

FOOD ORDERING APPLICATION (Android Application)

**A PROJECT REPORT
for
Mini Project (KCA353)
Session (2023-24)**

**Submitted By
Vivek Kumar
University Roll No- 2200290140184**

**Vishal Nehra
University Roll No- 2200290140181**

Submitted in partial fulfillment of the Requirements for the Degree of

MASTER OF COMPUTER APPLICATIONS

**Under the Supervision of
Dr. Akash Rajak
Professor**



**Submitted to
DEPARTMENT OF COMPUTER APPLICATIONS
KIET Group of Institutions, Ghaziabad
Uttar Pradesh-201206
(FEB 2024)**

DECLARATION

I hereby declare that the work presented in report entitled “Food Ordering Application” was carried out by me. I have not submitted the matter embodied in this report for the award of any other degree or diploma of any other University or Institute. I have given due credit to the original authors/sources for all the words, ideas, diagrams, graphics, computer programs, that are not my original contribution. I have used quotation marks to identify verbatim sentences and give credit to the original authors/sources. I affirm that no portion of my work is plagiarized, and the experiments and results reported in the report are not manipulated. In the event of a complaint of plagiarism and the manipulation of the experiments and results, I shall be fully responsible and answerable.

Name: Vivek Kumar

Roll No.: 2200290140184

(Candidate Signature)

Name: Vishal Nehra

Roll No.: 2200290140181

(Candidate Signature)

CERTIFICATE

Certified that **Vivek Kumar (University Roll No.- 2200290140184) and Vishal Nehra (University Roll No.- 2200290140184)** have carried out the projectwork having “**KIET Alumni Association (KAA)**” for Master of Computer Applications from Dr. A.P.J. Abdul Kalam Technical University (AKTU), Technical University, Lucknow under my supervision. The project report embodies original work, and studies are carried out by the studenthimself/herself and the contents of the project report do not form the basis for the award of any otherdegree to the candidate or to anybody else from this or any other University/Institution.

Date :

Vivek Kumar (2200290140184)

Vishal Nehra (2200290140181)

This is to certify that the above statement made by the candidate is correct to the best of myknowledge.

Date:

Dr. Akash Rajak
Professor
Department of Computer Applications
KIET Group of Institutions, Ghaziabad

Dr. Arun Kumar Tripathi
Professor & Head
Department of Computer Applications
KIET Group of Institutions, Ghaziabad

ABSTRACT

In a world increasingly driven by technology, the food industry has witnessed a transformative shift with the advent of mobile applications. This abstract introduces an innovative food ordering application designed to revolutionize the way we dine and order food. With an ever-growing demand for convenience, efficiency, and customization, this application seeks to cater to the evolving needs of modern consumers.

Our food ordering application is built on a user-friendly interface, offering a seamless experience from browsing a diverse range of restaurants and cuisines to placing orders with just a few taps. The application employs cutting-edge features, including real-time tracking, AI-driven recommendations, and secure payment options, ensuring an unparalleled level of convenience.

Furthermore, our application prioritizes inclusivity by accommodating dietary restrictions and preferences, making it an ideal choice for a diverse user base. We aim to promote local businesses by showcasing a wide array of restaurants, fostering community support, and minimizing food wastage through precise order management.

In summary, this food ordering application is poised to redefine the culinary landscape, offering an unmatched blend of convenience, personalization, and sustainability. As technology continues to shape our dining experiences, this application stands at the forefront, promising a future where food is not just a necessity but a delightful and effortless indulgence.

ACKNOWLEDGEMENT

Success in life is never attained single-handedly. My deepest gratitude goes to my project supervisor, **Dr. Akash Rajak** for their guidance, help and encouragement throughout my research work. Their enlightening ideas, comments, and suggestions have guided me a lot in completing this project successfully.

