contain built-in failover mechanisms that commence seamless switching to backup resources for situations where primary resources such as hardware or software fail. This is in all cases, having provision plans that are immediately active so that they do not cause a down time in the restaurants business.

These Failover mechanisms on their part even though they are of supporting nature shall be regularly exercised to confirm their functionality when required which shall make the system dependable even in unfavourable situations.

3.2.3 Performance in Delivery Management Subsystem

The delivery management system shall be performance-oriented, high-volume transaction processing in the least amount of time. Thus, it responds fast and maintains the load capacity during peak periods. In this way, by optimizing all these aspects, the satisfaction of users increases, which leads to timely deliveries and an efficient process in general.

1. Transaction Processing

A delivery management system shall handle thousands of parallel transactions efficiently. It may range from handling incoming orders to route calculation in real-time. Optimization of database queries together with effective data caching shall significantly enhance transaction speed, reducing wait times for the end-user.

2. Response Time

Quick response times are synonymous with user satisfaction. Users want immediate feedback on status updates to orders or tracking information. The system shall ensure that critical functionalities respond within the needed timeframes to keep users constantly engaged and informed.

3. Load Capacity

The system shall be designed to scale with business growth. It shall accommodate increasing loads during peak time (like holidays or promotions) without performance degradation. Implementing load balancing and cloud-based solutions shall help manage this scalability effectively.

3.2.4 Scalability in Payment Gateway Subsystem

The Payment Gateway Subsystem shall be scalable, it shall be capable of handling higher transactions in the course of the number of users which are likely to exponentially increase when the site Hari Merdeka is most likely to receive very many hits especially during the holidays or specials. This means that it shall work well even when many people are making their payment at any one time.

1. Dynamic Load Management

Load balancing shall be applied by the system in order to distribute the incoming payment requests among different servers. Thus, no server is overloaded with work and all the transactions shall be completed on time.

2. Horizontal Scaling

Another requirement that shall be met by the design of the Payment Gateway is the fact that it shall be horizontally scalable; this implies that as the number of transactions increases, it shall be easy to add a new server. It assists the system to keep on working without many interruptions because it is flexible.

3. Performance Monitoring

To this end, the system shall display tools that shall be used to check a systems efficiency in the performance of the tasks that it is allocated to do. This means we need to look at how quickly transactions are being cleared and the amount of traffic on any particular server. Thus, monitoring this data shall be done to see that all these aspects are optimized to work effectively.

4. Elastic Cloud Infrastructure

It is thus important that the Payment Gateway deploy cloud services that shall self-provision, that is, be able to allocate more or less resources depending on the required loads. This also means that if there is a good workload of transactions for instance, the system shall bring more resources to bear without having to inform us.

5. Database Optimization

There is thus the need to ensure that the database that supports the Payment Gateway shall be keen to support growth. This shall for example include sharding where data is spread across many different servers and replication where data is created copies to enhance performance as the transaction intensity rises.

3.2.5 Maintainability in Admin Dashboard Subsystem

The ease with which a software system may be changed to fix defects, bring improvements in performance, or accommodate changes in its operating environment. Regarding the admin dashboard, high maintainability would mean that the system is going to keep on evolving and become able for its usage. That is, the major factors of concern of maintainability are:

1. Code Quality

Standards Compliance Code should follow set coding standards such as naming convention, structure, and style guides. In such a way, this will make a working environment very easy for developers working in the codebase. Modularity: It makes segregation of functionalities into small, manageable components or services quite easy; thereby, each of them can be updated or debugged without any issues. Documentation: This includes everything from complete code comments and API documentation to user manuals. Good documentation explains something that either is, how something works, how the environment is set up, or how to get started for new developers.

2. Version Control

Source Code Management: With the use of any version control system, track all changes made, collaborate with team members, or manage releases. Thus, it helps record the modification history and trace back in case something goes wrong. Branching Strategy: Follow a strategy like Git Flow that will efficiently handle features, bugfixes, and releases. This separates the stable code from the active development ones.

