A picture containing emblem, symbol, crest, badge

Description automatically generated

**Food Delivery App Documentation**

**Submitted fulfillment**

**Of the requirements of**

**Graduation project**

|  |  |
| --- | --- |
| **Project Guide:**   * **Dr.** **Farhan Alebeisat** | Submitted by:   * Shadi alsoudi * Abdalrhman zahran |

# Table of Contents

1. ***INTODUCTION…………………………….……………………….…………………………….2***

*1.1 Problem Statement……………………………………………………….…….2*

*1.2 Purpose of the Project………………………….…………………………………2*

*1.3 Project Software and Hardware Requirement….…………………………..3*

*1.4 Project Schedule………………………….…………………………………………..……..5*

1. ***SYSTEM ANALYSIS…………………………………………………………………...6***

*2.1 STEEPLE ANALYSIS …………………………………………………………….6*

*2.2 System* requirements*…………………………………………….…………...7*

*2.2.2 Overview of Functional Requirements……………………………………8*

*2.2.3 non Functional Requirements…………………………………………………….11*

***3. SYSTEM DESIGN…….………………………………...……………………………………...13***

*3.1. Introduction.…….………………………………………………………………….….13*

*3.1.2 Context Diagram and DFD Diagram………………………………………………13*

*3.1.3 UML Use Case Diagram……………………………………………………………………….17*

*3.1.4 Activity Diagram………………………………………………………………………………..……19*

*3.1.4 Sequence Diagram………………………………………………………………………………………20*

***4. SYSTEM IMPLEMENTATION …………………………………………………….21***

*4.1.1 Introduction …………………………………………………………………21*

* + 1. *Database Implementation …………………………………………… ..22*

***5.******SYSTEM TESTING AND INSTALLATION ……………………………………………………..23***

*5.1 Requirements Validation and Completeness……………………………….23*

*5.2 System Installation……………………………………………………………………………..26*

**6. References or resources used………………………………………………………………….28**

**CHAPTER ONE: INTRODUCTION**

**1.1 Problem Statement for Food Delivery App**

*Now a day the use of the smartphone is increasing day by day. Every person needs his own phone device because it changes lifestyle and is very helpful for every part of life.*

*So we make an Application for a restaurant through which a customer can easily access all the things that belong to a restaurant while sitting anywhere instead of going to a restaurant and avoiding facing rush which is a very major problem in a restaurant the time is wasted too much in ordering food while preparing order, it took a lot of time.*

*The purpose of working on this project is to provide ease to customers so that they can order anything, anytime while sitting anywhere and it can change the food ordering system in a better and more appropriate way.*

*The app should focus on minimizing delivery times, ensuring accurate order fulfillment, offering a wide selection of restaurants and cuisines, providing a user-friendly interface, and delivering prompt and efficient customer support.*

**1.2 Purpose of the Project**

*The purpose of a food delivery app project is :*

* ***Convenience****: The primary goal is to provide a convenient way for customers to order food from various restaurants and have it delivered to their homes. The app simplifies the food ordering process by eliminating the need to physically visit a restaurant or make phone calls.*
* ***Efficiency****: The project aims to streamline the food delivery process for both customers and restaurants. The app facilitates quick and efficient order placement, payment processing, and delivery coordination through an online platform.*
* ***Increased accessibility****: The app intends to make food delivery accessible to a larger audience. It caters to people with mobility limitations, busy schedules, or those living in areas with limited dining options..*
* ***Enhanced customer experience****: The project strives to provide a user-friendly interface and intuitive features that improve the overall customer experience.*
* ***Business growth****: The foody delivery app project benefits restaurants and food establishments by expanding their customer base and reach. It provides them with a digital platform to showcase their offerings and attract new customers who prefer the convenience of online ordering.*

*In summary, a food delivery app project aims to create a seamless and convenient platform that connects customers with restaurants, facilitating efficient food delivery while offering an enhanced user experience for all stakeholders involved.*

**1.3 Project Software and Hardware Requirements**

**Software Requirements:**

* ***Flutter SDK****: Flutter is an open-source UI toolkit for building natively compiled applications for mobile, web, and desktop platforms. we would need to install the Flutter SDK, which includes the Flutter framework, Dart programming language, and command-line tools.*

*Compatible Operating Systems: Windows, macOS, Linux*

* ***Integrated Development Environment (IDE****):*

*Visual Studio Code (with Flutter and Dart extensions)*

*Android Studio (with Flutter and Dart plugins)*

* ***Firebase:*** *Firebase is a cloud-based platform provided by Google for building and managing backend services. We will need to set up a Firebase project and configure the necessary services, such as Firebase Authentication, Firebase Firestore (for database).*

