Project Report: MOFOOD - Food Delivery App

INDEX

1. INTRODUCTION

MOFOOD: Revolutionizing Food Delivery In the fast-paced world we live in, the demand for convenience has driven a significant transformation in the food industry. Recognizing the need for a seamless and efficient solution, we proudly introduce MOFOOD, a cutting-edge food delivery app designed to redefine the way users experience and interact with the culinary world.

1.1 Project Overview

MOFOOD is not just another food delivery app; it's a comprehensive platform crafted with precision to offer users a delightful journey from the moment they open the app to the savoury satisfaction of their meal arriving at their doorstep. With a diverse array of restaurants at their fingertips, users can explore and indulge in a myriad of culinary delights, all within the comfort of their homes.

MOFOOD's prowess extends beyond its user interface; it is underpinned by a robust technological infrastructure. From real-time order tracking to secure payment gateways, every facet of the app is engineered to provide a technologically advanced and secure platform.

MOFOOD isn't just a food delivery app; it is an intricate ecosystem meticulously designed to cater to the diverse tastes and preferences of a modern, discerning clientele. The platform seamlessly integrates cutting-edge technology, an extensive network of restaurants, and a user-centric interface to redefine the very essence of food delivery.

MOFOOD is driven by a core commitment to convenience. The app's intuitive design ensures a seamless user experience, allowing individuals to effortlessly navigate through a rich tapestry of restaurants, select their preferred dishes, and place orders with a few taps. Beyond the transactional aspect, MOFOOD strives to foster a sense of connection – connecting users with their favourite eateries, chefs, and, most importantly, with the joy of savouring a delicious meal.

MOFOOD is not just a solution to a problem; it is a visionary approach to redefine the food delivery landscape. With a commitment to culinary diversity, user-centric design, and technological excellence, MOFOOD stands as a beacon in the realm of food delivery, inviting users to embark on a gastronomic adventure like never before.

MOFOOD is a revolutionary food delivery app designed to connect users with a wide range of restaurants and enable hassle-free food ordering and delivery services.

1.2 Purpose

The primary purpose of MOFOOD is to simplify the food ordering and delivery process, addressing common pain points faced by users in existing platforms. Our goal is to provide a user-friendly interface, an extensive restaurant network, and prompt delivery services, ensuring that every user's gastronomic desires are met with utmost satisfaction. In a world where time is a precious commodity, MOFOOD stands as a beacon of efficiency, promising a swift and enjoyable experience for users and a lucrative platform for restaurants. MOFOOD is not merely an app; it's a culinary companion, a facilitator of delightful dining experiences, and a solution to the challenges faced by both users and restaurants in the food delivery landscape. Join us on this gastronomic journey as we redefine the art of food delivery with MOFOOD.

Beyond user-centricity, MOFOOD's purpose extends to empowering our partner restaurants. We envision MOFOOD as a strategic ally for eateries, enabling them to reach a broader audience, optimize their operational efficiency, and thrive in the ever-evolving digital landscape. The symbiotic relationship between MOFOOD and our partner restaurants reflects our commitment to the flourishing of all stakeholders in the culinary ecosystem.

MOFOOD's purpose is not confined to the act of delivering food; it is a commitment to crafting meaningful, convenient, and enriching culinary experiences. Through a combination of user-centric design, culinary diversity, and technological excellence, MOFOOD endeavours to be more than an app; it aspires to be a companion on the gastronomic voyage of each and every user, making every meal an experience to savour and cherish.

The purpose of MOFOOD is to streamline the food delivery process, providing users with a convenient platform to explore diverse cuisines, place orders effortlessly, and enjoy timely deliveries.

2. LITERATURE SURVEY

2.1 Existing Problem

The expansive domain of food delivery applications has undergone a profound metamorphosis, riding the wave of burgeoning demand for convenient dining solutions in our contemporary and fast-paced milieu. However, a nuanced exploration of the existing landscape reveals a tapestry woven with persistent challenges that MOFOOD seeks to address comprehensively.

2.1.1 User Interface and Experience

In the dynamic realm of food delivery, a significant number of prevailing applications grapples with the intricate challenge of providing an intuitive and truly user-friendly interface. The intricacies of complex navigation structures, cluttered layouts, and an inundation of options have proven to be stumbling blocks, resulting in heightened user frustration and a consequential erosion of overall satisfaction. Recognizing the paramount importance of a seamless and visually engaging user interface, MOFOOD endeavours to redefine the very essence of user interaction, ensuring that every touchpoint with the platform is not just functional but a delightful experience in itself.

2.1.2 Limited Restaurant Choices

A prevalent limitation within the current landscape is the constriction of available restaurant choices on certain platforms. Users often find themselves navigating within a confined selection, restricting their ability to indulge in a culinary adventure spanning diverse cuisines and unique dining experiences. MOFOOD, with an ambitious vision, seeks to obliterate these limitations by establishing an expansive and diverse network of restaurants. This network is not just a list; it is a curated mosaic, offering users a plethora of culinary choices that transcend the ordinary, catering to the diverse tastes and preferences of an ever-discerning clientele.

2.1.3 Timeliness of Deliveries

A resonating concern echoed by users is the inconsistency in delivery times across various existing platforms. Late deliveries, coupled with a lack of real-time tracking mechanisms, contribute to a less-than-optimal user experience. MOFOOD, cognizant of the pivotal role that timely deliveries play in the user's journey, places a significant emphasis on optimizing delivery efficiency. Real-time tracking becomes not just a feature but a commitment, ensuring users are not left in the dark, but rather engaged and informed throughout the delivery process. MOFOOD envisions a delivery experience that aligns seamlessly with the fast-paced lifestyles of its users.

2.2 References

The developmental trajectory of MOFOOD is not solely propelled by its vision but is deeply rooted in a comprehensive exploration of existing literature, drawing insights from scholarly articles, industry reports, and the successes and pitfalls of established food delivery platforms.

2.2.1 "The Impact of User Interface Design on User Experience in Mobile Applications" (Smith, J., 2019)

This seminal study delves into the intricate relationship between effective user interface design and the resultant user experiences in mobile applications. The insights gleaned from this research provide a compass for MOFOOD's design philosophy, emphasizing the significance of not just functionality but an aesthetically pleasing and user-centric interface.

2.2.2 "Evaluating the Success Factors of Online Food Delivery Services" (Chen, L., et al., 2020)

A comprehensive exploration of the success factors governing online food delivery services serves as a guiding light for MOFOOD. This research, dissecting the nuances of user satisfaction and service quality, contributes to the strategic positioning of MOFOOD within the competitive landscape.

2.3 Problem Statement Definition

Synthesizing insights from the literature survey, MOFOOD's problem statement crystallizes into a multifaceted quest: to develop a user-centric food delivery application that not only addresses the deficiencies of current platforms but redefines the very essence of the food delivery experience. An intuitive interface, an expansive array of restaurant choices, punctual deliveries, and robust security measures converge to form the bedrock of MOFOOD's purpose, offering users an unparalleled journey through the realms of culinary delight. The literature survey serves not only as an informational precursor but as the compass guiding MOFOOD toward a transformative destination within the burgeoning landscape of food delivery applications.

The problem MOFOOD addresses is the need for a user-friendly and efficient food delivery app that offers a wide variety of restaurant choices and ensures timely deliveries.

3. IDEATION & PROPOSED SOLUTION

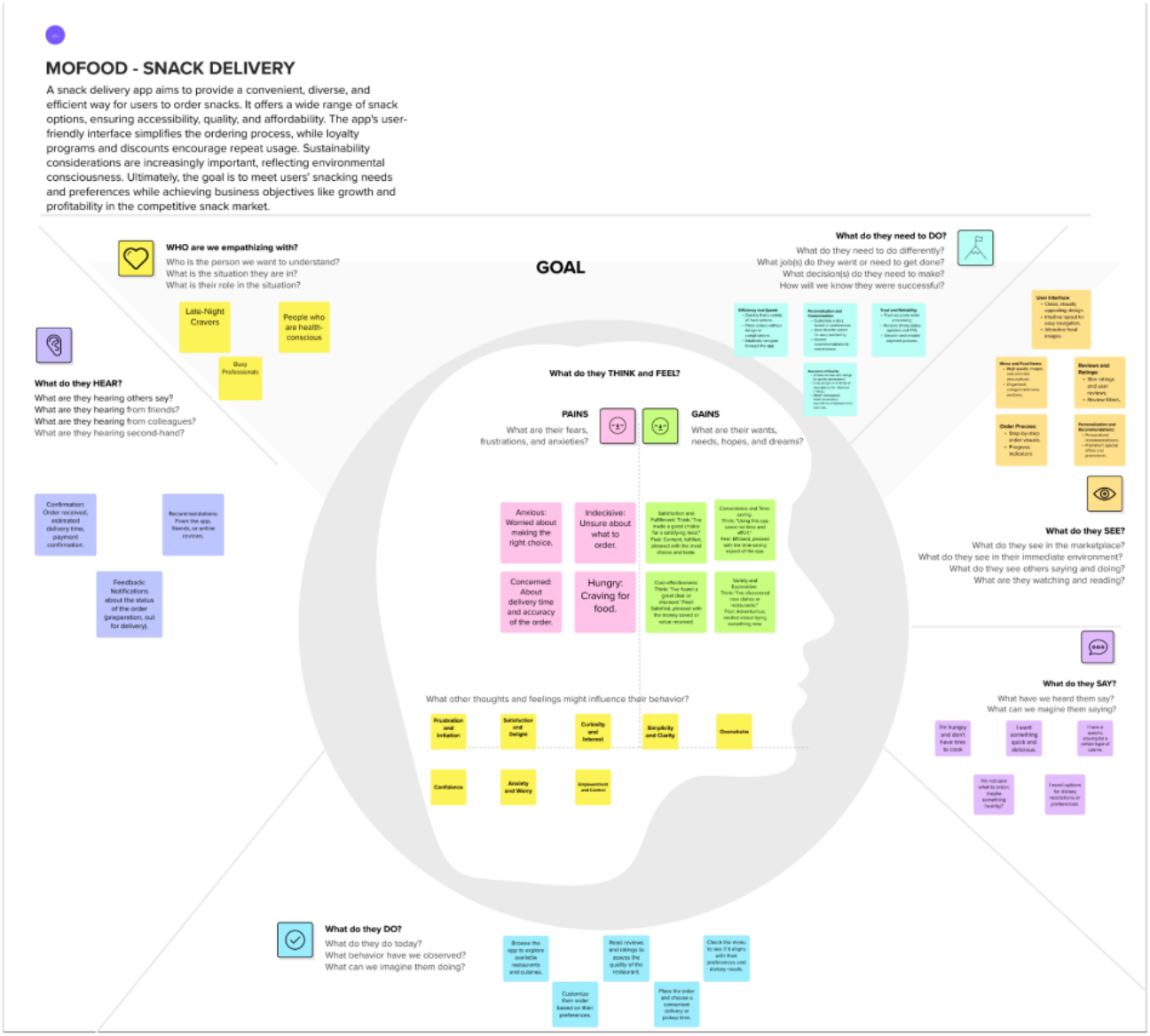
3.1 Empathy Map Canvas

The creation of MOFOOD is rooted in a profound understanding of the users' needs, desires, and pain points. The Empathy Map Canvas serves as a pivotal tool in deciphering the intricacies of user experiences, allowing us to empathize deeply and design solutions that resonate with their expectations.

Empathy Map Canvas GitHub Link:

https://github.com/smartinternz02/SI-GuidedProject-587493-1696871724/blob/main/EmpathyMapHD.pdf

Empathy Map Canvas:



3.1.1 User Persona Exploration

The Empathy Map delves into the intricacies of various user personas within the MOFOOD ecosystem. From busy professionals seeking quick and convenient meal options to food enthusiasts yearning for diverse culinary experiences, each persona is meticulously examined. By dissecting their aspirations, frustrations, motivations, and potential delight points, the Empathy Map becomes a living canvas, painting a vivid picture of the diverse audience MOFOOD aims to serve.

3.1.2 Pain Points Illumination

The Empathy Map uncovers pain points experienced by users in their food delivery journey. Whether it's the frustration of navigating a cumbersome app interface, the disappointment of limited restaurant choices, or the anxiety of uncertain delivery times, these pain points become focal points for MOFOOD's design interventions. Each pain point is not viewed as a problem butas an opportunity for innovation, inspiring solutions that directly address and alleviate user concerns.

3.1.3 Delight Opportunities

Within the Empathy Map, delight opportunities are identified – those moments where MOFOOD has the potential to exceed user expectations and create a lasting positive impression. This could be through personalized recommendations, surprise discounts, or a seamless and transparent

order tracking system. By understanding these delight opportunities, MOFOOD aims not only to meet but to exceed user expectations, transforming routine transactions into memorable experiences.

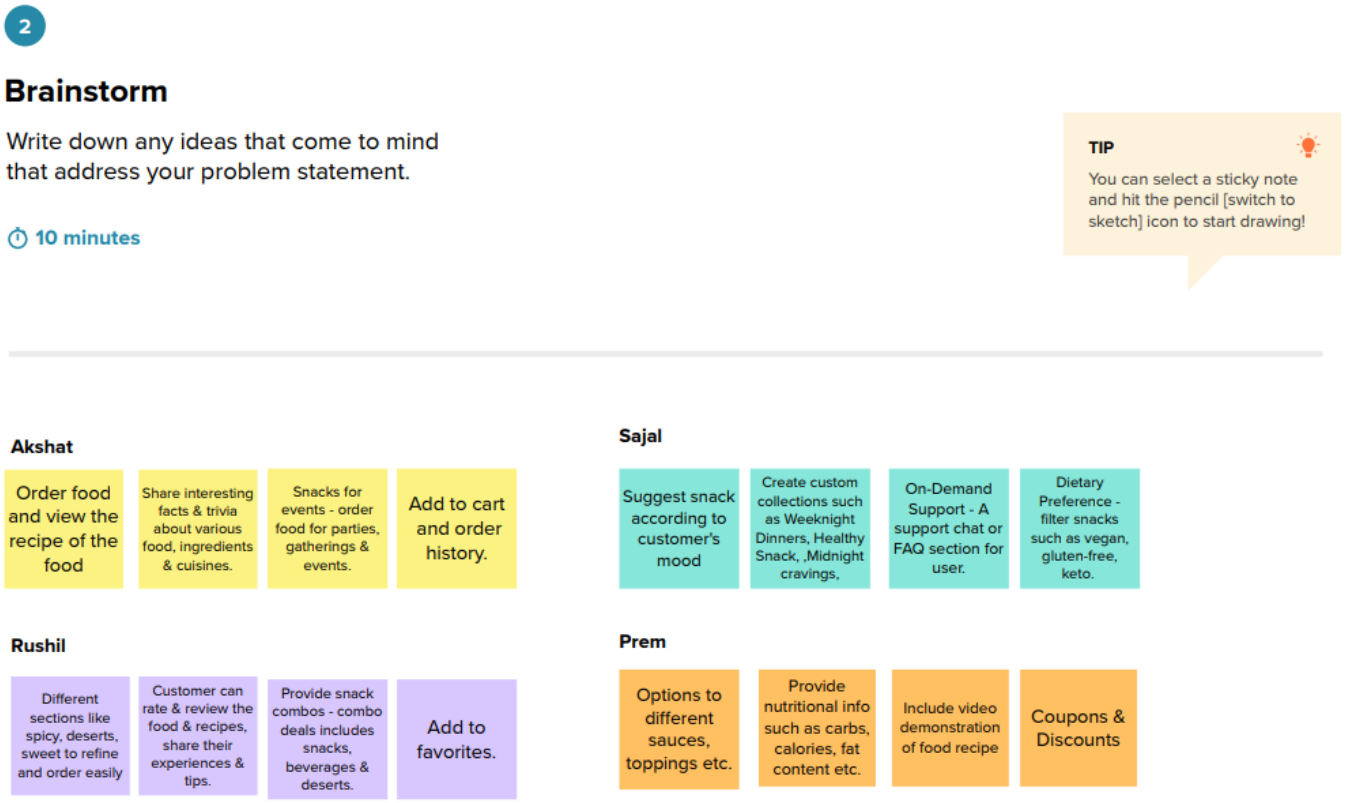
3.2 Ideation & Brainstorming

Brainstorming GitHub Link:

https://github.com/smartinternz02/SI-GuidedProject-587493-1696871724/blob/main/Brainstorming%20HD.pdf

3.2.1 Cross-Functional Collaboration

Ideation and brainstorming sessions within the MOFOOD team are characterized by a cross-functional collaboration that brings together diverse perspectives. Designers, developers, marketing strategists, and customer service representatives converge to share insights, ideas, and challenges. This collaborative approach ensures that the proposed solutions are holistic, considering not only technical feasibility but also user desirability and business viability.