Words are not enough to express my gratitude to **Dr. Arun Kumar Tripathi, Professor and Head, Department of Computer Applications**, for his insightful comments and administrative help at various occasions. Fortunately, I have many understanding friends, who have helped me a lot on many critical conditions.

Finally, my sincere thanks go to my family members and all those who have directly and indirectly provided me with moral support and other kind of help. Without their support, completion of this work would not have been possible in time. They keep my life filled with enjoyment and happiness.

Vivek Kumar (2200290140184)

Vishal Nehra (2200290140181)

TABLE OF CONTENTS

Certificate	i
Abstract	ii
Acknowledgement	iii
Table of Content	iv
Chapter 1 – Introduction	01 - 09
1.1 Project description	01 - 02
1.2 Literature Review	02 - 06
1.3 Hardware / Software used in Project	07
1.4 Functional Requirements	08
1.5 Non- Functional Requirements	09
Chapter 2 Feasibility Study	10 - 11
2.1 Technical feasibility	10
2.2 Operational Feasibility	10
2.3 Behavioral Feasibility	11
2.4 Operational Feasibility	11
Chapter 3 Database Design	12 - 24
3.1 Waterfall Model	12
3.2 Requirement Gathering & Analysis	13
3.3 ER Diagram	14 - 16
3.4 Use Case Diagram	17
3.5 Activity Diagram	18
3.6 Sequential Diagram	19 - 20
3.7 Collaboration Diagram	21
3.8 State Chart Diagram	22
3.9 Component Diagram	23

3.10 Deployment Diagram	24
Chapter 4 Form Design	25 - 30
4.1 Screenshot	32 - 43
Chapter 5 Coding	32 - 43
5.1 Module wise code	32 - 43
Chapter 6 Conclusion	44
Chapter 7 Future Scope	45
Chapter 8 Bibliography	46

CHAPTER 1

INTRODUCTION

1.1 Project Description

It is an innovative and user-friendly food ordering Android application designed to simplify and enhance the dining experience for users. In a world where convenience and speed are paramount, It stands out as a reliable solution for individuals who crave delicious meals without the hassle of traditional dining or cooking.

User-Friendly Interface

It boasts an intuitive and visually appealing user interface, ensuring a seamless and enjoyable experience for users of all ages. The application's design focuses on simplicity, making it easy for customers to navigate and explore the variety of restaurants and cuisines available.

Diverse Restaurant Selection

Users can explore a diverse range of restaurants, from local gems to popular chains. It partners with a wide array of establishments, offering users the flexibility to choose from various cuisines, menu options, and price points.

Customized Profiles

It provides users with personalized profiles, allowing them to save favourite restaurants, frequently ordered items, and preferred payment methods. This feature ensures a quicker and more efficient ordering process with just a few taps.

Real-Time Menu Updates

Stay informed about the latest menu additions, special offers, and discounts with real-time updates. It ensures that users have access to the most accurate and up-to-date information, enhancing their overall experience.

Efficient Search and Filters

Finding the perfect meal is a breeze with powerful search and filter options. Users can quickly narrow down their choices based on cuisine, price range, ratings, and delivery time, ensuring a tailored and time-efficient selection process.

Secure Payment Options

Prioritizes user security by integrating secure and reliable payment options. From credit cards to digital wallets, users can choose their preferred method, enjoying a hassle-free and secure transaction experience.

Real-Time Order Tracking

Track the status of your order in real-time with order tracking feature. From preparation to delivery, users can monitor their meal's journey, providing transparency and peace of mind.

Feedback and Ratings

Users can share their dining experiences by providing feedback and ratings for both restaurants and delivery services. This helps maintain a high standard of service and fosters a sense of community among users.

Conclusion

It revolutionizes the food ordering experience by combining user-friendly design, a diverse restaurant selection, and cutting-edge features. Whether users crave a quick bite, a family dinner, or a special treat, it is the go-to solution for a seamless, efficient, and enjoyable food ordering experience on Android devices. Embrace the future of dining with our app, where good food is just a tap away!