***Firebase SDKs and libraries****: we will need to include the appropriate Firebase SDKs and libraries in your Flutter project to integrate with Firebase services.*

***Hardware Requirements****:*

* ***Processor****: A multi-core processor with a clock speed of at least 2.0 GHz or higher is recommended for smooth development.*
* ***Memory (RAM):*** *A minimum of 8 GB RAM is recommended for efficient development, although more RAM can help with larger projects and faster compilation times.*
* ***Storage****: should have sufficient disk space for installing the required software, IDE, and SDKs. A minimum of 10 GB free disk space is recommended.*
* ***Display****: A monitor with a resolution of 1280x800 or higher is recommended for a comfortable development experience.*
* ***Internet Connection****: A stable internet connection is required for downloading dependencies, documentation, and accessing cloud-based services (Firebase).*

**1.4 Project Schedule:**

|  |  |  |
| --- | --- | --- |
| 1-Project Initiation | * Define project objectives and scope * Identify key features and functionalities of the app | **1 (week)** |
| 2-Design and Architecture | * Design the database schema and system architecture * Finalize the app's visual design | **(2 weeks)** |
| 3-Development | * Set up the development environment * Implement the front-end interface using Flutter framework. * Develop the back-end functionality | **(11 weeks)** |
| 4-Testing and Quality Assurance | * Conduct unit testing to ensure individual components work correctly * Perform integration testing to verify the interaction between different modules * Test the app's performance, usability * Identify and fix any bugs or issues | **(3weeks)** |

**CHAPTER TWO: SYSTEM ANALYSIS**

***2.1 STEEPLE ANALYSIS***

***A. Social Impact***

*The aim of this project is to help people choose their meals more accurately, by having*

*access to a larger choice of restaurants and menus. In order to make the process of*

*selection easier,,The online ordering feature may help workers and students order their lunch*

*online.This application would also help managers to better promote their restaurants.*

***B. Technological Impact***

*To build this application, new technologies and tools would be used. These*

*technologies are open-source and will be used to complete the project in the most*

*efficient way.*

***D. Economic Impact***

*With this application, restaurants may attract more customers, which will increase their*

*revenues. Since the main feature of the application are free for both the users and the*

*managers, there is no potential loss for any of them. Some additional features might be*

*added afterwards, to let managers have access to statistics about their restaurants, with the purpose of helping restaurants boost their revenues.*

***G. Ethical Impact***

*Ethically, it is extremely important to secure the application to avoid any data leak.*

*Any new feature should be tested to avoid vulnerabilities, passwords should be encrypted and stored securely in a database*

**2.2 System requirements**

**2.2.1 Clients, customer and users**

***Customers (End-users Placing Orders):***

*Customers are the end-users who use the app to browse restaurants, place orders, and make payments. Their requirements revolve around a seamless and convenient ordering experience. Consider the following requirements for customers:*

***User Registration****: Allow customers to create accounts, providing necessary details like name, contact information, and delivery address.*

***Restaurant Discovery****: Enable customers to search for restaurants based on location, cuisine, ratings, and other filters.*

***Menu Browsing****: Provide an intuitive interface for customers to browse menus, view dish details, check prices, and customize their orders.*

***Ordering and Payment****: Facilitate smooth order placement, including options for adding items to the cart, selecting delivery options,*

***Order Tracking****: Implement order tracking functionality that allows customers to track the status of their orders.*

***Users (Delivery Personnel, Restaurant Staff):***

*Users refer to the individuals involved in the delivery process, including delivery personnel and restaurant staff. the following requirements for users:*

***Delivery Management****: Provide a dedicated interface for delivery personnel to manage their availability, accept or reject delivery requests, and navigate efficiently using GPS integration.*

***Order Dispatching****: Enable the seamless dispatching of orders from restaurants to delivery personnel, ensuring efficient allocation and optimized routes.*

***Communication****: Include in-app communication features to facilitate communication between delivery personnel and customers/restaurants, allowing for updates, clarifications, and notifications.*

**2.2.2 Functional and Data Requirements:**

*Functional requirements define the specific functionalities and features that the food delivery app must possess to fulfill the needs of its users. Data requirements, on the other hand, outline the necessary data elements and structures that the app must handle. Here are some functional and data requirements:*

***Functional Requirements:***

***USERS:***

* ***User Registration and Authentication:***

*Allow users to create accounts, providing necessary information such as name, contact details, and delivery address.*