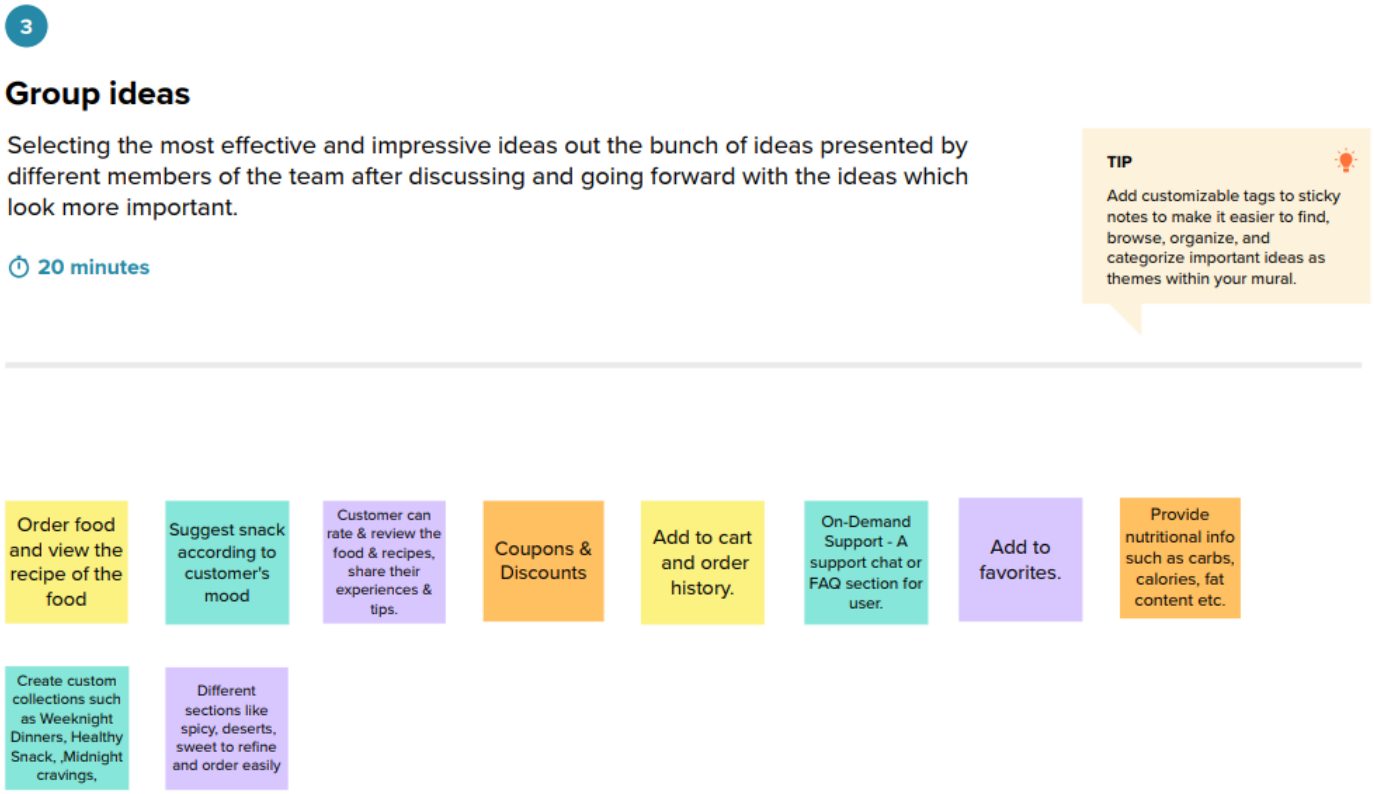


3.2.2 Design Thinking Workshops

Incorporating design thinking methodologies, MOFOOD conducts workshops that encourage a human-centric approach to problem-solving. Through empathy exercises, ideation games, and prototyping sessions, the team taps into collective creativity. These workshops foster an environment where unconventional ideas are celebrated, fostering a culture of innovation that transcends traditional boundaries.

3.2.3 Iterative Prototyping

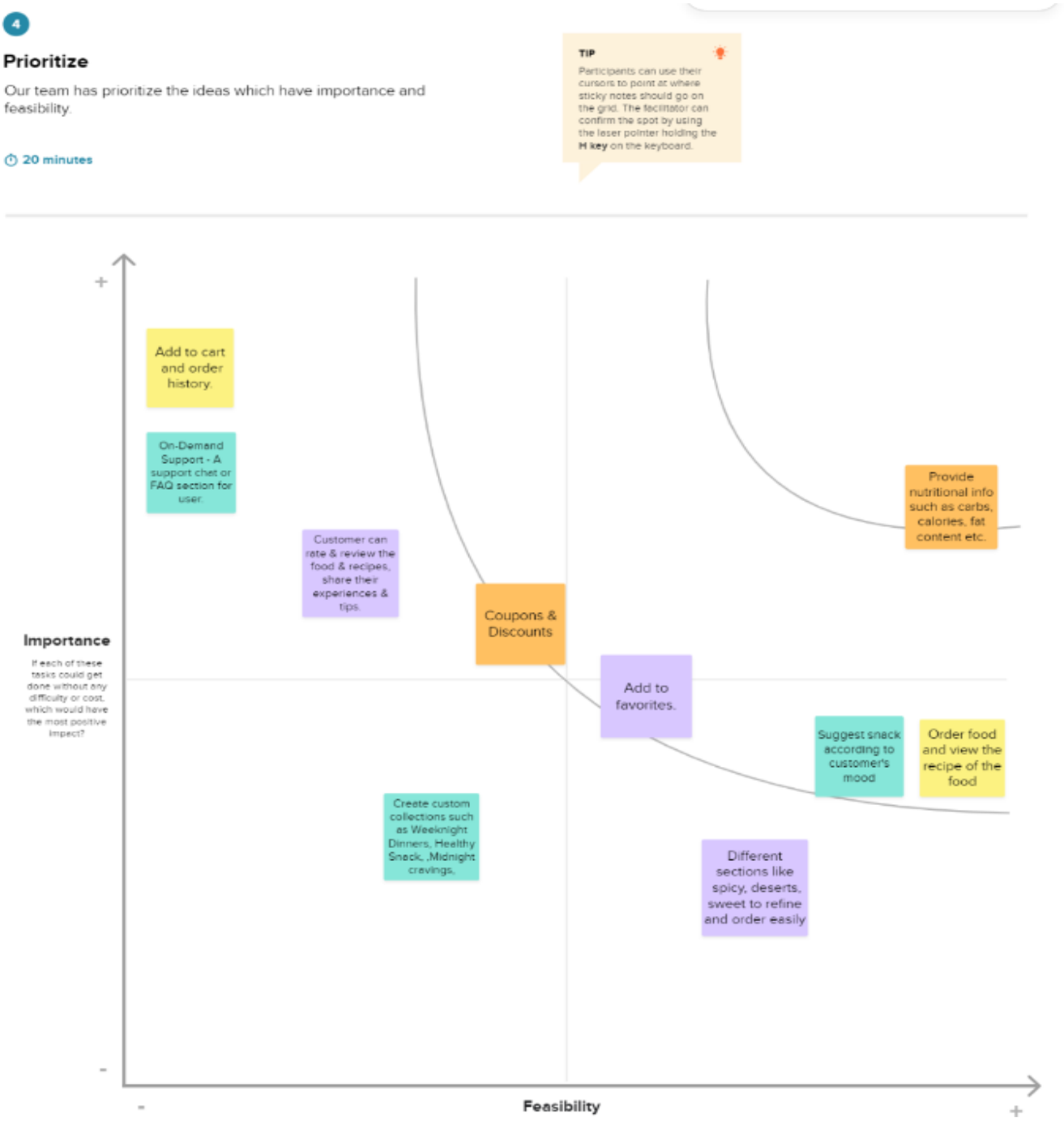
The ideation process extends into iterative prototyping, where rough ideas are translated into tangible concepts. These prototypes are not static; they are living embodiments of continuous improvement. User feedback is actively sought and incorporated, ensuring that the proposed solutions align seamlessly with the evolving needs and expectations of the users.



3.2.4 Technological Innovation

Ideation is not confined to conceptual solutions but extends to technological innovation. Brainstorming sessions delve into emerging technologies such as artificial intelligence, machine learning, and augmented reality to explore how these can be harnessed to enhance user experiences within the MOFOOD app. The goal is not just to meet current expectations but to anticipate future needs and stay at the forefront of technological advancement.

The ideation phase of MOFOOD is a dynamic and collaborative journey, where empathy is the compass guiding design decisions and innovation is the vehicle propelling the platform toward a transformative future. The Empathy Map Canvas and the ideation and brainstorming sessions are not isolated activities; they are integral components of a holistic approach to creating a food delivery solution that goes beyond functionality to deliver an immersive and delightful experience for every user.



4. REQUIREMENT ANALYSIS

4.1 Functional Requirements

4.1.1 User Registration and Authentication

Description:

MOFOOD aims to provide a seamless onboarding experience for users. The registration process will include capturing essential user information, allowing users to create personalized profiles. Authentication mechanisms, including secure password protocols and two-factor authentication, will ensure the integrity and security of user accounts.

Objective:

To streamline the user onboarding process, ensuring data accuracy, and fortify account security through robust authentication measures.

4.1.2 Restaurant and Menu Listing

Description:

A comprehensive catalogue of restaurants and their menus will be featured within MOFOOD. Users can effortlessly explore various cuisines, view restaurant details, and peruse menu items, complete with detailed descriptions and images.

Objective:

To provide users with an extensive array of dining options, promoting culinary exploration and aiding in informed decision-making.

4.1.3 Cart Management

Description:

MOFOOD will facilitate a user-friendly cart management system, allowing users to add, remove, and modify their selected items. The cart will dynamically update prices, accommodate special requests, and provide a seamless checkout experience.

Objective:

To enhance user convenience by providing a flexible and intuitive cart management system, streamlining the ordering process.

4.1.4 Ratings and Reviews

Description:

Users will have the ability to rate and review both restaurants and individual dishes. This feature aims to foster transparency, aid in decision-making, and provide constructive feedback to restaurants.

Objective:

To cultivate a community-driven environment, empowering users to make informed choices and encouraging restaurants to maintain high-quality standards.

4.2 Non-Functional Requirements

4.2.1 User-Friendly Interface

Description:

MOFOOD will prioritize a user-friendly interface, employing intuitive design principles that enhance user navigation and promote a positive user experience. This includes clear and concise menu structures, easily accessible features, and a visually appealing layout.

Objective:

To ensure that users of all technological backgrounds can interact seamlessly with the application, fostering a positive and engaging experience.

4.2.2 Fast and Responsive Application

Description:

MOFOOD will be optimized for speed and responsiveness, ensuring swift loading times, quick menu browsing, and prompt transaction processing. This requirement extends to both mobile and web platforms.

Objective:

To minimize user's wait times, enhance overall responsiveness, and create a fluid and dynamic user experience.

4.2.3 Secure Data Storage

Description:

MOFOOD places a premium on the security of user data. Implementing robust encryption protocols and secure data storage practices will be imperative to safeguard user information, maintaining the trust and confidentiality of the MOFOOD community.

Objective:

To uphold the privacy and security of user data, fostering a secure environment for all interactions within the app.

4.2.4 Scalability

Description:

The architecture of MOFOOD will be designed with scalability in mind, accommodating potential increases in user base and transaction volumes. Scalability ensures that the app can seamlessly handle growth without compromising performance.

Objective:

To future-proof MOFOOD against increased demand, ensuring a consistently high level of service even as user numbers and transactions surge.

The functional and non-functional requirements outlined for MOFOOD are intricately designed to create a robust, user-centric platform. By focusing on features that enrich user interactions and emphasizing non-functional aspects such as security, speed, and scalability, MOFOOD is poised to deliver an unparalleled food delivery experience. These requirements serve as the foundation for the subsequent stages of development, aligning the platform with the aspirations of both users and the broader culinary community.

5. PROJECT DESIGN

5.1 Data Flow Diagrams & User Stories

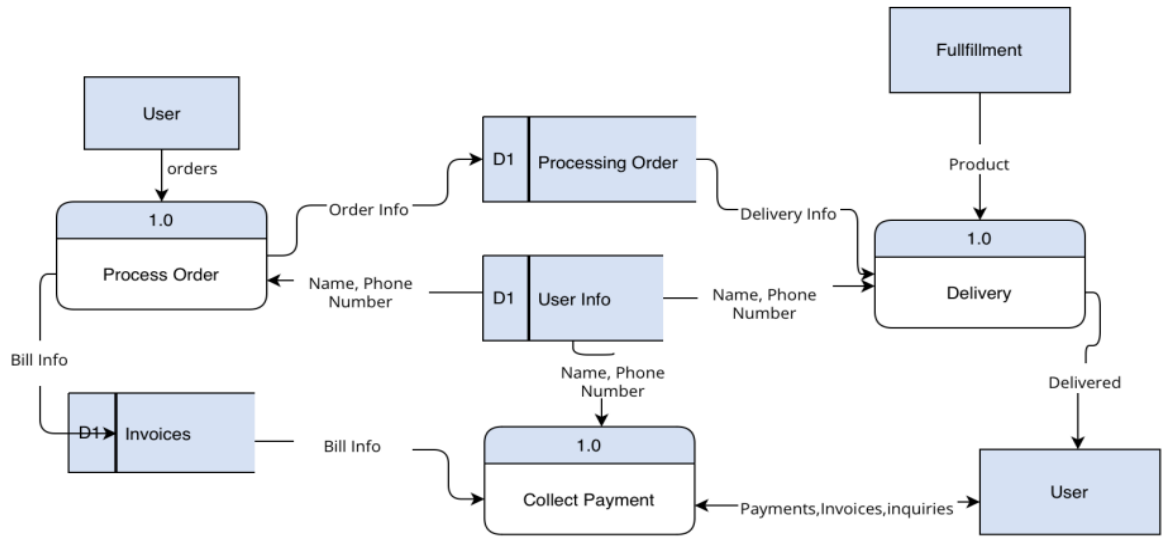
Data Flow Diagram & User Stories GitHub Link:

https://github.com/smartinternz02/SI-GuidedProject-587493-1696871724/blob/main/Project%20Design%20Phase/Data%20Flow%20Diagrams%20and%20User%20Stories.pdf