3. Testing

Testing Automation: Automated unit tests, integration tests, and end-to-end tests shall be applied. That helps find issues early in the development life cycle and reduces the possibility of bugs slipping into production. Test Coverage: Higher test coverage whereby most of the critical paths are ensured. The extent of the test coverage can be ascertained using the available tools to prepare metrics and reports.

4. Handling errors and logging

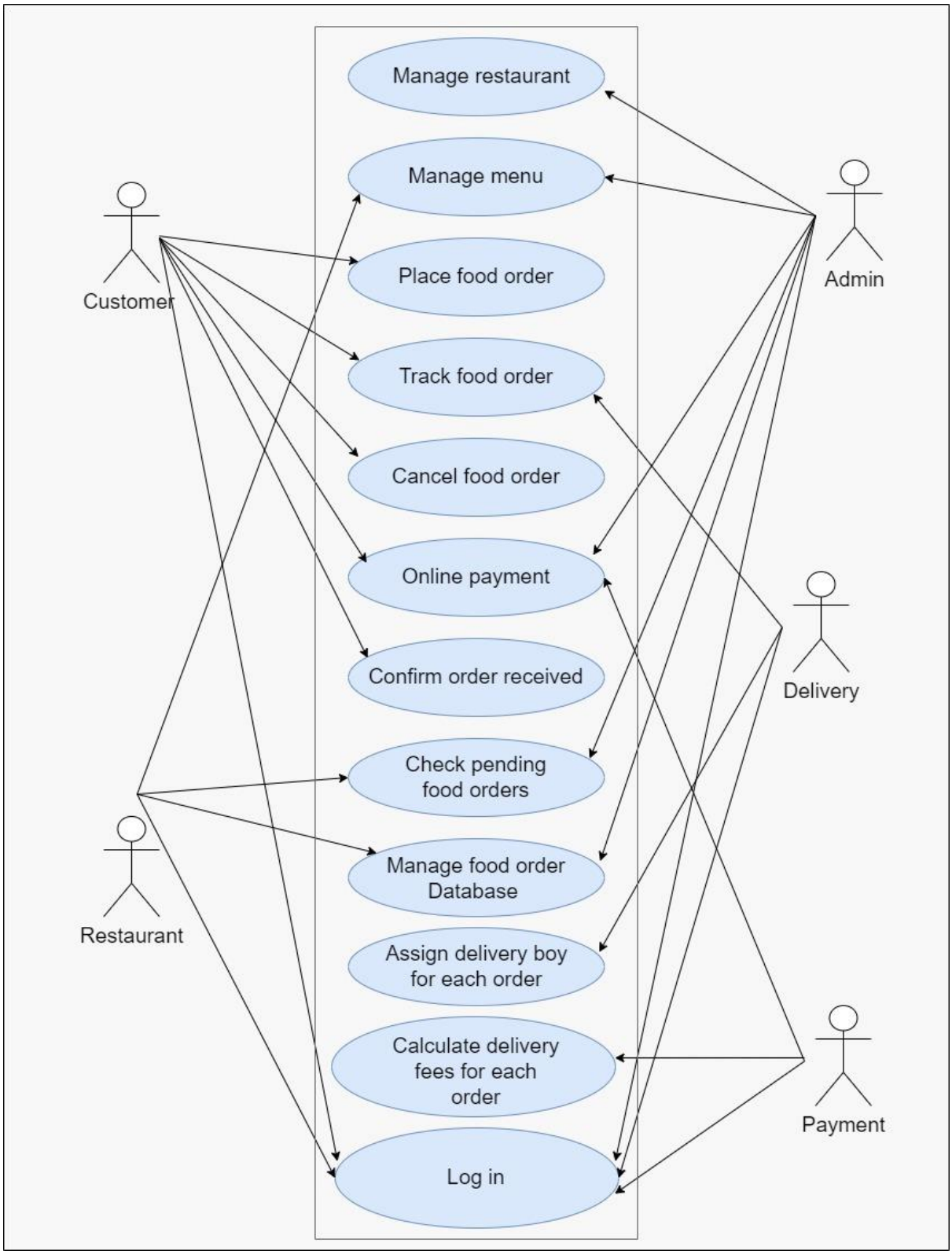
Comprehensive Error Handling: Include robust error handling for any exception, thus allowing ease in sending error messages to the user. Centralized Logging: A mechanism for centralized logging can be utilized to fetch logs from all components composing the dashboard. It helps locate the problem faster, hence providing help for debugging. 5. Dependencies Management Package Management: Any third-party libraries required are to be maintained using any utility like npm or Maven; every dependency needs updating for security patches and other improvements. Compatibility Tests: Run compatibility tests for any update of the involved libraries that might break the functionality.