* ***Restaurant and Menu Listing:***

*Enable users to browse and search for restaurants*

*Display comprehensive information about each restaurant, including menus, prices and special offers.*

* ***Menu Customization and Ordering:***

*Allow users to customize their orders by selecting options like quantity,*

*Provide a cart feature for users to add items, review the order, and proceed to payment.*

*Provide a dedicated section within the app where users can view their order history.*

*Display the relevant details of each order, such as the restaurant name, ordered items, order date and time, total cost, and delivery status.*

* ***Order History Display:***

*Provide a dedicated section within the app where users can view their order history.*

*Display the relevant details of each order, such as the restaurant name, ordered items, order date and time, total cost, and delivery status.*

* ***Search Functionality:***

*Implement a search feature to help users find specific food or restaurant based on keywords, restaurant names, or any other relevant search criteria.*

***Restaurant:***

* ***Add Menu****: This functionality allows the restaurant to create and add menus to their profile within the app.*
* ***Add Items to Menu****: With this feature, the restaurant can add individual food items to the menus they have created. They can provide details for each item, including name, description, price, ingredients, and any customization options available.*
* ***Delete Menu or Item****: The restaurant should have the ability to delete menus or individual menu items when needed. This allows them to manage their offerings and update their menus based on availability.*
* ***Receiving New Orders****: This functionality enables the restaurant to receive incoming orders placed by customers through the app. When a new order is placed, the restaurant receives a notification and can view the order details, including the items ordered, customer information, delivery address.*
* ***Display Total Earnings****: The app should provide a feature that displays the restaurant's total earnings or revenue generated through the app. This allows the restaurant to track and monitor their financial performance within the app.*
* ***Display History Orders****: This feature allows the restaurant to view their order history, including details of past orders that have been fulfilled. The history orders section provide information such as order dates, items ordered, customer details, order status, and order totals.*
* ***Sign Out****: The restaurant should have the ability to sign out of their account within the app..*

***Rider:***

***Login and Authentication****: Riders should be able to log in to the app using their credentials and authenticate themselves.*

***Order Management****: Riders should receive assigned delivery orders from the system, including order details such as pickup location, delivery location, items, and any special instructions.*

***Navigation Assistance****: The app should provide navigation assistance to riders, guiding them to the pickup and delivery locations using maps and directions.*

***Order Confirmation:*** *Riders should have the ability to confirm the successful pickup of an order once they have collected it from the restaurant or vendor.*

***Delivery Confirmation****: Riders should be able to confirm the successful delivery of an order to the customer's specified location.*

***Earnings and Payments:*** *The app should provide visibility into the rider's earnings, including details of completed deliveries, tips, and payment processing.*

***Admin:***

* ***Admin Can Block Any User****: This functionality allows the admin to block or suspend user accounts if they violate app policies or engage in inappropriate behavior. Blocking a user prevents them from accessing the app and placing orders permanently.*
* ***Admin Can Approve Any Restaurant Account****: When a restaurant registers for an account on the app, they go through an approval process. The admin has the authority to review and approve or reject restaurant account requests. This ensures that only legitimate and compliant restaurants are allowed to operate within the app.*
* ***Admin Can Block Any Restaurant Account****: Similar to blocking users, the admin has the capability to block or suspend restaurant accounts if they violate app policies or fail to meet certain requirements. Blocking a restaurant account restricts their ability to receive orders or manage their profile on the app.*
* ***Admin Can View Info About User or Restaurant:*** *This functionality enables the admin to access and view detailed information about users and restaurants registered on the app. This includes user details such as name, contact information, order history, and any reported issues. For restaurants, the admin can view their profile information, menu, order history.*

**2.2.3 Non-functional Requirements:**

*Non-functional requirements specify the attributes and characteristics of the food delivery app that are not directly related to its functionality but are essential for its overall performance, usability, security, and portability.*

***2.2.3.1 Look and Feel Requirements:***

*The app should have an attractive and visually appealing user interface (UI) design.*

*The UI elements, color schemes, and typography should be consistent and visually coherent throughout the app.*

*The app should provide a user-friendly and intuitive navigation system.*

*The app should be responsive and adapt to different screen sizes and orientations.*

***2.2.3.2 Usability Requirements:***

*The app should be easy to learn and use for both customers and restaurant owners.*

*The user interface should be intuitive, minimizing the need for extensive user training or assistance.*

***2.2.3.3 Security Requirements:***

*The app should implement robust security measures to protect user data, including personal information and payment details.*

*The app should use secure communication protocols to transmit data between users, restaurants, and the server.*

*Access to sensitive app features or administrative functions should be restricted to authorized individuals or roles.*