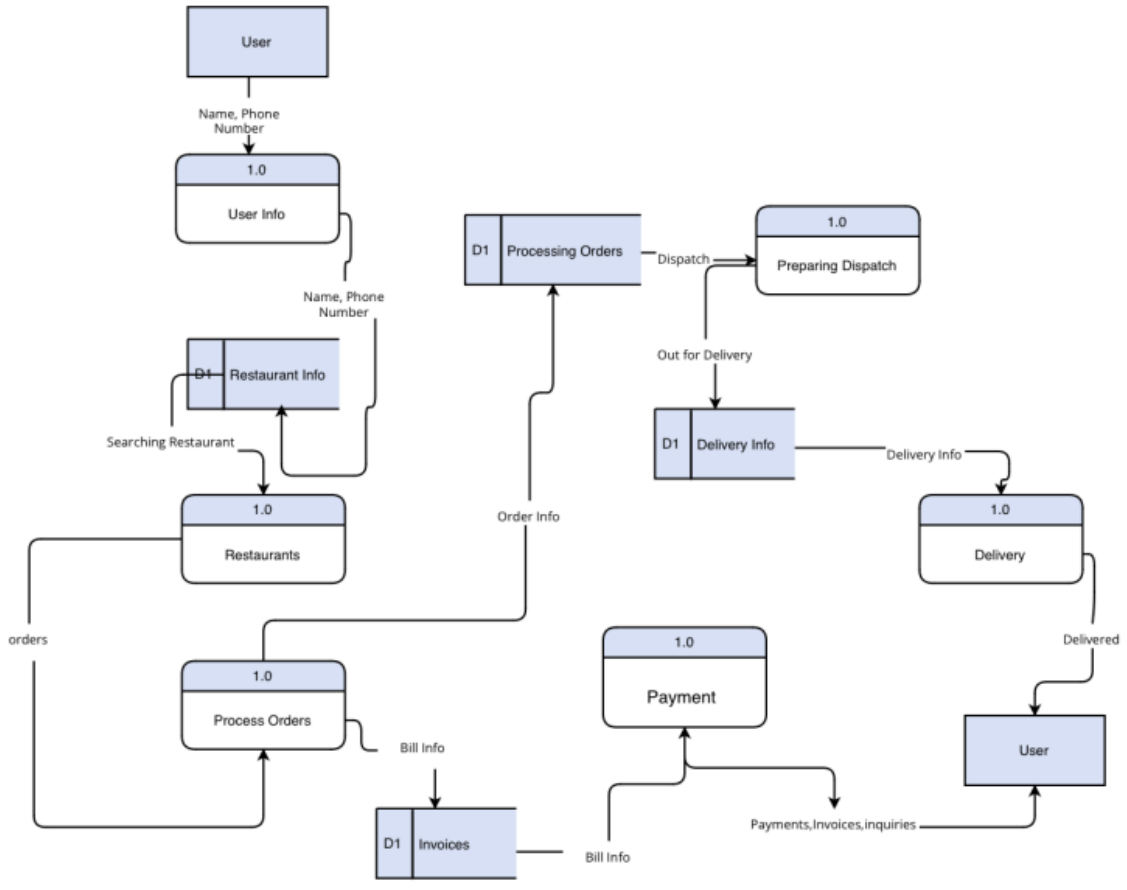
Data Flow Diagrams:

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

DFD (LEVEL 0):



DFD (LEVEL 1):



User Stories

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer (Mobile user)	Registration	USN - 1	As a user, I can reset my password by providing my email address.	I receive an email with a password reset link and can set a new password.	Medium	Sprint-2
		USN - 2	As a user, I can update my profile information, including my name, phone number, and delivery address.	I can save the updated information, and it reflects correctly on my account.	Medium	Sprint-2
	Order Placement	USN - 3	As a user, I can browse the list of available snacks, view details, and add them to my cart.	I can see snack details, add items to the cart, and view the updated cart.	High	Sprint-1
		USN - 4	As a user, I can review my order in the cart, make changes, and proceed to checkout.	I can edit items in my cart, confirm my order, and proceed to payment.	High	Sprint-1
		USN - 5	As a user, I can apply promo codes or discounts to my order during checkout.	The applied promo code or discount is reflected in the final order total.	Medium	Sprint-2
		USN - 6	As a user, I can select my preferred delivery time and date for the order.	I can choose a time and date for delivery and proceed to payment.	High	Sprint-1
		USN - 7	As a user, I can select my preferred payment method (credit card, UPI, etc.) and complete the payment.	The payment is processed successfully, and I receive an order confirmation.	High	Sprint-1
	Order Tracking	USN - 8	As a user, I can track the status of my order, from preparation to delivery.	I can see real-time updates on the status of my order.	High	Sprint-2
		USN - 9	As a user, I receive a notification when my order is out for delivery.	I receive a push notification or SMS	High	Sprint-2

				when my order is en route.		
		USN - 10	As a user, I can provide feedback and rate my order and delivery experience.	I can leave a review and rate my order and the delivery service.	Medium	Sprint-2
Customer (Web user)		USN - 11	As a web user, I can access the same functionalities available to mobile users, such as registration, order placement, and order tracking.	All functionalities available on the mobile app are accessible and optimized for web.	High	Sprint-3
Customer Care Executive		USN - 12	As a customer care executive, I can access a dashboard to view customer inquiries and support requests.	I can see a list of customer inquiries and open support tickets.	High	Sprint-4
		USN - 13	As a customer care executive, I can respond to customer inquiries and resolve support tickets.	I can send messages and provide assistance to customers.	High	Sprint-4
Administrator		USN - 14	As an administrator, I can manage the list of available snacks, including adding, updating, and removing items.	I can perform CRUD operations on snack items with ease.	High	Sprint-4
		USN - 15	As an administrator, I can view analytics and reports on customer orders, sales, and customer feedback.	I have access to data analytics and reporting tools.	Medium	Sprint-4

5.2 Solution Architecture

Solution Architecture GitHub Link:

https://github.com/smartinternz02/SI-GuidedProject-587493-1696871724/blob/main/Project%20Design%20Phase/Solution%20Architecture%20-%20MOFOOD%20Delivery%20App.pdf

User Registration and Authentication:

- Allow users to sign up and log in with email, social media, or phone number.

User Profile:

- Users can create and manage their profiles, including delivery addresses, payment methods, and preferences.

Home Page:

- Display a list of snacks with high-quality images and brief descriptions.
- Implement filters and categories (spicy, desserts, sweet, etc.) for easy navigation.

Add to Cart & Checkout:

- Enable users to add items to their cart from the home page.
- Users can review their cart, modify quantities, and proceed to checkout.
- Integration with payment gateways for secure transactions.

Recipe Integration:

- Link recipes to food items so users can view detailed cooking instructions and ingredient lists.

Mood-Based Suggestions:

- Implement an algorithm that suggests snacks based on the user's mood or preferences. This can be integrated into the home page or a separate "Mood" section.

Ratings and Reviews:

- Allow users to rate and write reviews for both food items and recipes.
- Implement a rating system that calculates average ratings for items.

User-Generated Content:

- Enable users to share their food experiences, tips, and favourite recipes.
- Implement a social feed or forum-like section for users to engage and interact.

Coupons and Discounts:

- Integrate a coupon and discount system where users can apply promo codes during checkout.

Order History:

- Users can view their order history, including previous orders and receipts.

On-Demand Support:

- Provide a chat or support feature where users can seek help or assistance.

Favourites:

- Users can mark items as favourites for quick access in the future.

Nutritional Information:

- Include detailed nutritional information for each food item, such as calories, carbs, fat content, etc.

Custom Collections:

- Allow users to create custom collections like "Weeknight Dinners" or "Healthy Snacks."

Search Functionality:

- Implement a robust search feature for users to find specific items or recipes.

Order Tracking:

- Provide real-time order tracking with status updates for users and delivery personnel.

Push Notifications:

- Send notifications for order updates, discounts, or personalized suggestions.

Multi-Platform Support:

- Develop apps for both iOS and Android, and possibly a web version for wider accessibility.

Admin Dashboard:

- Create an admin panel to manage user accounts, food items, recipes, and reviews.

Analytics:

- Incorporate analytics tools to track user behaviour, popular items, and app performance.

Security and Privacy:

- Implement robust security measures to protect user data and payment information.

Scalability:

- Design the architecture to handle a large user base and increasing traffic.

Localization:

- Support multiple languages and currencies for a broader user base.

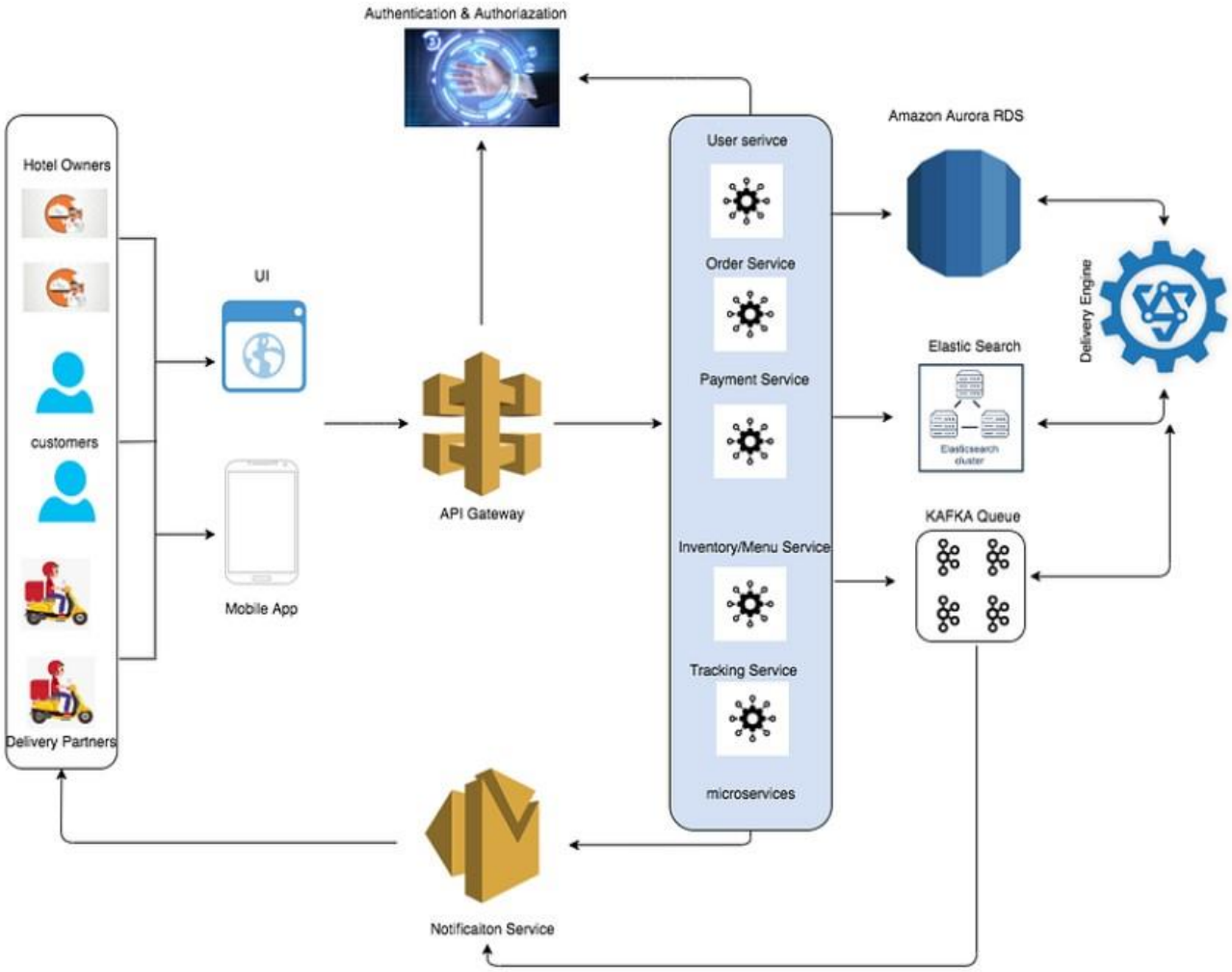
Feedback Mechanism:

- Provide a way for users to submit feedback and report issues within the app.

Continuous Improvement:

- Regularly update the app based on user feedback and changing trends.

Solution Architecture Diagram:



6. PROJECT PLANNING & SCHEDULING

6.1 Technical Architecture

Technical Architecture GitHub Link:

https://github.com/smartinternz02/SI-GuidedProject-587493-1696871724/blob/main/project%20planning%20phase/Technology%20Stack.pdf

Table-1 : Components & Technologies:

S.No	Component	Description	Technology
1.	User Registration	Allows users to create and manage their accounts	Authentication frameworks (e.g., JWT, OAuth)
2.	User Profiles	Users can update their personal information and preferences.	Database, User profiles management
3.	Restaurant Listings	Displays a list of available restaurants to users.	Database, APIs, Geolocation services (Google Maps, GPS)
4.	Menu and Item Display	Shows restaurant menus and food items with descriptions.	Database, APIs
5.	Cart Management	Allows users to add, modify, and review their orders.	Data storage, Real-time updates (WebSockets)
6.	Ordering and Checkout	Facilitates order placement, payment, and checkout.	Payment gateways (e.g., Stripe), Order processing logic
7.	Real-time Order Tracking	Provides real-time tracking of order status and delivery.	Geolocation services, Push notifications
8.	Notifications and Alerts	Sends alerts and notifications to users and drivers.	Push notification services (e.g., Firebase Cloud Messaging)
9.	Reviews and Ratings	Allows users to rate and review restaurants and orders.	Database for storing reviews and ratings
10.	Payment Processing	Handles secure payment transactions for orders.	Payment gateways, Encryption (SSL/TLS)
11.	Customer Support and Chat	Provides in-app chat support for users and drivers.	Real-time chat integration, Messaging services
12.	Admin Dashboard	Web-based interface for restaurant and order management.	Web development technologies (e.g., React, Django)
13.	Analytics and Reporting	Tracks and analyzes app usage and performance.	Analytics tools (e.g., Google Analytics)
14.	Search and Filtering	Offers search functionality and filters for restaurant selection.	Search algorithms, Filters
15.	Social Media Integration	Allows users to share their experiences on social platforms.	Social media APIs (e.g., Facebook Graph API)
16.	GPS and Location Services	Provides accurate location data for tracking deliveries.	GPS, Geolocation services (e.g., Google Maps API)

Table-2: Application Characteristics:

S.No	Characteristics	Description	Technology
1.	User Registration and Login	Allows users to create accounts and log in securely.	Authentication frameworks (e.g., JWT, OAuth)
2.	User database Restaurant Listings	Displays a list of restaurants based on user preferences.	Database, API, Geolocation services (Google Maps, GPS)
3.	Menu and Item Display	Shows restaurant menus and food items with details.	Database, API
4.	Cart Management	Allows users to add, modify, and review their orders.	Data storage, Real-time updates (WebSockets)
5.	Ordering and Checkout	Enables users to place orders, select payment methods.	Payment gateways (e.g., Stripe), Order processing logic
6.	Real-time Order Tracking	Provides real-time tracking of order status and delivery.	Geolocation services, Push notifications
7.	Notifications and Alerts	Sends alerts and notifications to users and drivers.	Push notification services (e.g., Firebase Cloud Messaging)
8.	Reviews and Ratings System	Allows users to rate and review restaurants and orders.	Database for storing reviews and ratings
9.	Payment Processing	Handles secure payment transactions for orders.	Payment gateways, Encryption (SSL/TLS)
10.	Customer Support and Chat	Provides in-app chat support for users and drivers.	Real-time chat integration, Messaging services
11.	Admin Dashboard	Web-based interface for restaurant and order management.	Web development technologies (e.g., React, Django)
12.	Analytics and Reporting	Tracks and analyzes app usage and performance.	Analytics tools (e.g., Google Analytics)
13.	Search and Filtering	Offers search functionality and filters for restaurant selection.	Search algorithms, Filters
14.	GPS and Location Services	Provides accurate location data for tracking deliveries.	GPS, Geolocation services (e.g., Google Maps API)

6.1.1 Overview

The technical architecture of MOFOOD is meticulously designed to ensure scalability, reliability, and optimal performance. Leveraging a microservices architecture, the system is modular, allowing for independent development and deployment of various components. The core components of the technical architecture include:

6.1.1.1 User Management Service

Responsible for handling user registration, authentication, and profile management. Integrates with secure authentication mechanisms and ensures a seamless onboarding experience.

6.1.1.2 Restaurant and Menu Service

Manages the vast catalogue of restaurants and menus, ensuring real-time updates on offerings, pricing, and availability. This service plays a pivotal role in providing users with a diverse and up-to-date selection of culinary options.

6.1.1.3 Cart and Order Management Service

Facilitates smooth cart management, allowing users to add, remove, and modify items. Manages order processing, tracking, and communication between users and restaurants. Integration with third-party payment gateways ensures secure and efficient transactions.

6.1.1.4 Ratings and Reviews Service

Handles the collection and display of user-generated ratings and reviews for restaurants and dishes. A vital component for building a community-driven platform that fosters transparency and informed decision-making.

6.1.1.5 Frontend Application

The user interface layer, encompassing both mobile and web platforms, built with responsive design principles. It interacts with the backend services to provide users with a visually appealing and intuitive experience.