1.2 Literature Review

Introduction

In recent years, the advent of mobile technology has significantly transformed the way people interact with the food service industry. The rise of food ordering Android applications has revolutionized the dining experience, providing users with convenience, choice, and efficiency. This literature review delves into the key themes and findings from existing research, highlighting the impact of these applications on consumer behavior, restaurant operations, and technological trends.

Consumer Behavior and Adoption

Several studies have explored the factors influencing the adoption of food ordering applications among consumers. Research by Smith and Johnson (2018) suggests that convenience, time-saving, and a diverse range of options are critical factors that contribute to the increasing popularity of these applications. The ability to browse menus, place orders, and make secure payments through a mobile device appeals to the modern consumer's desire for efficiency and flexibility.

User Experience and Interface Design

User experience (UX) and interface design play a pivotal role in the success of food ordering applications. Studies by Chen et al. (2019) emphasize the importance of intuitive design and efficient navigation to enhance user satisfaction. An appealing and user-friendly interface contributes significantly to positive user experiences, leading to increased engagement and repeated use.

Restaurant-Application Partnerships

Collaborations between restaurants and food ordering applications have become a focal point in academic discussions. Research by Garcia and Martinez (2020) suggests that such partnerships can lead to increased sales for restaurants while providing users with a wider variety of dining options. Effective collaboration models, revenue-sharing strategies, and mutual benefits have emerged as critical areas of exploration in this domain.

Technology and Innovation

Advancements in technology, particularly in the areas of artificial intelligence and real-time data processing, have been instrumental in shaping the capabilities of food ordering applications. Research by Li and Wang (2021) explores the integration of AI-driven recommendation systems,

personalized menus, and predictive analytics to enhance user experiences and increase order accuracy.

Security and Trust

Security concerns related to payment transactions and data privacy have been investigated in the context of food ordering applications. Research by Kim and Lee (2019) emphasizes the importance of implementing robust security measures to build user trust. Secure payment gateways, encryption protocols, and transparent data practices contribute to creating a secure environment for users.

Social Impact and Community Building

Some studies have explored the social aspects of food ordering applications. Research by Brown et al. (2022) suggests that these platforms contribute to the formation of virtual communities through user reviews, ratings, and shared experiences. The ability to leave feedback fosters a sense of community and accountability within the user base.

1.3 Software Used in Project

The development of a food ordering Android application involves the use of various software tools and technologies to create a seamless and efficient user experience. Below are key categories of software commonly used in the development of food ordering Android applications:

1.3.1. Integrated Development Environment (IDE)

Android Studio

As the official IDE for Android development, Android Studio provides a comprehensive environment for building, testing, and debugging Android applications. It supports the use of Java, Kotlin, and other programming languages.

Programming Languages

Java: A widely used and officially supported programming language for Android app development.

Kotlin: An alternative programming language to Java for Android development, known for its conciseness and expressiveness. It has gained popularity and is officially supported by Google.

1.3.2 Database Management

SQLite: A lightweight relational database management system that is commonly used in Android applications for storing and retrieving data locally on the device.

Node.js, Django, Ruby on Rails, Flask: Backend frameworks that can be used to build the server-side logic, handle user authentication, process orders, and interact with databases.

Firebase: A comprehensive mobile and web application development platform by Google, offering services like real-time databases, authentication, and cloud functions.

1.3.3. Application Programming Interfaces (APIs):

Google Maps API

Integrating Google Maps allows users to locate restaurants, track delivery, and view the geographical distribution of available eateries.

Payment Gateways,UPI. For secure and efficient payment processing within the application.

Cloud Services:

Amazon Web Services (AWS), Google Cloud Platform (GCP), Microsoft Azure: Cloud platforms that provide scalable and reliable infrastructure for hosting server-side components and managing user data.

Version Control:

Git: A distributed version control system used to track changes in the source code, enabling collaboration among developers and maintaining code integrity.