***2.2.3.4 Performance Requirement:***

*The app should have fast response times and minimal loading delays to provide a seamless user experience.*

*The app should handle concurrent user requests and traffic without significant performance degradation.*

*The app should efficiently process and display menus, restaurant listings, and order information.*

*The app should optimize data storage and retrieval operations for efficient performance.*

***2.2.3.5 Portability Requirements:***

*The app should be compatible with multiple platforms and devices, such as iOS and Android smartphones and tablets.*

*The app should support different screen resolutions and orientations to ensure consistent user experience across devices.*

*The app should be easily deployable and upgradable to accommodate future.*

**CHAPTER THREE: SYSTEM DESIGN**

***3.1 Introduction***

The system design phase is a critical step in the software development life cycle where the requirements gathered during the analysis phase are transformed into a detailed system design.

The main objective of the system design phase is to create a blueprint or roadmap for the development team to follow. It involves defining the system architecture, designing the database schema, creating user interfaces, and specifying the interaction between various system components.

***3.2 Context Diagram and Data Flow Diagram (DFD)***

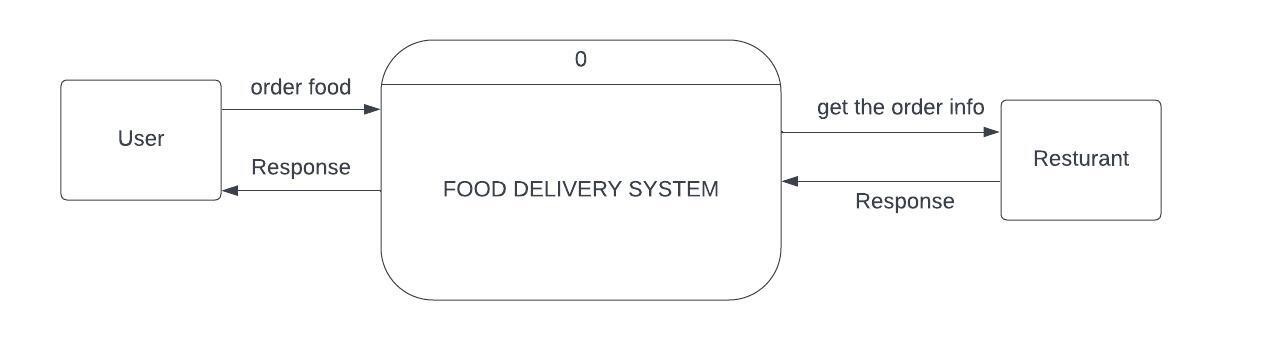
*Context diagram It's a basic overview of the whole system or process being analyzed or modeled. It's designed to be an at-a-glance view, showing the system as a single high-level process, with its relationship to external entities*

*A Data Flow Diagram (DFD) is a traditional visual representation of the information flows within a system. A neat and clear DFD can depict the right amount of the system requirement graphically. It can be manual, automated, or a combination of both.*

*It shows how data enters and leaves the system, what changes the information, and where data is stored.*

*The objective of a DFD is to show the scope and boundaries of a system as a whole. It may be used as a communication tool between a system analyst and any person who plays a part in the order that acts as a starting point for redesigning a system. The DFD is also called as a data flow graph or bubble chart.*