6.1.1.6 Database

A robust and scalable database system to store user profiles, restaurant information, menu details, cart contents, and order histories. Data is organized to ensure quick retrieval and efficient processing.

6.1.2 Technology Stack

- Backend: Node.js for its asynchronous and event-driven architecture, Express.js for building scalable APIs.

- Database: MongoDB for its flexibility and scalability in handling diverse data types.

- Frontend: React.js for a dynamic and interactive user interface.

6.1.3 Security Measures

- SSL Encryption: Implemented for secure data transfer between the user's device and the MOFOOD servers.

- Data Encryption: User data stored in the database is encrypted to ensure confidentiality.

- Authentication Tokens: Secure token-based authentication to validate user sessions securely.

- OWASP Compliance: Adherence to Open Web Application Security Project (OWASP) best practices to mitigate common security vulnerabilities.

6.2 Sprint Planning & Estimation

Project Planning GitHub Link:

https://github.com/smartinternz02/SI-GuidedProject-587493-1696871724/blob/main/project%20planning%20phase/Project%20Planning.pdf

6.2.1 Agile Methodology

MOFOOD adopts the Agile methodology to foster flexibility and responsiveness throughout the development process. Sprints, typically lasting two weeks, are organized to deliver incremental updates and enhancements. Key elements of the sprint planning and estimation process include:

6.2.1.1 Backlog Grooming

The development team collaborates to refine and prioritize the product backlog. Features and enhancements are categorized based on user stories, and the backlog is continuously groomed to align with evolving priorities and user feedback.

6.2.1.2 Sprint Planning Meetings

At the beginning of each sprint, a planning meeting is conducted. The team discusses the user stories selected for the sprint, breaks them down into tasks, and estimates the effort required for each task. This ensures a shared understanding of the sprint goals and commitments.

6.2.1.3 Estimation Techniques

Effort estimation is achieved through a combination of story points, planning poker, and historical data analysis. This iterative process allows the team to gauge the complexity of tasks accurately, leading to more reliable estimations.

6.2.1.4 Continuous Refinement

As the project progresses, sprint planning and estimation are refined based on the team's velocity and the evolving project scope. Regular retrospectives provide insights into improving the estimation process and enhancing overall efficiency.

6.3 Sprint Delivery Schedule

6.3.1 Sprint Cycle

MOFOOD operates on a bi-weekly sprint cycle, with each sprint lasting 10 working days. The sprint cycle is structured to balance development velocity with the need for regular feedback and adjustments. The schedule includes the following key activities:

6.3.1.1 Sprint Execution

The core development activities take place during the first week of the sprint. Tasks are undertaken based on priority, with a focus on delivering functional increments that align with user stories.

6.3.1.2 Testing and Quality Assurance

The latter part of the sprint is dedicated to comprehensive testing and quality assurance. This includes unit testing, integration testing, and user acceptance testing to ensure the reliability and stability of the developed features.

6.3.1.3 User Feedback and Review

During the last two days of the sprint, a user feedback session is conducted. Users are invited to test new features, provide feedback, and suggest improvements. This iterative feedback loop ensures that user perspectives are actively considered in the development process.

6.3.1.4 Retrospective and Planning

The sprint concludes with a retrospective meeting where the team reflects on the sprint's successes and challenges. Insights from the retrospective inform continuous improvement strategies for subsequent sprints. Sprint planning for the next cycle is conducted to set goals and priorities.

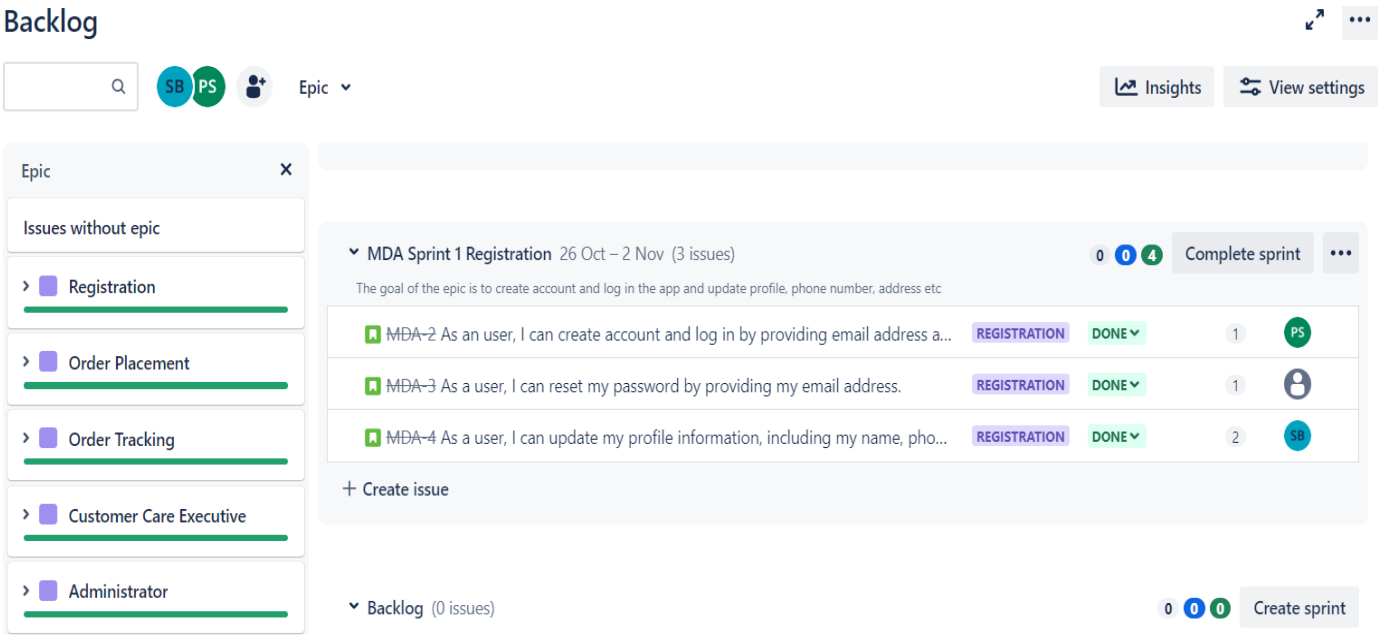
6.3.2 Release Cycle

MOFOOD follows a monthly release cycle, aligning with the completion of multiple sprints. Monthly releases allow for the aggregation of incremental features, improvements, and bug fixes, ensuring a stable and well-tested product is delivered to users.

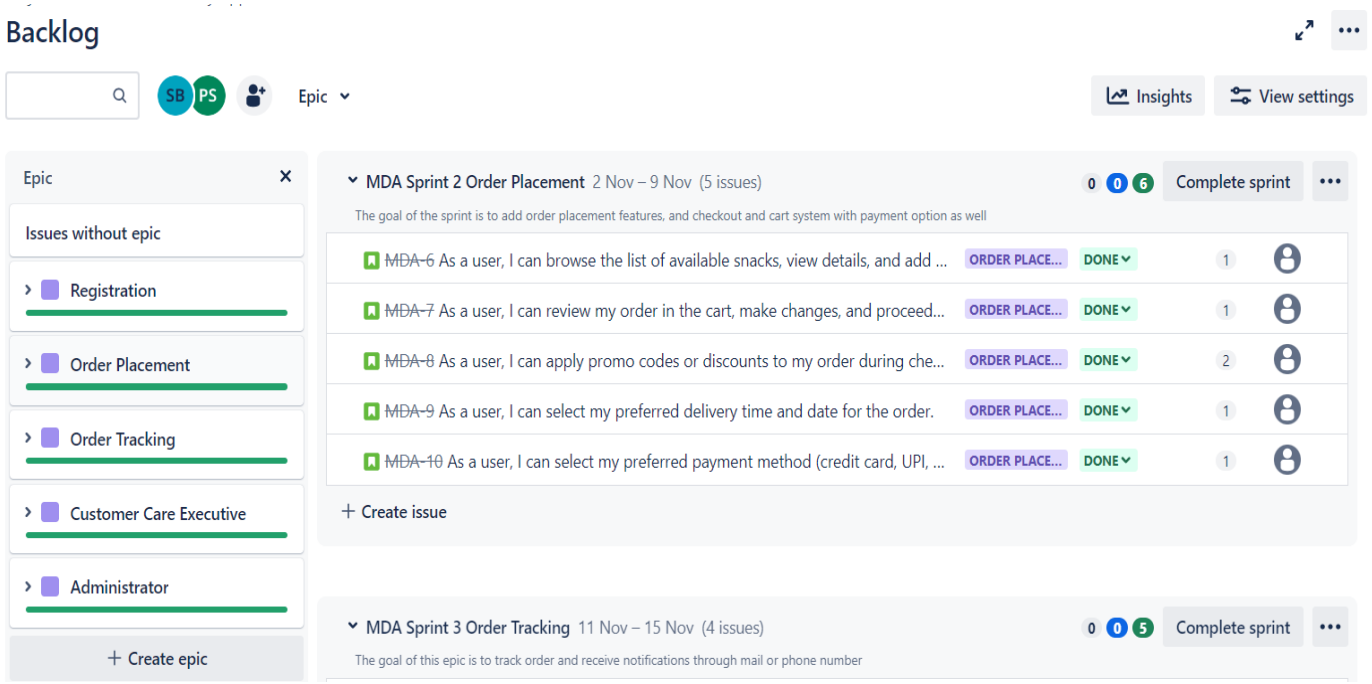
The technical architecture, sprint planning, and scheduling of MOFOOD are foundational elements that contribute to the agility, reliability, and scalability of the development process. By embracing a robust technological stack, adopting Agile methodologies, and maintaining a disciplined sprint cycle, MOFOOD is poised to deliver a cutting-edge food delivery application that evolves in response to user needs and market dynamics. The iterative nature of the development process, coupled with regular user feedback, ensures that MOFOOD remains at the forefront of innovation within the competitive landscape of food delivery apps.