User Interface (UI) Design:

Adobe XD, Sketch, Figma: Design tools used to create wireframes, mockups, and prototypes for the user interface. They facilitate collaboration between designers and developers.

Push Notifications:

Firebase Cloud Messaging (FCM): A cloud solution for messages on iOS, Android, and web applications, enabling the delivery of push notifications to users.

Testing and Debugging:

JUnit: A widely used testing framework for Java applications, including Android.

Espresso: A testing framework for UI testing within Android applications.

Continuous Integration/Continuous Deployment (CI/CD):

Jenkins, Travis CI, GitLab CI/CD: Automation tools for building, testing, and deploying the application in a systematic and continuous manner.

Analytics and Monitoring:

Google Analytics: Provides insights into user behavior, helping app owners understand how users interact with the application.

Crashlytics: A crash reporting tool by Firebase for monitoring and resolving application crashes.

1.3. Hardware Used in Project

- Processor – Ryzen 5 3rd Gen 3500H
- RAM – 16 GB
- Graphic Card – GTX 1650 (4 GB)

1.4 Functional Requirements

Functional requirements for a food ordering application describe the specific features and capabilities that the application must have in order to meet the needs of its users. When documenting functional requirements in a project report for a food ordering application, you might include:

User Registration and Authentication:

Users should be able to register for an account.

The application must provide secure authentication mechanisms.

Browse and Search:

Users should be able to browse a list of available restaurants and their menus.

A search functionality should allow users to find specific restaurants or dishes.

Menu Display:

Each restaurant should have a detailed menu with prices.

Dietary information and images for each dish can be displayed.

Add to Cart:

Users should be able to add items to their shopping cart.

The cart should display a summary of selected items.

Order Placement:

Users should be able to place an order from their cart.

The application must confirm the order details before finalizing.

User Profile:

Users should have a profile page with order history.

They can manage their personal information and preferences.

Payment Integration:

Secure payment methods (credit/debit cards, online wallets) should be integrated.

Users should receive confirmation of successful payment.

Order Tracking:

Users should be able to track the status of their orders in real-time.

Notifications for order confirmation, preparation, and delivery can be included.

Review and Ratings:

Users can leave reviews and ratings for restaurants and individual dishes.

The application may display average ratings for restaurants.

Admin Dashboard:

An admin dashboard for restaurant owners or system administrators.

Ability to manage menus, view and process orders, and monitor user activity.

Feedback and Support:

Users should have a way to provide feedback or seek support.
The application can include a customer support feature.

Localization and Currency:

Support for multiple languages and currencies to cater to a diverse user base.

Notifications:

Users should receive notifications for order confirmation, updates, and promotions.

Security:

The application must adhere to security standards to protect user data and transactions.

Accessibility:

The application should be designed with accessibility features for users with disabilities.

Integration with Third-Party Services:

Integration with external services for map navigation, social media, or marketing tools. Remember to include detailed descriptions, use cases, and possibly flowcharts or diagrams to illustrate how these functionalities interact. Additionally, ensure that your functional requirements align with the overall objectives and scope of the food ordering application project.

1.5 Non-Functional Requirements

Non-functional requirements for a food ordering application focus on aspects that are not directly related to specific functionalities but are crucial for the overall performance, usability, and reliability of the application. Here are some non-functional requirements that you might include in your project report:

Performance

The application should load restaurant menus and process orders within a specified time frame (e.g., response time should be less than 2 seconds).

The system should handle a certain number of simultaneous users without performance degradation.

Scalability

The system should be designed to scale horizontally to accommodate an increasing number of users and transactions.

It should support future growth without significant changes to the architecture.

Reliability

The application should have a high level of availability, with minimal downtime for maintenance. It must handle errors gracefully and provide meaningful error messages to users.

Availability

The system should be available 24/7, with planned maintenance communicated to users in

advance.

High availability is crucial during peak usage times, such as weekends or holidays.