**Context Diagram (DFD level 0)**

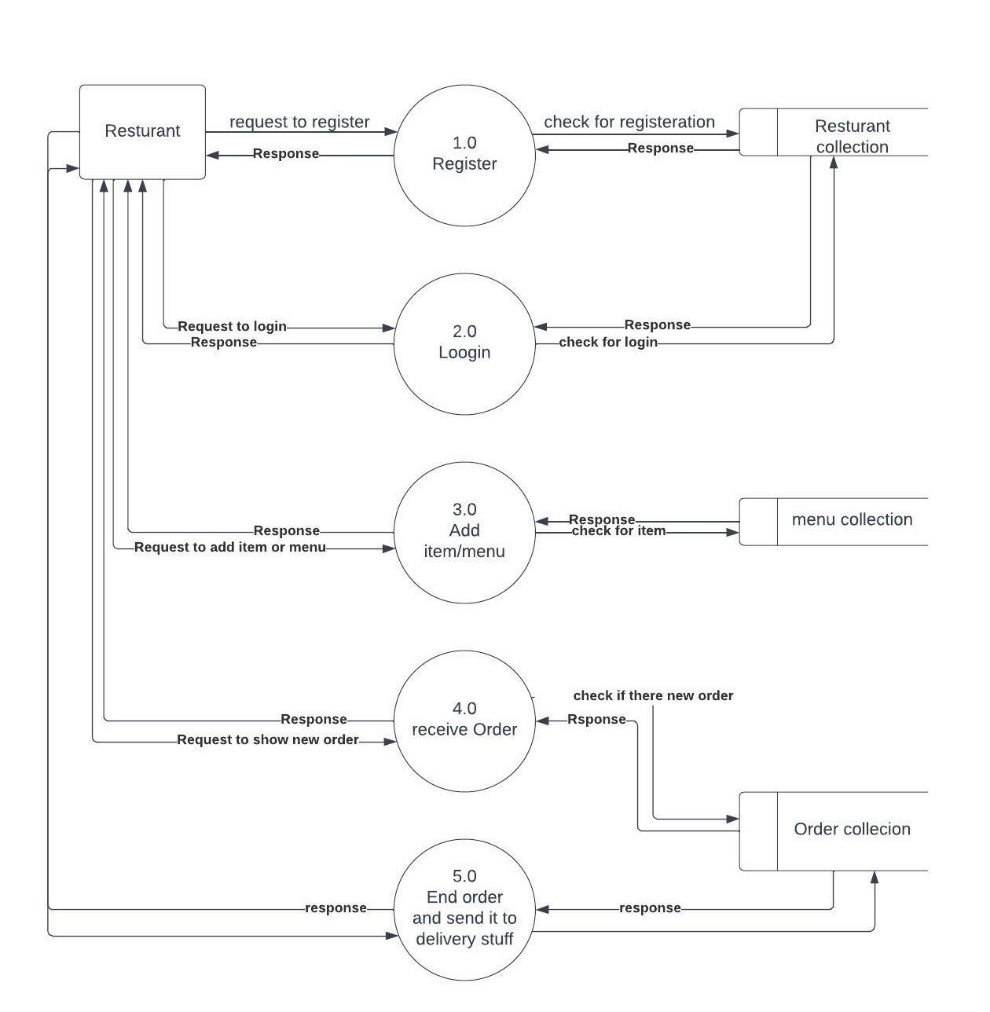
****

**User DFD diagram level 1**

**A picture containing text, diagram, parallel, plan

Description automatically generated**

**Resturant DFD diagram level 1**

****

**3.3UML Use Case Diagram**

*A UML use case diagram is the primary form of system/software requirements for a new software program underdeveloped. Use cases specify the expected behavior (what), and not the exact method of making it happen (how). Use cases once specified can be denoted both textual and visual representation ,A key concept of use case modeling is that it helps us design a system from the end user's perspective. It is an effective technique for communicating system behavior in the user's terms by specifying all externally visible system behavior.***A picture containing text, diagram, sketch, plan

Description automatically generated**

Use case diagram

Explain some of important use case:

***Use Case Name: Order Food***

*Use Case Number: UC01*

*Actors: User (Customer)*

*Overview: This use case captures the process of a user ordering food through the app.*

*Related Use Cases: None*

*Typical Flow Description:*

*User is logged into the app and has selected a restaurant.*

*user browses the menu and selects the desired items to add to the order.*

*User can customize the order by specifying options, quantities, or any special instructions.*

*User reviews the order details, including the selected items, quantities, and total cost.*

*User proceeds to checkout and selects the preferred payment method.*

*User provides the necessary payment information and confirms the order.*

*The system verifies the payment and sends a confirmation to the user.*

*The order is forwarded to the restaurant for preparation and delivery.*

***Use Case Name: Add Menu and Item***

*Use Case Number: UC02*

*Actors: Restaurant*

*Overview: This use case captures the process of a restaurant adding a menu and its items to the app.*

*Related Use Cases: None*

*Typical Flow Description:*

*Restaurant is logged into the app and has access to their account.*

*Restaurant selects the option to add a new menu.*

*Restaurant enters the menu details, including the name, description*

*Restaurant adds individual items to the menu, including the item name, description, price, and any additional options or variations.*

*Restaurant reviews the menu and item details to ensure accuracy.*

*Restaurant saves the menu and items, making them available for customers to view and order.*

*Postcondition: The menu and items are added to the restaurant's profile and can be accessed by customers.*

***Use Case Name: Approve and Block***

*Use Case Number: UC04*

*Actors: Admin*

*Overview: This use case captures the process of the admin approving or blocking user and restaurant accounts.*

*Related Use Cases: None*

*Typical Flow Description:*

*Admin is logged into the admin panel with appropriate privileges.*

*Admin accesses the user or restaurant account management section.*

*Admin selects a user or restaurant account that requires approval or blocking.*

*Admin reviews the account details, including user information or restaurant details.*