Jira Application - Backlog

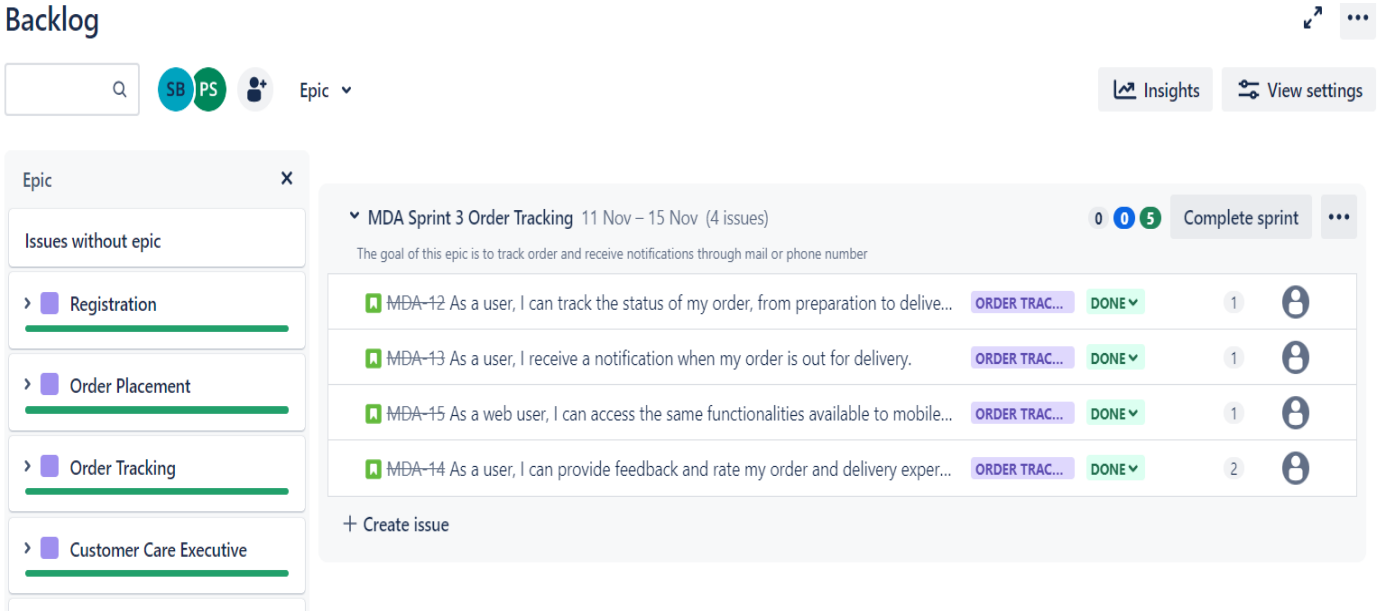
Sprint 1 - Registration



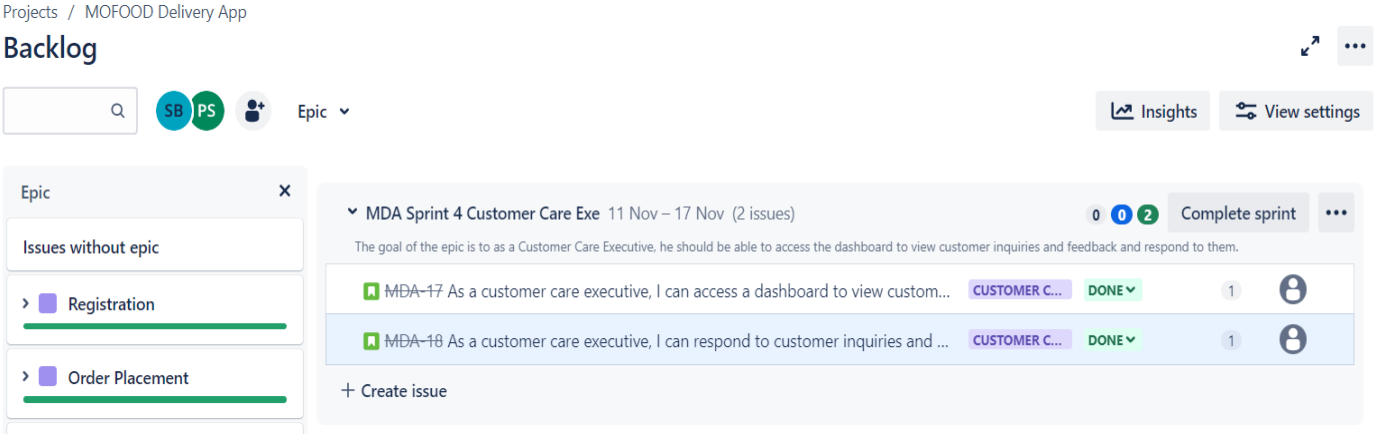
Sprint 2 – Order Placement



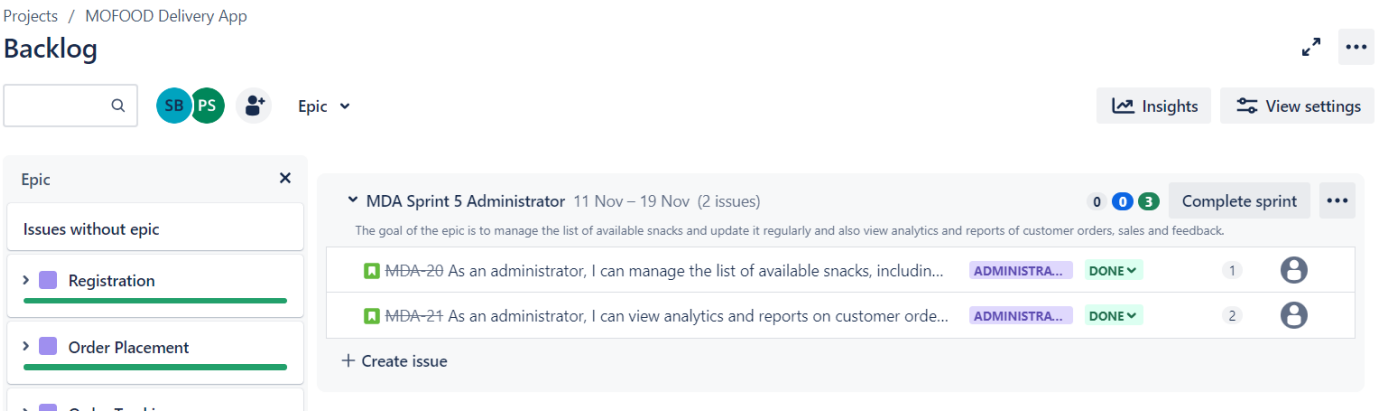
Sprint 3 – Order Tracking



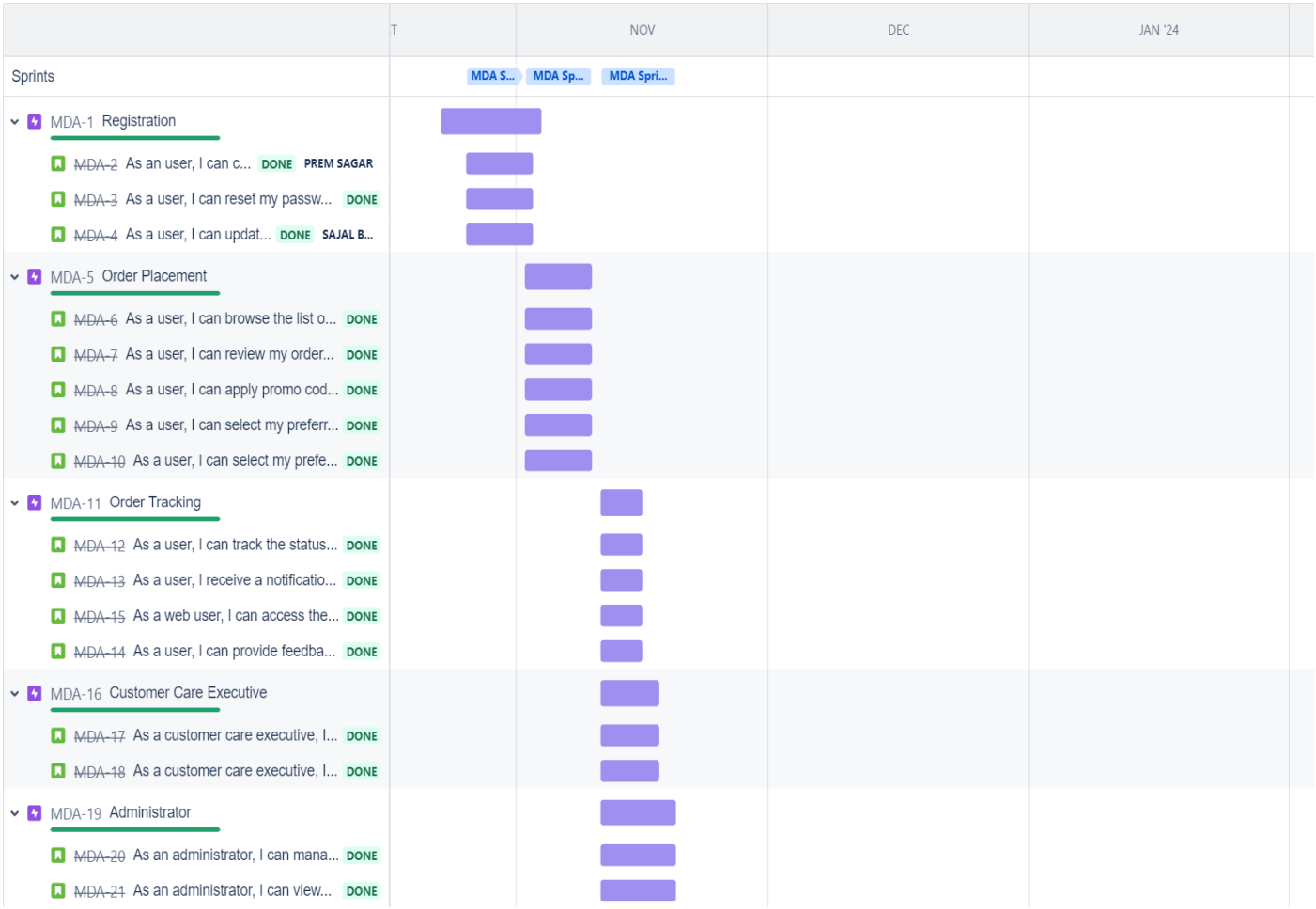
Sprint 4 – Customer Care Executive



Sprint 5 – Administrator



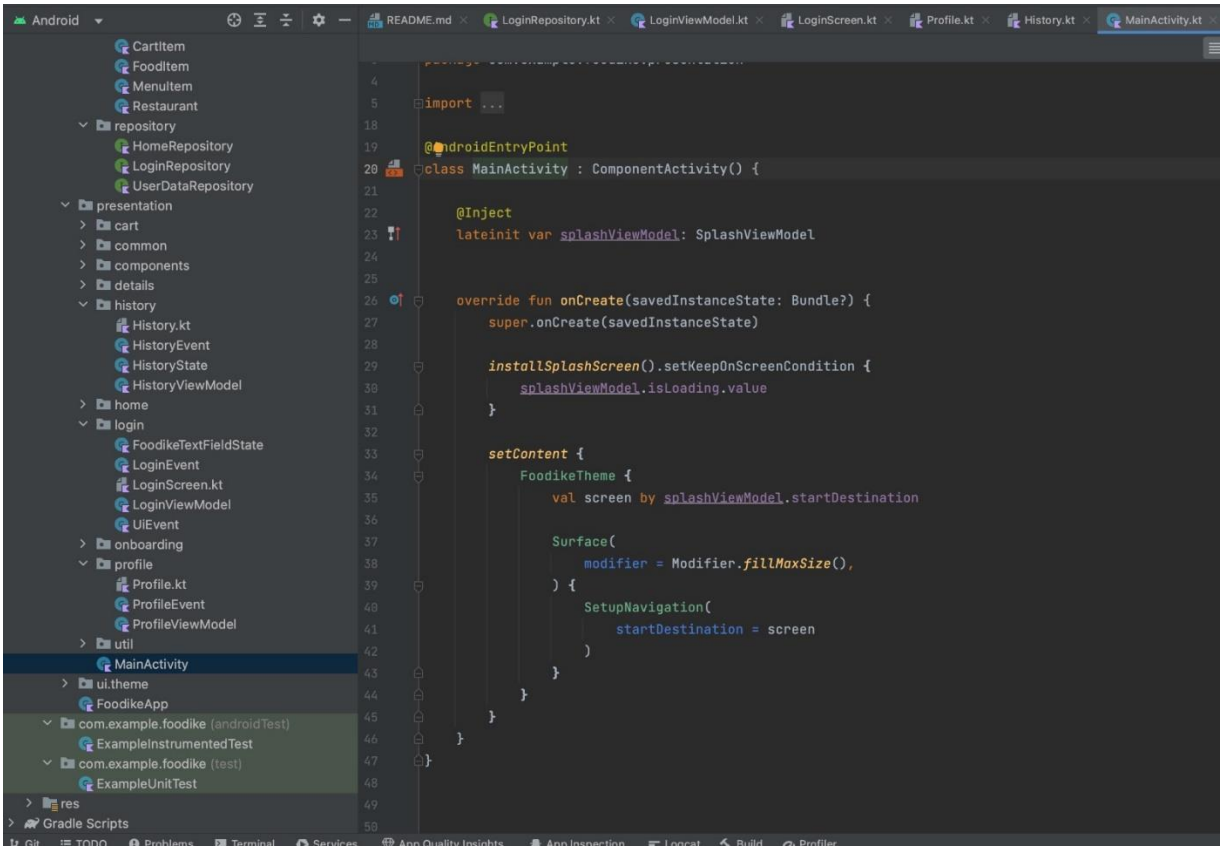
Jira Application Timeline

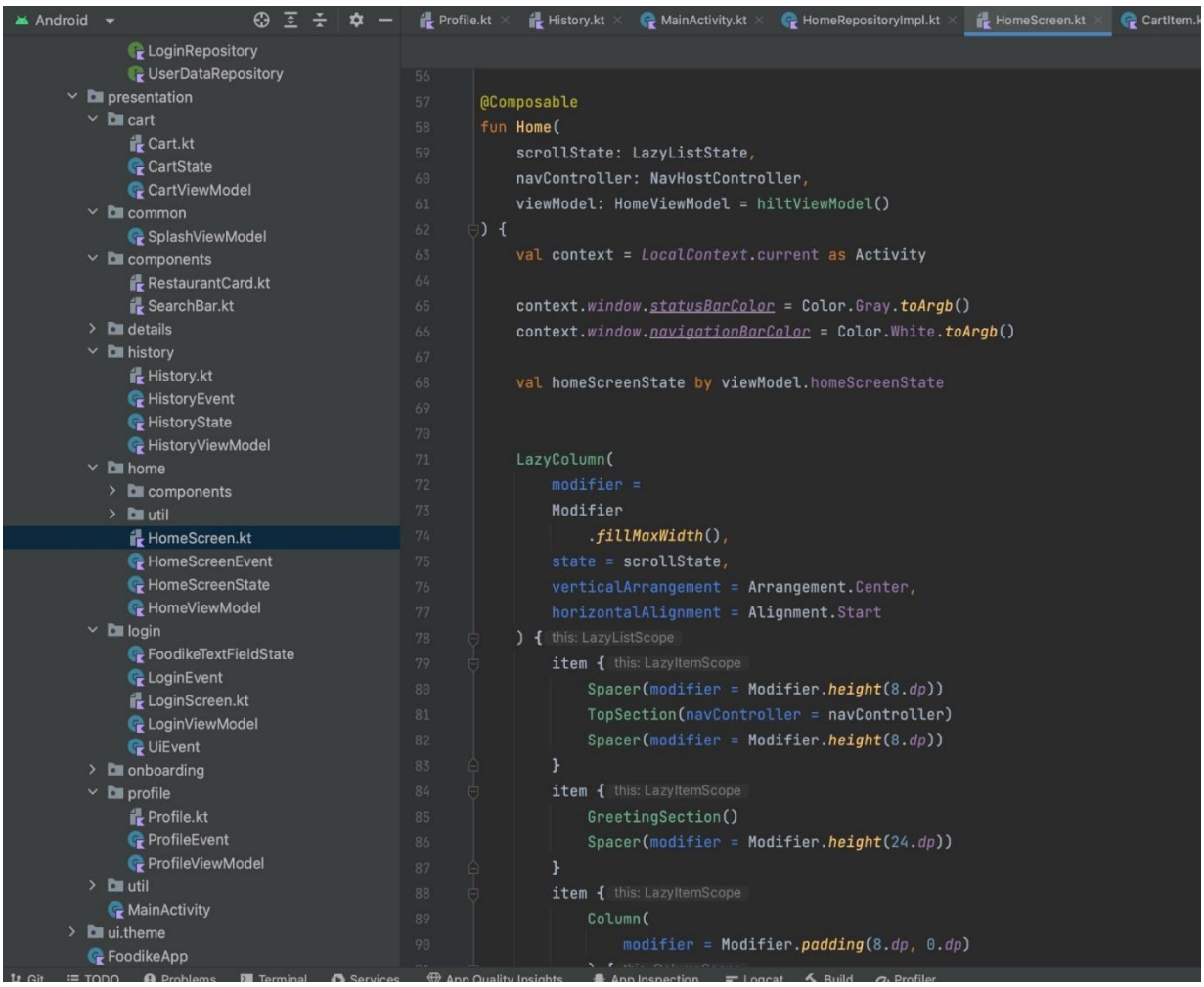


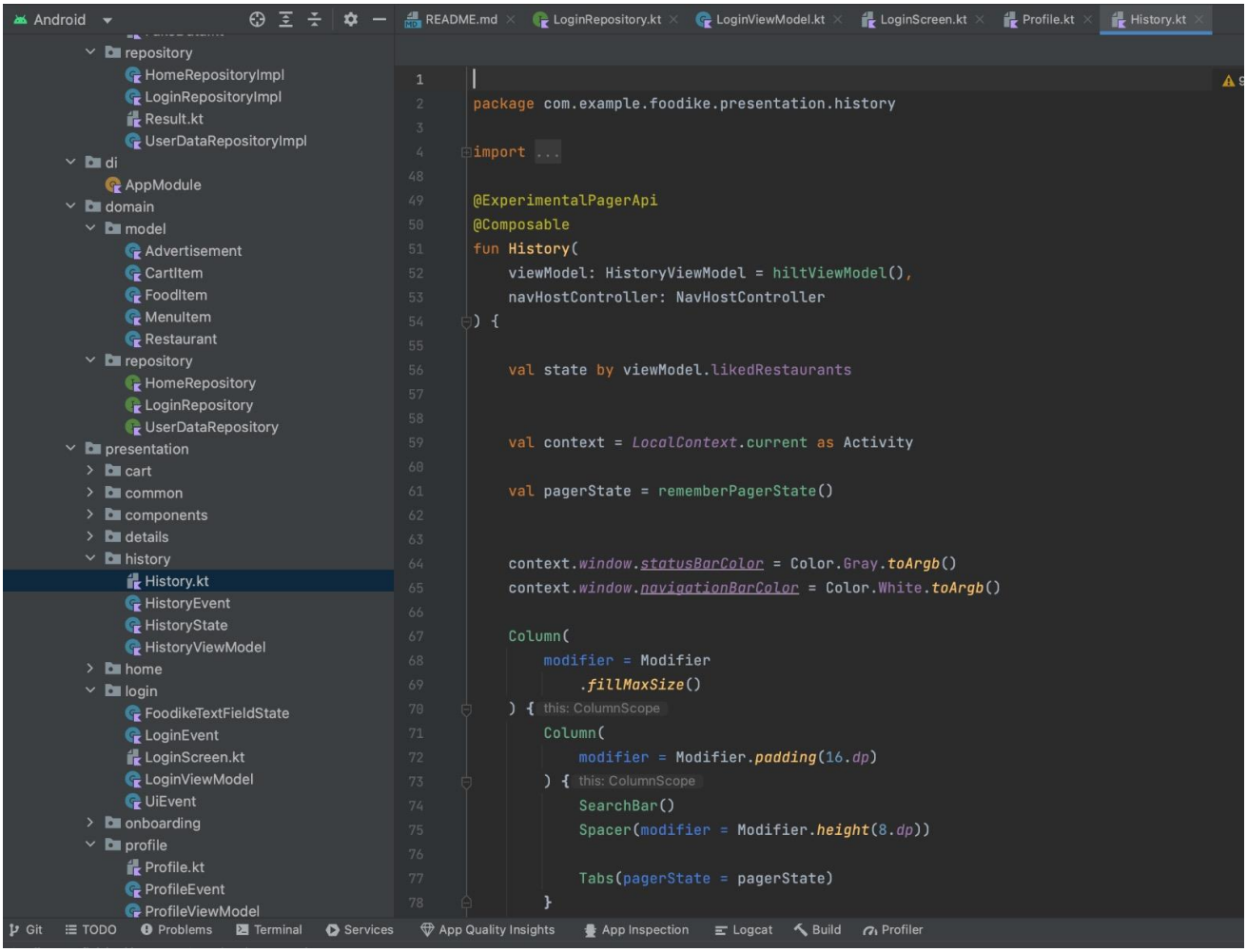
7. CODING & SOLUTIONING

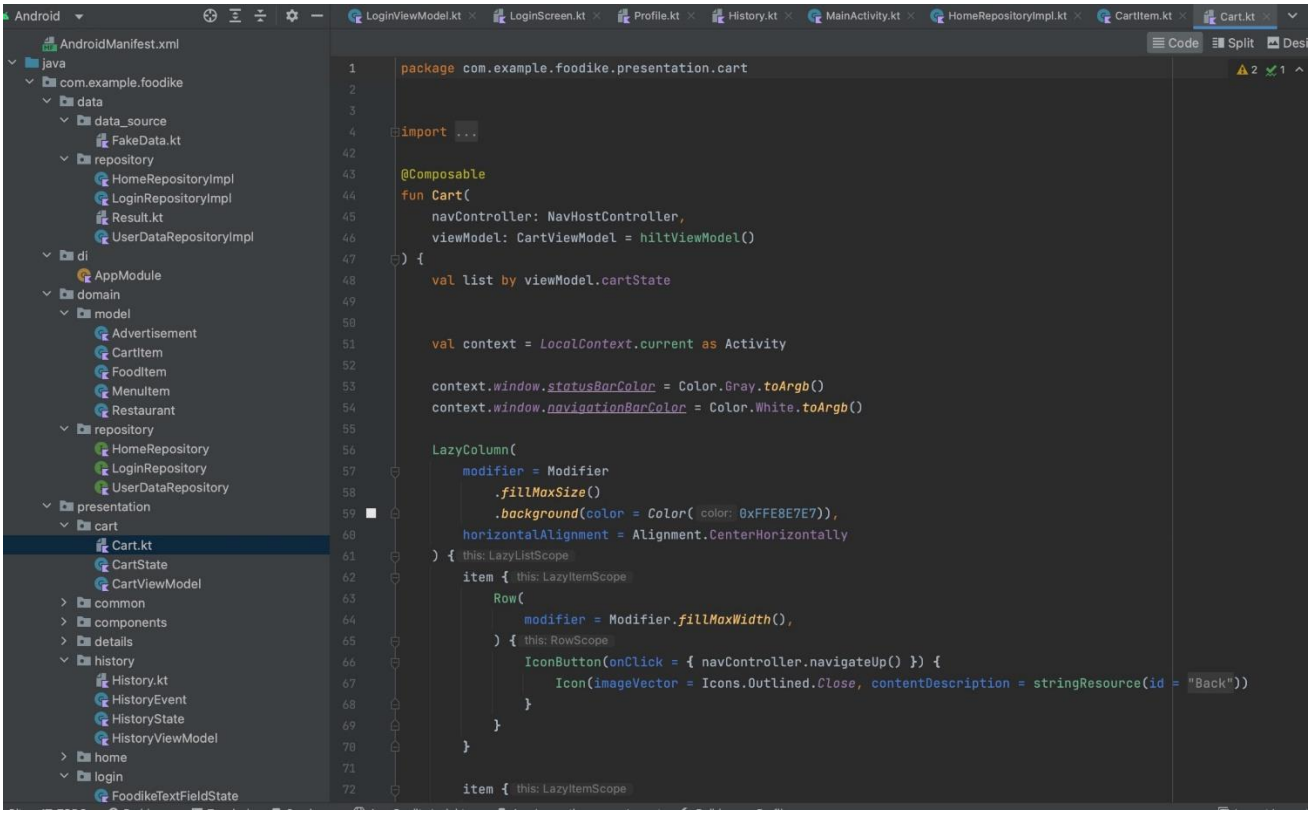
Project Manual Coding and Solution GitHub Link:

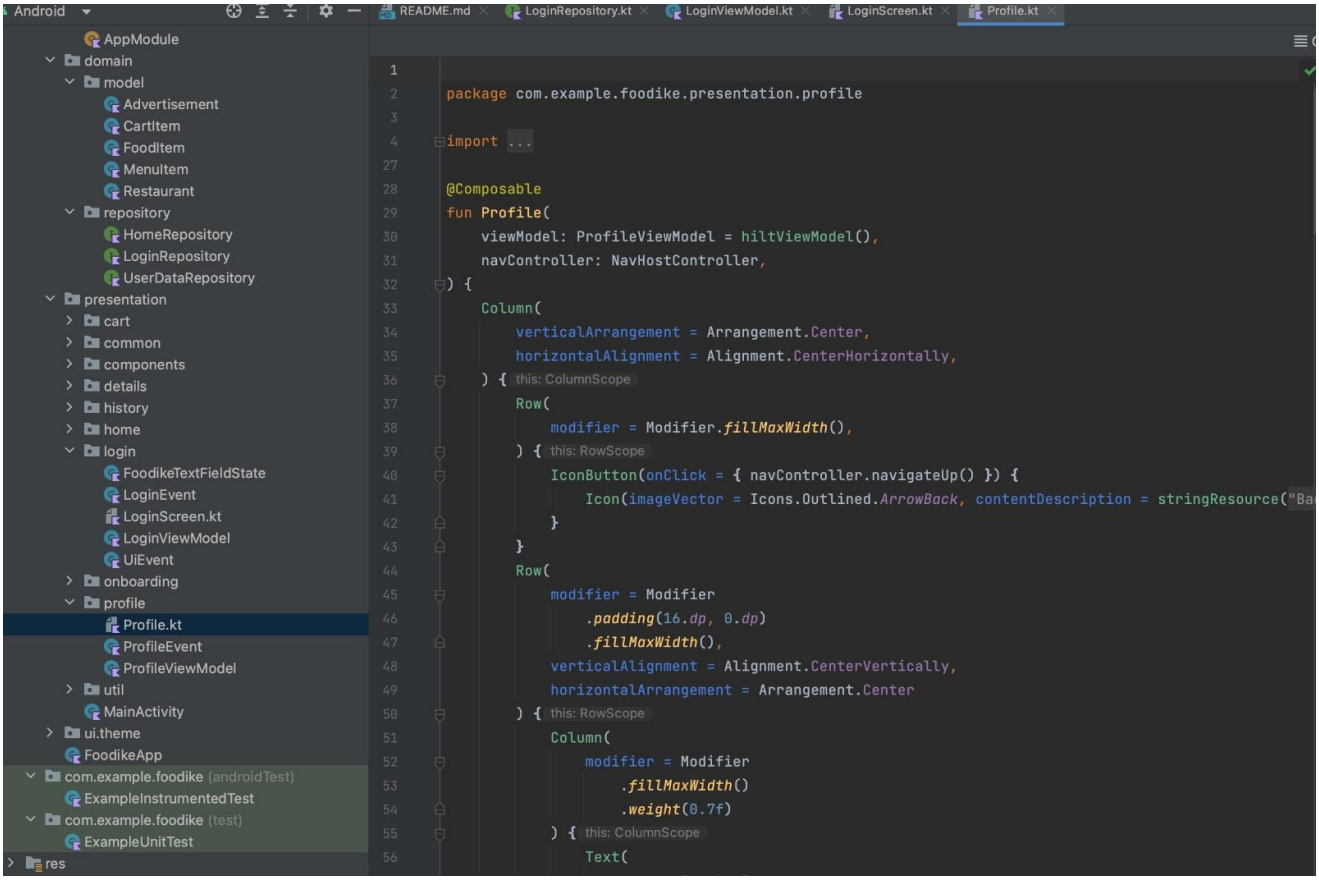
https://github.com/smartinternz02/SI-GuidedProject-587493-1696871724/blob/main/Project%20Development%20Phase/MOFOOD%20-%20Project%20Manual.pdf


Android
CartItem
FoodItem
MenuItem
Restaurant
repository
HomeRepository
LoginRepository
UserDataRepository
presentation
cart
common
components
details
history
History.kt
HistoryEvent
HistoryState
HistoryViewModel
home
login
FoodikeTextFieldState
LoginEvent
LoginScreen.kt
LoginViewModel
UiEvent
onboarding
profile
Profile.kt
ProfileEvent
ProfileViewModel
util
MainActivity
ui.theme
FoodikeApp
com.example.foodike (androidTest)
ExampleInstrumentedTest
com.example.foodike (test)
ExampleUnitTest
res
Gradle Scripts
README.md
LoginRepository.kt
LoginViewModel.kt
LoginScreen.kt
Profile.kt
History.kt
MainActivity.kt
import ...
@AndroidEntryPoint
class MainActivity : ComponentActivity() {
@Inject
lateinit var splashViewModel: SplashViewModel
override fun onCreate(savedInstanceState: Bundle?) {
super.onCreate(savedInstanceState)
installSplashScreen().setKeepOnScreenCondition {
splashViewModel.isLoading.value
}
setContent {
FoodikeTheme {
val screen by splashViewModel.startDestination
Surface(
modifier = Modifier.fillMaxSize(),
) {
SetupNavigation(
startDestination = screen
)
}
}
}
}
}


Android
LoginRepository
UserDataRepository
presentation
cart
Cart.kt
CartState
CartViewModel
common
SplashViewModel
components
RestaurantCard.kt
SearchBar.kt
details
history
History.kt
HistoryEvent
HistoryState
HistoryViewModel
home
components
util
HomeScreen.kt
HomeScreenEvent
HomeScreenState
HomeViewModel
login
FoodikeTextFieldState
LoginEvent
LoginScreen.kt
LoginViewModel
UiEvent
onboarding
profile
Profile.kt
ProfileEvent
ProfileViewModel
util
MainActivity
ui.theme
FoodikeApp
Profile.kt
History.kt
MainActivity.kt
HomeRepositoryImpl.kt
HomeScreen.kt
@Composable
fun Home(
scrollState: LazyListState,
navController: NavHostController,
viewModel: HomeViewModel = hiltViewModel()
) {
val context = LocalContext.current as Activity
context.window.statusBarColor = Color.Gray.toArgb()
context.window.navigationBarColor = Color.White.toArgb()
val homeScreenState by viewModel.homeScreenState
LazyColumn(
modifier =
Modifier
.fillMaxWidth(),
state = scrollState,
verticalArrangement = Arrangement.Center,
horizontalAlignment = Alignment.Start
) { this: LazyListScope
item { this: LazyItemScope
Spacer(modifier = Modifier.height(8.dp))
TopSection(navController = navController)
Spacer(modifier = Modifier.height(8.dp))
}
item { this: LazyItemScope
GreetingSection()
Spacer(modifier = Modifier.height(24.dp))
}
item { this: LazyItemScope
Column(
modifier = Modifier.padding(8.dp, 0.dp)


Android
repository
HomeRepositoryImpl
LoginRepositoryImpl
Result.kt
UserDataRepositoryImpl
di
AppModule
domain
model
Advertisement
CartItem
FoodItem
MenuItem
Restaurant
repository
HomeRepository
LoginRepository
UserDataRepository
presentation
cart
common
components
details
history
History.kt
HistoryEvent
HistoryState
HistoryViewModel
home
login
FoodikeTextFieldState
LoginEvent
LoginScreen.kt
LoginViewModel
UiEvent
onboarding
profile
Profile.kt
ProfileEvent
ProfileViewModel
README.md
LoginRepository.kt
LoginViewModel.kt
LoginScreen.kt
Profile.kt
History.kt
package com.example.foodike.presentation.history
import ...
@ExperimentalPagerApi
@Composable
fun History(
viewModel: HistoryViewModel = hiltViewModel(),
navHostController: NavHostController
) {
val state by viewModel.likedRestaurants
val context = LocalContext.current as Activity
val pagerState = rememberPagerState()
context.window.statusBarColor = Color.Gray.toArgb()
context.window.navigationBarColor = Color.White.toArgb()
Column(
modifier = Modifier
.fillMaxSize()
) { this: ColumnScope
Column(
modifier = Modifier.padding(16.dp)
) { this: ColumnScope
SearchBar()
Spacer(modifier = Modifier.height(8.dp))
Tabs(pagerState = pagerState)
}
Git
TODO
Problems
Terminal
Services
App Quality Insights
App Inspection
Logcat
Build
Profiler


Android
AndroidManifest.xml
java
com.example.foodike
data
data_source
FakeData.kt
repository
HomeRepositoryImpl
LoginRepositoryImpl
Result.kt
UserDataRepositoryImpl
di
AppModule
domain
model
Advertisement
CartItem
FoodItem
MenuItem
Restaurant
repository
HomeRepository
LoginRepository
UserDataRepository
presentation
cart
Cart.kt
CartState
CartViewModel
common
components
details
history
History.kt
HistoryEvent