5. User Testing and Iteration

Feedback System: This allows users to give feedback with regard to the dashboard-on feature requests, reporting bugs, among others. This will help guide maintenance efforts in the quest for improvement. Iterative Development: To utilize agile methods and processes for iterative development and deliverables, hence fast and effective response to the day-to-day user's needs and issues.

4. Case Diagram

Food Delivery Management System



5. References

Scribd. (n.d.). *Literature Review On Food Delivery Service | PDF | Fast Food Restaurants | Foods*. [online] Available at: https://www.scribd.com/document/461418440/literature-review-on-food-delivery-service/

www.linkedin.com. (n.d.). Architecture and Design Principle for Online Food Delivery System. [online] Available at: https://www.linkedin.com/pulse/architecture-design-principle-online-food-delivery-system-sharma/

mohadikar, sachin (2020). *Food delivery & Agile Methodology*. [online] Medium. Available at: https://medium.com/@skmohadikar/food-delivery-agile-methodology-85f929b50db9/

Sharma, M. (2024). *Mastering Menu Management: A Guide for Restaurant Enterprises - Retrotorts Blog*. [online] Restroworks Blog. Available at: https://www.restroworks.com/blog/mastering-menu-management-a-guide-for-restaurant-enterprises/

Allred, D. (2023). *Restaurant Inventory Management [Your Ultimate Guide for Success] - TheRealBarman*. [online] TheRealBarman - The Ultimate Resource For All Things Bar & Restaurant. Available at: https://therealbarman.com/restaurant-inventory-management-success/

D, L. (2022). *Delivery Management System: A Definitive Guide*. [online] Locus Blog. Available at: https://blog.locus.sh/delivery-management-system/

Altexsoft (2023). *Functional and Non-functional Requirements: Specification an*. [online] AltexSoft. Available at: https://www.altexsoft.com/blog/functional-and-non-functional-requirements-specification-and-types/

WEB Pays (2024). *Explore the importance of scalability in payment processing solutions, and how it supports business growth, enhances customer experience, and ensures operationa*. [online] Linkedin.com. Available at: https://www.linkedin.com/pulse/importance-scalability-payment-processing-solutions-webpays-akr9c/

Payment (2024). *Payment Scalability and Performance: Driving Business Success through Payment Scalability and Performance - FasterCapital*. [online] FasterCapital. Available at: https://fastercapital.com/content/Payment-Scalability-and-Performance--Driving-Business-Success-through-Payment-Scalability-and-Performance.html/

[Accessed 5 Oct. 2024].

Checkout.com. (2023). *Features to look for in an online payment system*. [online] Available at: https://www.checkout.com/blog/features-online-payment-system/

[Accessed 5 Oct. 2024].

Nuclino (2024). *A Guide to Functional Requirements (with Examples)*. [online] www.nuclino.com. Available at: https://www.nuclino.com/articles/functional-requirements/

Altexsoft (2023). *Functional and Non-functional Requirements: Specification an*. [online] AltexSoft. Available at: https://www.altexsoft.com/blog/functional-and-non-functional-requirements-specification-and-types/.

GeeksforGeeks. (2023). *Maintainability in System Design*. [online] Available at: https://www.geeksforgeeks.org/maintainability-in-system-design/.