*Admin verifies the account information and checks for any required documents or criteria.*

*Admin approves the account if all requirements are met and the account is deemed valid.*

*If the account does not meet the requirements or violates any policies, the admin has the option to block the account.*

*The system updates the account status accordingly, either approving or blocking the account.*

**3.4 activity diagram:**

*Activity diagram is another important diagram in UML to describe the dynamic aspects of the system.*

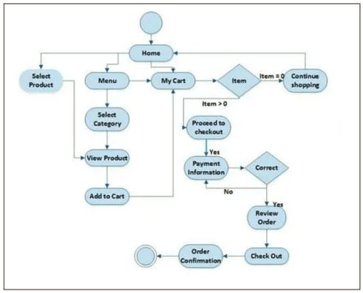
*Activity diagram is basically a flowchart to represent the flow from one activity to another activity. The activity can be described as an operation of the system.*

*The control flow is drawn from one operation to another. This flow can be sequential, branched, or concurrent. Activity diagrams deal with all type of flow control by using different elements*

**A picture containing diagram, screenshot, line

Description automatically generated**

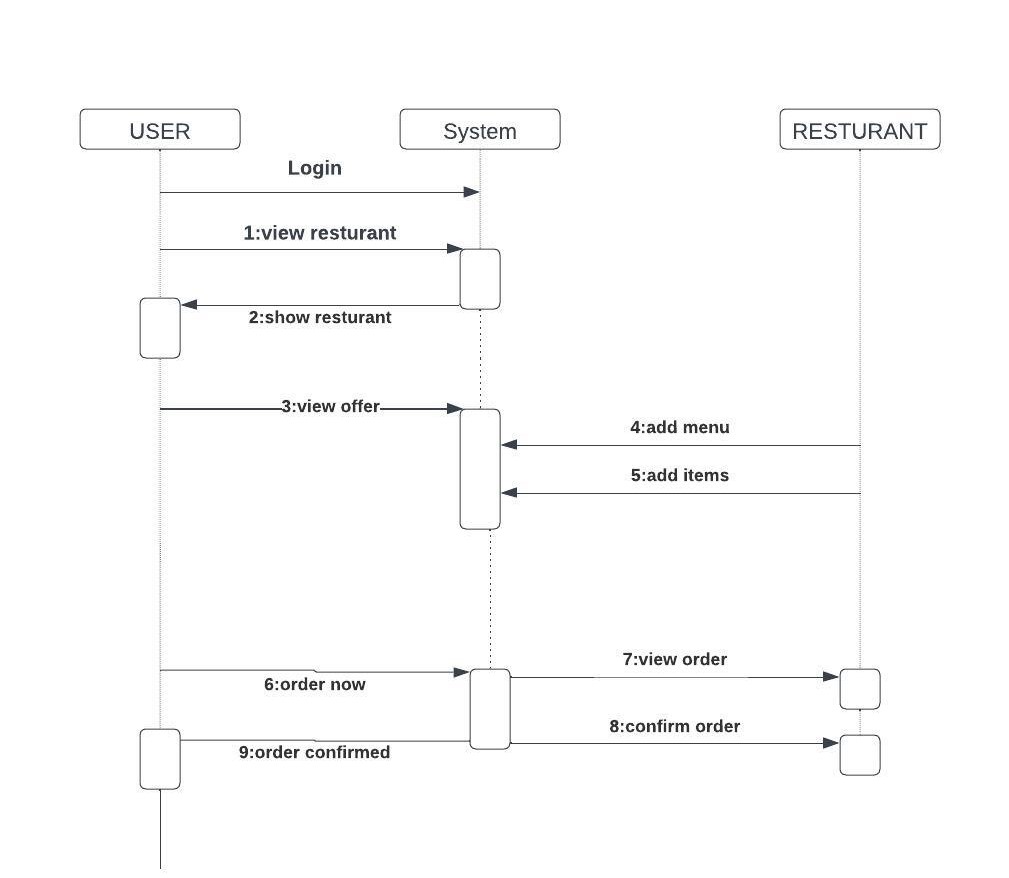
Activity diagram

****

Activity diagram

**3.5 Sequence diagram**

*A sequence diagram or system sequence diagram (SSD) shows process interactions arranged in time sequence in the field of software engineering. It depicts the processes and objects involved and the sequence of messages exchanged between the processes and objects needed to carry out the functionality. Sequence diagrams are typically associated with use case realizations in the architectural view model of the system under development. Sequence diagrams are sometimes called event diagrams or event scenarios.*