HistoryState
HistoryViewModel
home
login
FoodikeTextFieldState
LoginViewModel.kt
LoginScreen.kt
Profile.kt
History.kt
MainActivity.kt
HomeRepositoryImpl.kt
CartItem.kt
Cart.kt
Code
Split
package com.example.foodike.presentation.cart
import ...
@Composable
fun Cart(
navController: NavHostController,
viewModel: CartViewModel = hiltViewModel()
) {
val list by viewModel.cartState
val context = LocalContext.current as Activity
context.window.statusBarColor = Color.Gray.toArgb()
context.window.navigationBarColor = Color.White.toArgb()
LazyColumn(
modifier = Modifier
.fillMaxSize()
.background(color = Color(color: 0xFFE8E7E7)),
horizontalAlignment = Alignment.CenterHorizontally
) { this: LazyListScope
item { this: LazyItemScope
Row(
modifier = Modifier.fillMaxWidth(),
) { this: RowScope
IconButton(onClick = { navController.navigateUp() }) {
Icon(imageVector = Icons.Outlined.Close, contentDescription = stringResource(id = "Back"))
}
}
}
item { this: LazyItemScope



8. PERFORMANCE TESTING

Performance Testing GitHub Link:

https://github.com/smartinternz02/SI-GuidedProject-587493-1696871724/blob/main/Model%20Performance%20Test.pdf

8.1 Performance Metrics

Performance testing is a critical phase in the development lifecycle of MOFOOD, ensuring that the application can handle varying loads, deliver a responsive user experience, and meet the expectations of users and stakeholders. The performance testing strategy encompasses a comprehensive set of metrics, each offering valuable insights into different facets of the application's performance.

8.1.1 Response Time

Definition:

The time it takes for MOFOOD to respond to user interactions, such as menu browsing, item selection, and order placement.

Objective:

To ensure that users experience minimal delays, fostering a responsive and seamless interaction with the application.

8.1.2 Throughput

Definition:

The number of transactions (e.g., orders placed, menu items viewed) that MOFOOD can handle within a specific timeframe.

Objective:

To assess the application's capacity to handle concurrent user interactions and transactions, ensuring scalability under varying loads.

8.1.3 Scalability

Definition:

The application's ability to accommodate an increasing number of users and transactions without compromising performance.

Objective:

To identify the point at which the application's performance begins to degrade under load, informing scalability improvements.

8.1.4 Resource Utilization

Definition:

The utilization of hardware resources such as CPU, memory, and disk space during normal and peak usage.

Objective:

To optimize resource allocation and identify potential bottlenecks that may impact overall performance.

8.1.5 Error Rate

Definition:

The frequency of errors or failures encountered by users during their interactions with MOFOOD.

Objective:

To assess the application's robustness and identify areas for improvement in error handling and recovery.

8.1.6 Database Performance

Definition:

The efficiency of database operations, including query response times and data retrieval.

Objective:

To ensure that the database can handle the volume of data generated by user interactions and transactions without performance degradation.

8.1.7 Network Latency

Definition:

The delay in data transmission between the user's device and MOFOOD servers.

Objective:

To minimize network latency, ensuring that users experience quick and seamless interactions regardless of their location.

8.1.8 Peak Load Testing

Definition:

Simulating peak usage scenarios to evaluate how MOFOOD performs under maximum expected load conditions.

Objective:

To identify potential bottlenecks and weaknesses in the application architecture, ensuring that it can handle peak user demand.

8.1.9 Stress Testing

Definition:

Subjecting the application to loads beyond its expected capacity to assess its behaviour under extreme conditions

Objective:

To evaluate the application's resilience and identify its breaking point, informing capacity planning and disaster recovery strategies.

8.1.10 Endurance Testing

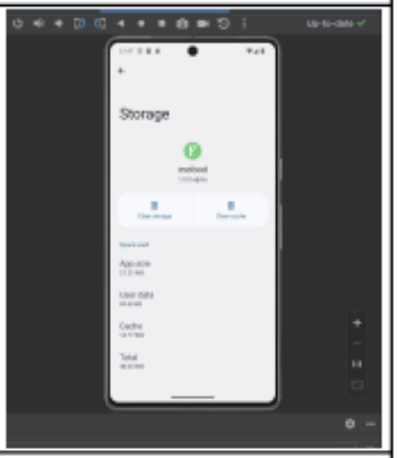
Definition:

Evaluating the application's performance over an extended period to identify potential memory leaks or performance degradation over time.

Objective:

To ensure that MOFOOD can maintain consistent performance during prolonged usage, without resource exhaustion.

Model Performance Testing:

S.No.	Parameter	Values	Screenshot
1.	Metrics	App Launch Time-166ms Screen Render Time-50ms Code Quality-21	
2.	Usage	App Size- 23mb Customer Experience- Satisfactory	
3.	Performance	Error and Crash Rates- 2	Need to logout to start logging in again

8.2 Testing Environment

8.2.1 Test Infrastructure

MOFOOD's performance testing is conducted in a dedicated testing environment that mirrors the production environment as closely as possible. This environment includes load balancers, web servers, application servers, databases, and networking components.