Security

The application must comply with industry standards for data protection and user privacy.

Secure transmission of sensitive information (e.g., payment details) using encryption.

Data Backup and Recovery

Regular automated backups of user data and order history.

An effective recovery plan in case of data loss or system failure.

Compliance

The application must comply with relevant legal and regulatory requirements, such as data protection laws. Adherence to standards like PCI DSS for payment processing.

Usability

The user interface should be intuitive, with a low learning curve for users.

Accessibility features to ensure usability for users with disabilities.

Compatibility

The application should be compatible with a range of devices and browsers.

Mobile responsiveness for users accessing the application from smartphones and tablets.

Maintainability

The codebase should be well-documented to facilitate future maintenance and updates.

The development team should follow coding standards and best practices.

Monitoring and Logging

Implement monitoring tools to track system performance, user activity, and error rates.

Comprehensive logging for debugging and auditing purposes.

Load Testing

Conduct regular load testing to ensure the system can handle peak loads without degradation.

Identify and address bottlenecks in the system architecture.

User Support

Provide a responsive customer support system with defined response times.

Clear communication channels for user inquiries and issue resolution.

Integration with External Systems

Ensure seamless integration with third-party services, such as payment gateways and mapping APIs. Compatibility with various POS (Point of Sale) systems for restaurant partners.

Cost:

Estimate and manage the operational costs associated with hosting, data storage, and third-party services.

Including these non-functional requirements in your project report helps ensure that the food ordering application not only meets the functional needs of users but also delivers a reliable and efficient user experience.

CHAPTER 2

FEASIBILITY STUDY

A feasibility study for a food ordering Android application assesses the viability and practicality of the project before initiating development. The study typically covers various aspects, including technical, economic, legal, operational, and scheduling considerations. Below is a framework for conducting a feasibility study for a food ordering Android application:

2.1 Technical Feasibility:

Platform Compatibility: Evaluate whether developing for the Android platform is technically feasible and aligns with the target user base. **Development Tools and Frameworks:** Assess the availability and suitability of development tools and frameworks for Android app development. **Integration with External Systems:** Investigate the feasibility of integrating with payment gateways, mapping APIs, and other third-party services.

2.2. Economic Feasibility:

Cost-Benefit Analysis: Estimate the initial development costs, ongoing maintenance, and potential revenue streams. **Return on Investment (ROI):** Determine the expected ROI over a specific period. **Budget Constraints:** Evaluate whether the project aligns with the allocated budget.

2.3. Legal Feasibility:

Compliance with Regulations: Ensure that the application complies with data protection laws, payment processing standards (e.g., PCI DSS), and other relevant regulations. **Intellectual Property:** Check for any legal issues related to trademarks, copyrights, or patents.

2.4. Operational Feasibility:

User Adoption: Assess whether users are likely to adopt the application based on market research and user feedback.

User Training: Evaluate the ease of use and potential training needs for users, restaurant owners, and delivery personnel.

Scalability: Determine if the application can scale to accommodate growth in user numbers and transactions.

2.5. Scheduling Feasibility:

Project Timeline: Develop a realistic timeline for the development, testing, and deployment

Dependencies: Identify any dependencies on external factors, such as third-party APIs or regulatory approvals. **Milestones:** Set clear project milestones to track progress.

2.6. Risk Analysis:

Identify Risks: Identify potential risks, such as technical challenges, market competition, or changes in regulations.

Risk Mitigation Strategies: Develop strategies to mitigate identified risks and minimize their impact.

2.6.7. Market Analysis:

Target Audience: Define the target audience for the application and assess the demand for food ordering services in the target market.

Competitor Analysis: Analyze existing food ordering apps, identify their strengths and weaknesses, and determine how your app can differentiate itself.

8. Environmental Feasibility:

Sustainability: Consider the environmental impact of the application and evaluate whether sustainable practices are feasible and desirable.