Sequence diagram

**CHAPTER FOUR: SYSTEM IMPLEMENTATION**

4.1 Introduction

*The implementation phase is a crucial part of the software development life cycle where the design specifications are transformed into a working system. In this chapter, we will provide an introduction to the system implementation process for the food delivery app.*

*During the implementation phase, the development team takes the design artifacts, such as the system architecture, database schema, and user interface design, and translates them into executable code. This involves writing the necessary code, configuring the database, integrating external APIs or services, and creating the graphical user interface (GUI).*

**4.2: Database Implementation using Firebase for a Food Delivery App**

*When using Firebase as the database for the food delivery app, the traditional concept of tables and relationships is replaced with collections and documents. Firebase uses a NoSQL database structure, which is flexible and scalable. Here's an Scheme of how the database implementation can be done using Firebase Firestore:*

*Collections and Documents:*

*Instead of tables, we have collections that contain documents. Each document represents an entity and has its own unique identifier. Here are the collections and their respective documents:*

**Collection: users**

Document: { userId, name, email, UserAddress, phone , PhotoUrl , UserCart,status}

**Collection: Seller**

Document: {sellerUid, SellerName, address, phone, sellerEmail, sellerAvatarUrl ,status}

**Collection: menus**

Document: { menuId, restaurantId, menuInfo,menuTitle,status,image, }

**Collection: items**

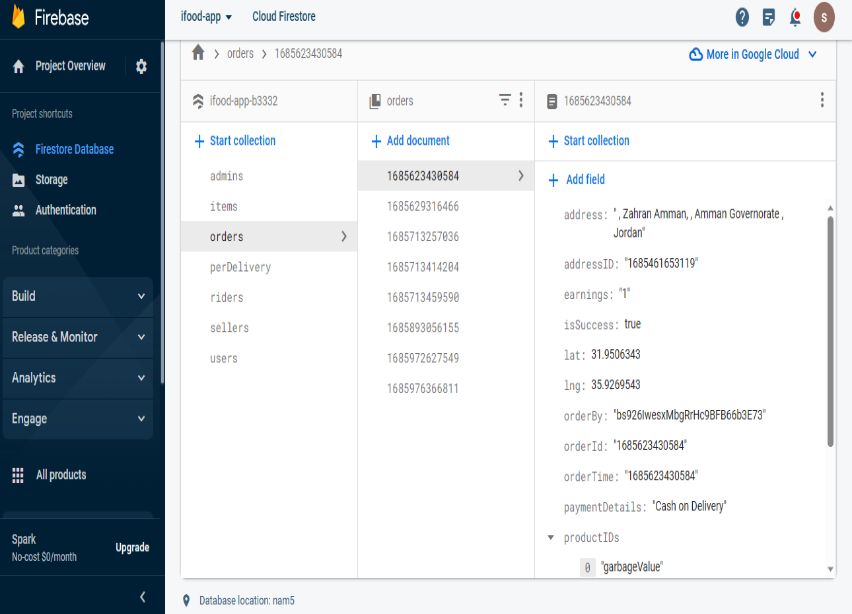
Document: { itemId, menuId,sellerId, name,shortInfo, description, price, image }

**Collection: orders**

Document: { orderId, userId, restaurantId, status, totalAmount, deliveryAddress, orderDate,Payment Detail }

**Collection: admin**

Document: {email,name}



**CHAPTER FIVE: SYSTEM TESTING AND INSTALLATION**

***5.1 Requirements Validation and Completeness:***

***Testing Phase:***

*The testing phase has been conducted after the completion of implementation for this application.*

*The purpose of the testing phase helps to test this application has operated successfully regarding the needs of users. System functional tests are crucial for an application because it helps to guarantee the application operates successfully and is able to find out the errors and bugs within the application for purposes to*

*improve in future works. The Tables show the result of the test cases of the application.*

***User Registration and Authentication:***

|  |  |  |
| --- | --- | --- |
| ***Test Cases*** | ***Expected Result*** | ***Actual Result*** |
| *User registration with valid details* | *User registration is successful.* | *pass* |
| *User registration with existing email* | *Registration fails. Error message displayed.* | *pass* |
| *User login with valid credentials* | *User is successfully logged in.* | *pass* |
| *User login with incorrect password* | *Login fails.* | *Pass* |