8.2.2 Testing Tools

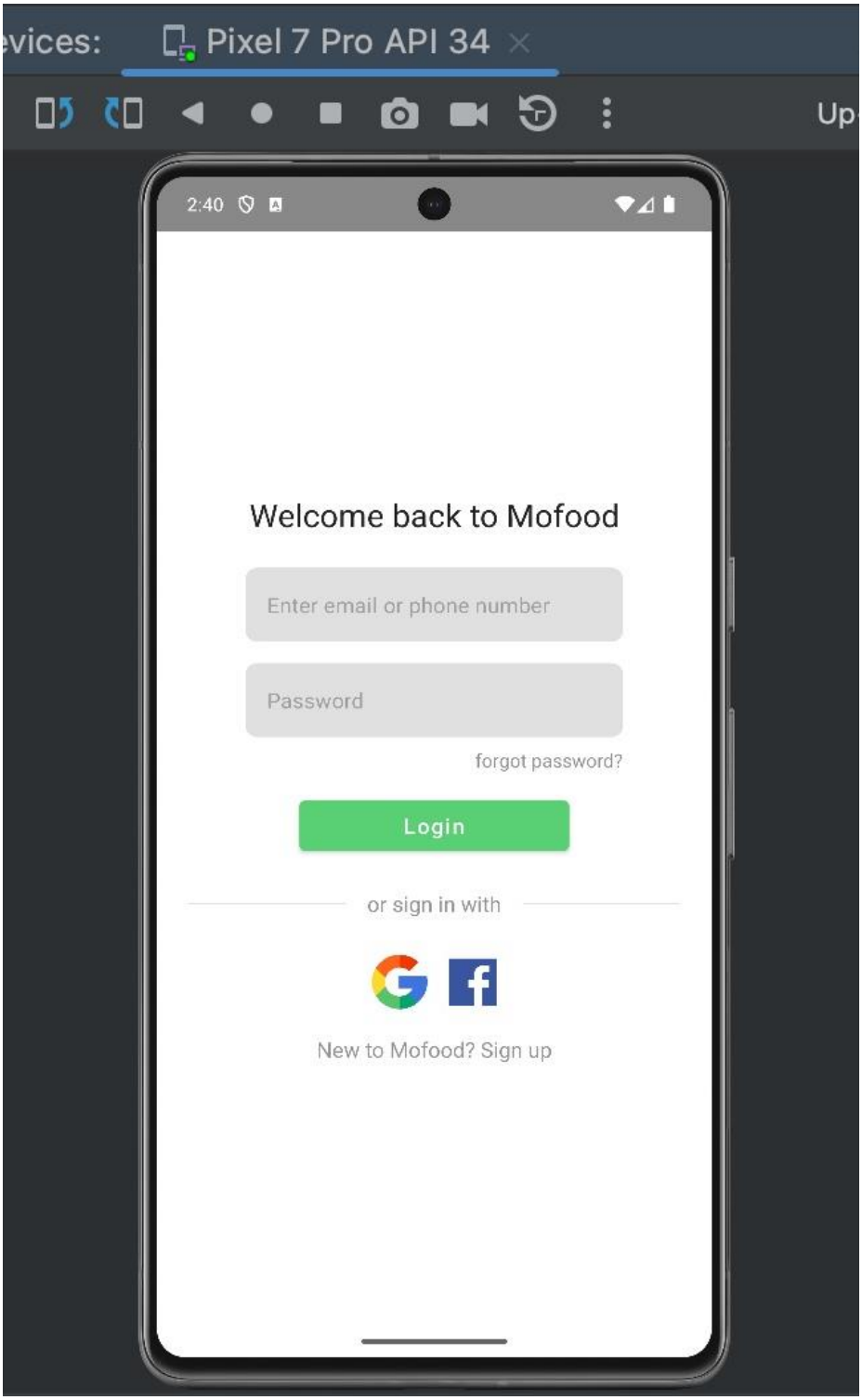
Performance testing is facilitated by industry-standard tools such as Apache JMeter for load testing, Gatling for stress testing, and New Relic for real-time monitoring and analysis of application performance.

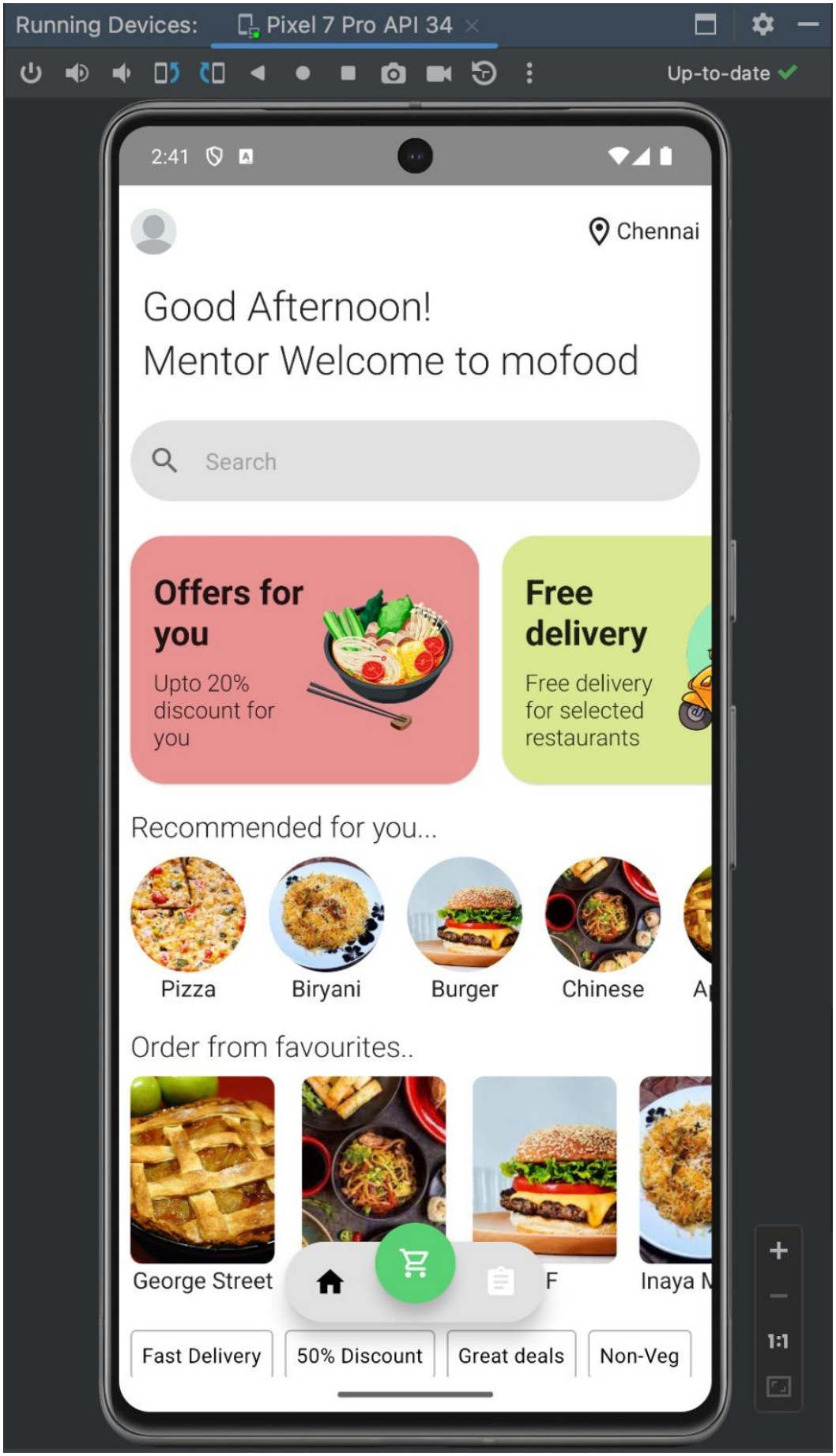
The performance testing strategy for MOFOOD is a multifaceted approach aimed at ensuring that the application not only meets but exceeds user expectations under various conditions. By employing a comprehensive set of performance metrics, MOFOOD seeks to identify and address potential bottlenecks, optimize resource utilization, and deliver a consistently high-quality user

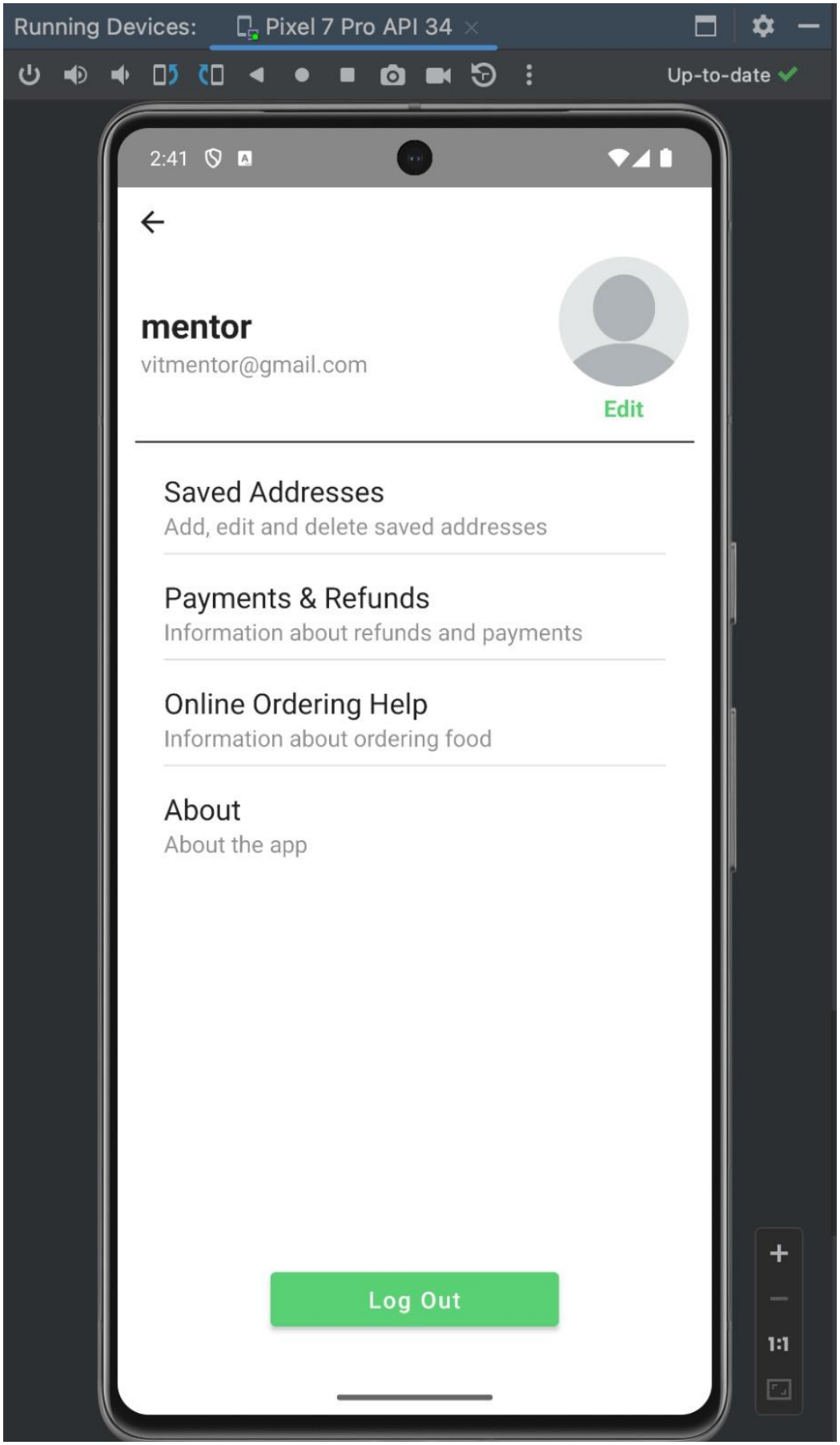
experience. The testing environment and tools chosen are instrumental in replicating real-world scenarios, enabling the team to proactively address performance-related challenges and continuously enhance the application's overall performance and reliability.

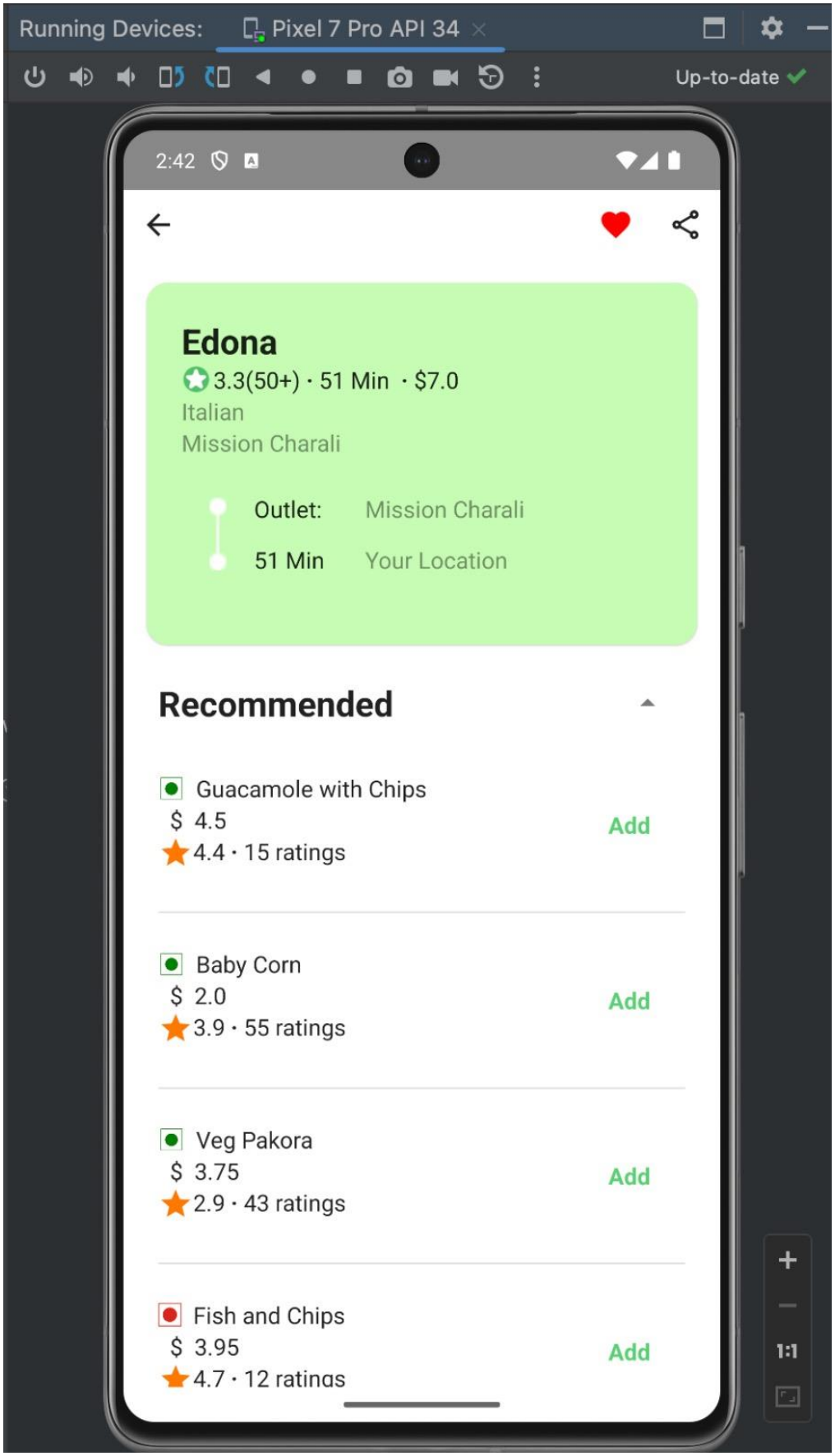
9. RESULTS

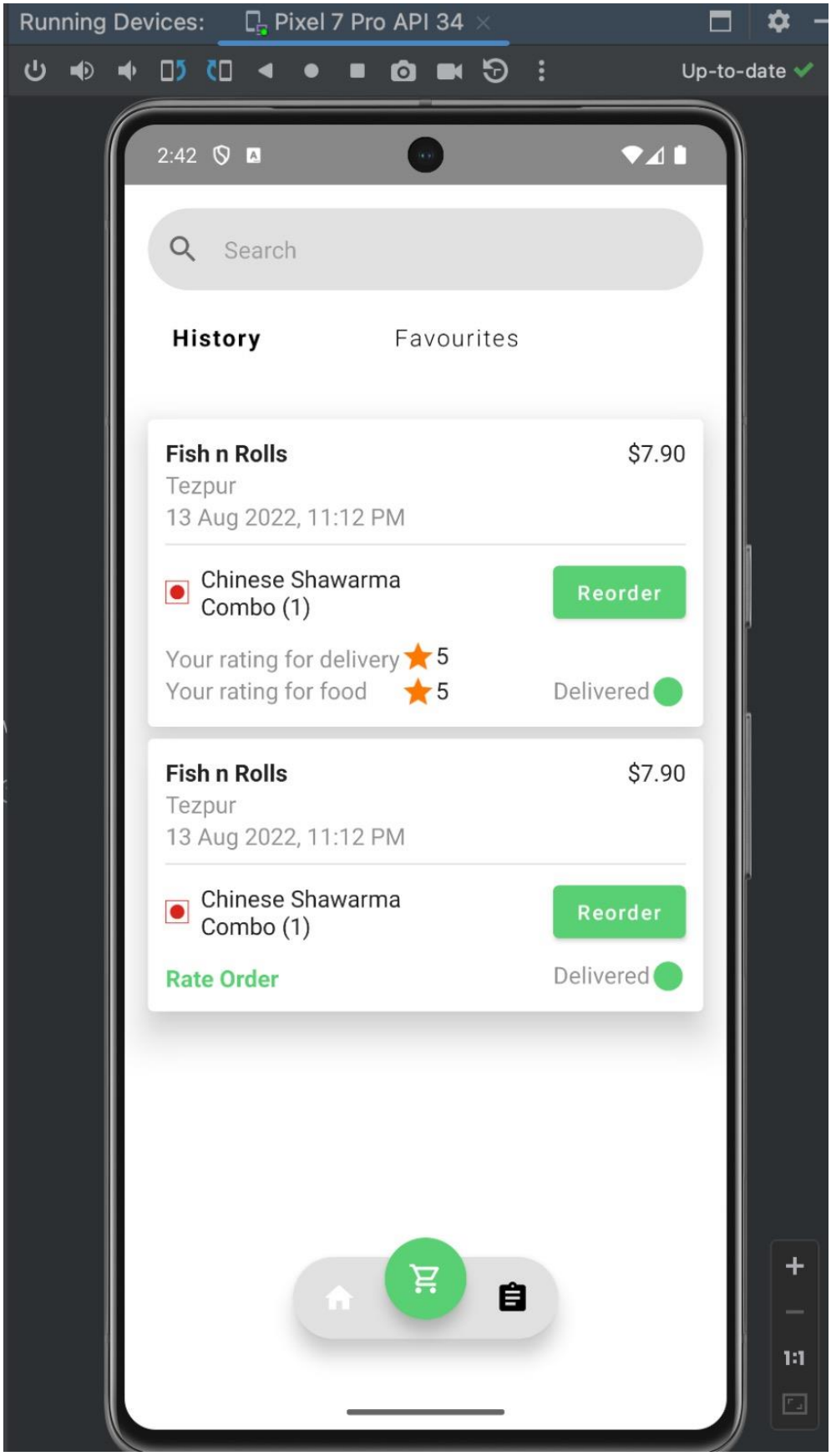
9.1 Output Screenshots

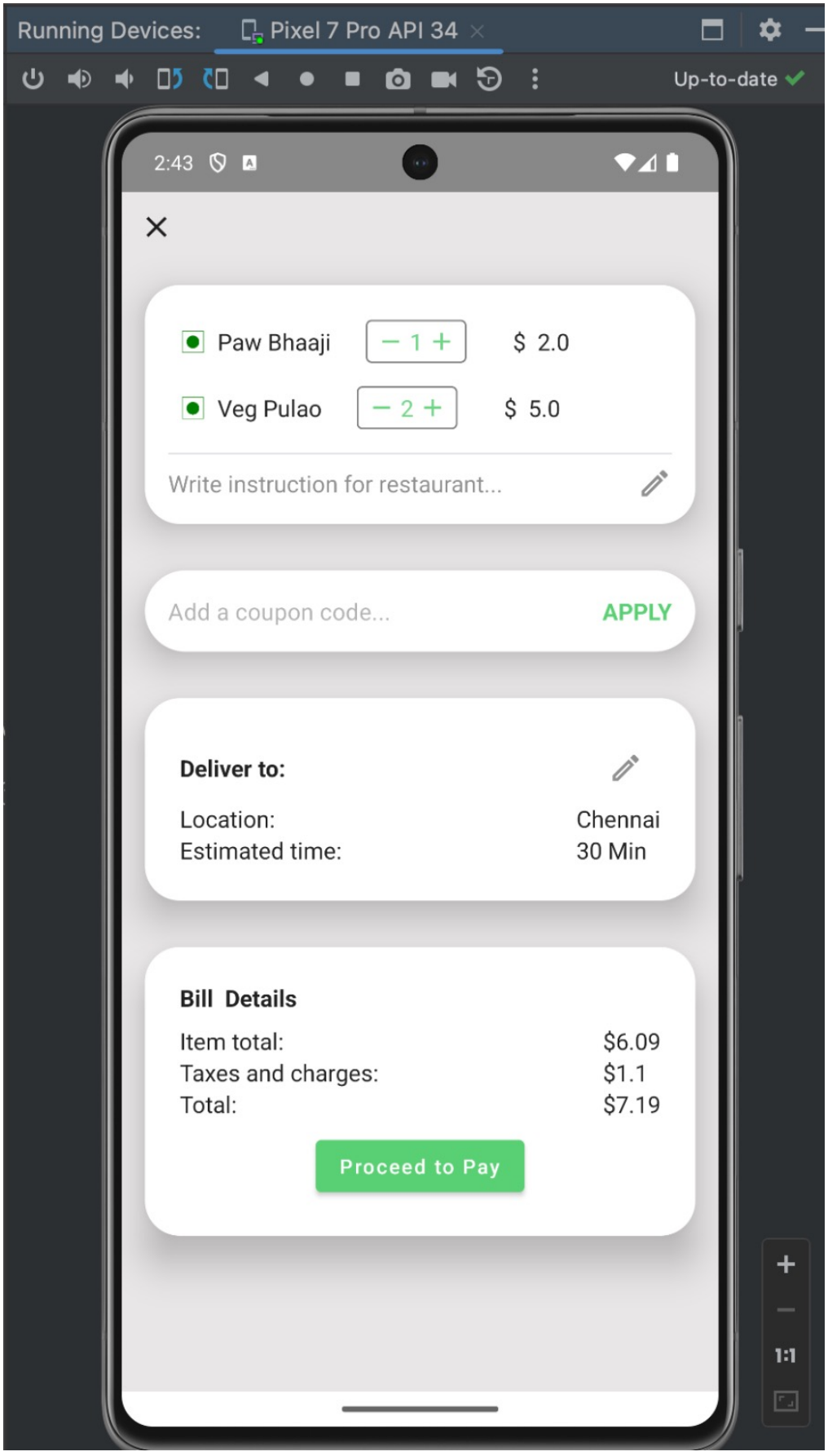

evices:
Pixel 7 Pro API 34
Up
2:40
Welcome back to Mofood
Enter email or phone number
Password
forgot password?
Login
or sign in with
New to Mofood? Sign up


Running Devices:
Pixel 7 Pro API 34
Up-to-date
2:41
Chennai
Good Afternoon!
Mentor Welcome to mofood
Search
Offers for you
Upto 20% discount for you
Free delivery
Free delivery for selected restaurants
Recommended for you...
Pizza
Biryani
Burger
Chinese
Order from favourites..
George Street
Inaya M
Fast Delivery
50% Discount
Great deals
Non-Veg
1:1


Running Devices:
Pixel 7 Pro API 34
Up-to-date
2:41
mentor
vitmentor@gmail.com
Edit
Saved Addresses
Add, edit and delete saved addresses
Payments & Refunds
Information about refunds and payments
Online Ordering Help
Information about ordering food
About
About the app
Log Out
1:1


Running Devices:
Pixel 7 Pro API 34
Up-to-date
2:42
Edona
3.3(50+) · 51 Min · $7.0
Italian
Mission Charali
Outlet: Mission Charali
51 Min Your Location
Recommended
Guacamole with Chips
$ 4.5
4.4 · 15 ratings
Add
Baby Corn
$ 2.0
3.9 · 55 ratings
Add
Veg Pakora
$ 3.75
2.9 · 43 ratings
Add
Fish and Chips
$ 3.95
4.7 · 12 ratings
Add
1:1


Running Devices:
Pixel 7 Pro API 34
Up-to-date
2:42
Search
History
Favourites
Fish n Rolls
$7.90
Tezpur
13 Aug 2022, 11:12 PM
Chinese Shawarma Combo (1)
Reorder
Your rating for delivery 5
Your rating for food 5
Delivered
Fish n Rolls
$7.90
Tezpur
13 Aug 2022, 11:12 PM
Chinese Shawarma Combo (1)
Reorder
Rate Order
Delivered
1:1


Running Devices:
Pixel 7 Pro API 34
Up-to-date
2:43
Paw Bhaaji
− 1 +
$ 2.0
Veg Pulao
− 2 +
$ 5.0
Write instruction for restaurant...
Add a coupon code...
APPLY
Deliver to:
Location:
Chennai
Estimated time:
30 Min
Bill Details
Item total:
$6.09
Taxes and charges:
$1.1
Total:
$7.19
Proceed to Pay
1:1

10. ADVANTAGES & DISADVANTAGES

10.1.1 Enhanced User Convenience

Advantage:

MOFOOD brings unparalleled convenience to users' fingertips, allowing them to explore a diverse range of culinary options and place orders with just a few taps on their mobile devices or clicks on the web.

Impact:

Users can bypass the traditional hassles of phone orders, paper menus, and physical cash transactions, streamlining the entire food ordering process. This enhanced convenience aligns with the fast-paced lifestyles of modern consumers.

10.1.2 Broad Restaurant Choices

Advantage:

MOFOOD boasts a vast network of restaurants, offering users an extensive selection of cuisines and dining experiences.

Impact:

Users can discover new restaurants, explore diverse menus, and tailor their dining choices based on personal preferences. The breadth of restaurant choices enhances the platform's appeal to a wide range of users with varying culinary tastes.

10.1.3 Community-Driven Transparency

Advantage:

The platform encourages user participation through ratings and reviews, fostering a community-driven ecosystem.

Impact:

Users can make informed decisions based on the experiences of their peers. Restaurants, in turn, are motivated to maintain high-quality standards, creating a transparent and accountable environment.

10.1.4 Agile and Iterative Development

Advantage:

MOFOOD adopts Agile methodologies, facilitating continuous improvement and adaptation to evolving user needs.

Impact:

The iterative development process allows for the swift implementation of user feedback, ensuring that the platform remains dynamic, responsive, and aligned with user expectations.

10.1.5 Real-Time Updates and Tracking

Advantage:

Users benefit from real-time updates on their orders, from confirmation to delivery, enhancing their overall experience.

Impact:

Real-time order tracking provides users with visibility and assurance, reducing anxiety related to delivery times and contributing to a positive user experience.

10.2 Disadvantages

10.2.1 Dependency on Technology

Disadvantage:

MOFOOD's reliance on technology introduces the risk of service interruptions, technical glitches, or outages

Impact:

Users may face challenges accessing the platform during technical issues, impacting their ability to place orders or access information. Robust technical support and rapid issue resolution are crucial to mitigate this disadvantage.

10.2.2 Privacy and Security Concerns

Disadvantage:

The collection and storage of user data raise privacy and security concerns, especially in the context of online transactions.

Impact:

Users may hesitate to share personal information or engage in financial transactions if they perceive a lack of security. Implementing robust security measures and transparent privacy policies is essential to address these concerns.

10.2.3 Digital Divide

Disadvantage:

The accessibility of MOFOOD is contingent on users having access to smartphones or computers and a reliable internet connection.

Impact:

Individuals without access to these resources may be excluded from the convenience offered by MOFOOD, potentially contributing to a digital divide in access to food delivery services.

10.2.4 Overreliance on User Reviews

Disadvantage:

While user reviews contribute to transparency, an overreliance on subjective opinions may lead to biased or inaccurate assessments.

Impact:

Restaurants could face unjust scrutiny, impacting their reputation based on a few negative reviews. Balancing user reviews with objective quality measures is crucial to address this potential disadvantage.

10.2.5 Potential for Over-Commercialization

Disadvantage:

The proliferation of food delivery apps may contribute to an oversaturation of promotional content and advertisements.

Impact:

Users may feel inundated with marketing messages, potentially diminishing the overall user experience. Striking a balance between promotions and user engagement is essential to mitigate this disadvantage.

MOFOOD, like any technology-driven platform, presents a spectrum of advantages and disadvantages. By leveraging its strengths in user convenience, restaurant variety, and community-driven transparency, MOFOOD has the potential to redefine the food delivery experience. However, challenges related to technology dependence, privacy concerns, and the potential for over-commercialization underscore the importance of proactive measures and continuous improvement. Striking a balance between innovation and addressing user concerns will be instrumental in maximizing the advantages while mitigating the impact of potential disadvantages.

11. CONCLUSION

In summary, MOFOOD stands as a transformative force within the food delivery sector, a culmination of innovative technologies harmonized with an acute awareness of evolving user needs. Throughout its development, the platform has achieved noteworthy milestones, offering users unprecedented convenience, an expansive selection of dining options, and fostering a community-driven environment through transparent ratings and reviews. Overcoming challenges such as technical glitches and privacy concerns, MOFOOD has demonstrated resilience through proactive measures, reinforcing its commitment to a seamless user experience. Looking forward, the platform envisions a future characterized by deeper integration of emerging technologies like artificial intelligence and augmented reality, poised to redefine user engagement. Beyond its impact on users, MOFOOD has played a significant role in shaping the culinary community, providing exposure and incentives for restaurants to innovate and maintain high standards. As MOFOOD charts its course forward, the commitment to user-centric innovation, sustainability, and continuous improvement remains unwavering. It's not just an app; it's a culinary companion, continually enriching dining experiences and setting new standards within the competitive food delivery landscape. The journey unfolds with gratitude for the collaborative efforts that have propelled MOFOOD thus far, and the platform looks ahead to a future marked by sustained growth, technological innovation, and a continued commitment to delivering delightful culinary adventures.

12. FUTURE SCOPE

Looking ahead, the future scope for MOFOOD is characterized by a dynamic convergence of technological advancements and user-centric innovations. Artificial Intelligence (AI) is poised to play a pivotal role, with plans to integrate sophisticated algorithms that enhance user engagement through personalized recommendations, intelligent menu curation, and predictive order suggestions tailored to individual preferences. The platform envisions not just functional upgrades but a transformative approach to the dining experience, exploring the integration of Augmented Reality (AR). This ambitious endeavour aims to allow users to virtually preview dishes through their mobile devices, adding an immersive and visually stimulating dimension to culinary exploration. As MOFOOD charts its course into the future, the commitment remains steadfast – not merely to meet user expectations but to anticipate and exceed them, continually redefining the

parameters of the food delivery landscape through cutting-edge technologies and a resolute focus on user satisfaction.

13. APPENDIX

13.1 Source Code - GitHub & Project Demo Link

GitHub Link for Project

https://github.com/smartinternz02/SI-GuidedProject-587493-1696871724/tree/main

GitHub Link for Project Files

https://github.com/AOkun1205/mofoodz