***Restaurant and Menu Listing:***

|  |  |  |
| --- | --- | --- |
| ***Test Cases*** | ***Expected Result*** | ***Actual Result*** |
| *Browse restaurants* | *Restaurants are displayed* | *pass* |
| *Search for a restaurant by name* | *The restaurant appear in search page* | *Fail* |
| *View restaurant details and menu* | *The page displays the restaurant menu.* | *pass* |
| *User login with incorrect password* | *Login fails.* | *Pass* |

***Order History Display and Ordering:***

|  |  |  |
| --- | --- | --- |
| ***Test Cases*** | ***Expected Result*** | ***Actual Result*** |
| *Add items to cart* | *The selected items are added to the cart.* | *Pass* |
| *Place an order* | *The order is successfully placed.* | *Pass* |
| *View order history* | *The user's order history is displayed.* | *Pass* |
| *View order details* | *The detailed information of the selected order is displayed.* | *Pass* |

**Restaurant Features:**

|  |  |  |
| --- | --- | --- |
| ***Test Cases*** | ***Expected Result*** | ***Actual Result*** |
| *Add menu* | *The new menu is successfully added to the restaurant's profile.* | *Pass* |
| *Add a new food item to a menu* | *The new food item is successfully added to the menu.* | *Pass* |
| *Delete a menu* | *The menu is successfully deleted from the restaurant's profile.* | *Pass* |
| *Delete a food item* | *The food item is successfully deleted from the menu.* | *Pass* |
| *Receive a new order* | *The restaurant receives a new order and display it .* | *Pass* |
| *View order details* | *The detailed information of the selected order is displayed.* | *Pass* |
| *Display Total earnings* | *The earnings section is displayed.* | *Pass* |

**5.2 System Installation for Food Delivery App**

*To install and set up the food delivery app built with Flutter and Firebase, follow the steps outlined below:*

***Flutter SDK Installation:***

*Download and install the Flutter SDK from the official Flutter website (https://flutter.dev).*

*Set up the necessary environment variables for Flutter on your development machine.*

*Verify the installation by running flutter doctor in the command line to ensure all dependencies are properly installed.*

***Firebase Configuration****:*

*Create a Firebase project on the Firebase Console (https://console.firebase.google.com).*

*Set up Firebase services required for the app, such as Firebase Authentication, Cloud Firestore (database), Firebase Storage.*

*Obtain the Firebase configuration files (google-services.json for Android, GoogleService-Info.plist for iOS) specific to your project.*

***Development Environment Setup****:*

*Choose an Integrated Development Environment (IDE) for Flutter development, such as Android Studio or Visual Studio Code.*

*Install the necessary Flutter and Dart plugins in your IDE to enable Flutter development.*

*Import the Flutter project into your chosen IDE by selecting the project directory.*

***Dependency Configuration:***

*Open the pubspec.yaml file in your Flutter project.*

*Add the required Firebase dependencies, such as firebase\_core, firebase\_auth, cloud\_firestore, firebase\_storage, etc., to the dependencies section.*

*Save the file and run flutter pub get in the command line to fetch the added dependencies.*

**Firebase Integration:**

*Follow the FlutterFire documentation (https://firebase.flutter.dev) to integrate Firebase services into your Flutter app.*

*Configure Firebase Authentication to enable user registration, login, and authentication in the app.*

*Set up Cloud Firestore as the database for storing restaurant information, menus, orders, and other relevant data.*

*Configure Firebase Storage to handle image uploads and storage for restaurant menus and profile pictures.*

*Testing and Deployment:*

*Use the Flutter development tools to test and debug the app on emulators, simulators, or physical devices.*

*Ensure that the app's functionality, such as restaurant browsing, menu selection, order placement works as expected.*

**References:**

1. *Flutter Documentation: Official documentation for the Flutter framework, providing comprehensive guides, tutorials, and API references. Available at: https://flutter.dev/docs*
2. *Firebase Documentation: Official documentation for Firebase, covering various Firebase services such as Firebase Authentication, Cloud Firestore, Firebase Storage, and more. Available at: https://firebase.google.com/docs*
3. *FlutterFire: A set of Flutter plugins that enable Flutter apps to use Firebase services. The documentation provides instructions on integrating Firebase into a Flutter app. Available at: https://firebase.flutter.dev*
4. *Material Design Guidelines: Google's official design guidelines for creating visually appealing and user-friendly apps using the Material Design principles. Available at:* [*https://material.io/design*](https://material.io/design)
5. *Foodpanda, 2012. Available: Foodpanda, https://www.foodpanda.my. [Accessed October 8, 2019]*
6. [*https://www.udemy.com/course/learn-flutter-dart-to-build-ios-android-apps/*](https://www.udemy.com/course/learn-flutter-dart-to-build-ios-android-apps/)