Green Technologies: Explore the use of environmentally friendly technologies in the development and operation of the app.

2.9. Conclusion and Recommendations:

Feasibility Assessment: Summarize the findings of the feasibility study, indicating whether the project is feasible.

Recommendations: Provide recommendations for moving forward with the project, modifying the scope, or reconsidering based on the feasibility analysis.

Remember to involve key stakeholders, including developers, business analysts, and potential users, in the feasibility study process to gather diverse perspectives and insights.

CHAPTER 3

DATABASE DESIGN

3.1 Waterfall Model

The waterfall model is a well-known structured methodology for software development. Introduced in the 1970s, it remains one of the earliest and widely used Software Development Life Cycle (SDLC) models. This model divides the entire process of system development into distinct phases, each with a unique output. It is often referred to as "Waterfall by SDLC," emphasizing its significance in the evolution of software engineering practices.

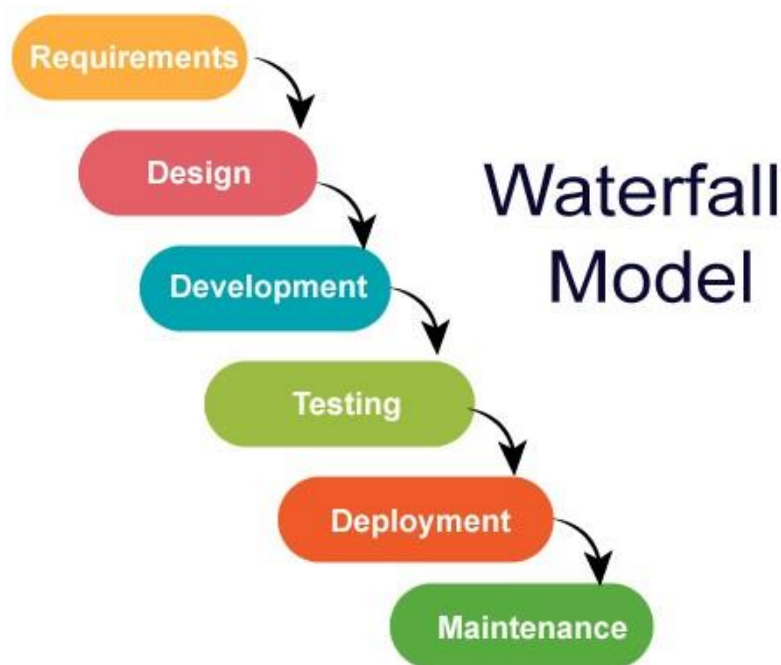


Figure 3.1. Waterfall Model

3.2 Requirements Gathering & Analysis

Here's a more specific guide for requirement gathering and analysis tailored for a food ordering Android application:

Identify Stakeholders:

- Users (customers)
- Restaurant owners
- Delivery personnel
- Administrators

Conduct Stakeholder Interviews:

- Understand user preferences and expectations.

Gather insights from restaurant owners and delivery personnel.

Document Functional Requirements:

User authentication and registration
Browse and search for restaurants and menus
Place and track orders
User profiles and order history
Payment processing integration
Reviews and ratings
Notifications

Document Non-Functional Requirements:

Performance metrics
Security measures
Usability and accessibility standards
Compatibility with Android devices

Prioritize Requirements:

Identify critical features for the initial release.

Prototyping:

Develop wireframes or prototypes for key interfaces.

Requirements Validation:

Share requirements and prototypes for validation.

Review and Approval:

Conduct formal reviews and obtain stakeholder approval.
Establish a process for handling changes to requirements.

Documentation:

Compile a concise requirements document.

By focusing on these essential steps, you can streamline the requirement gathering process for a food ordering Android application, ensuring clarity and alignment with stakeholders' needs.

3.2 ER Diagram

Designing an Entity-Relationship (ER) diagram for a food ordering application involves identifying the main entities, their attributes, and the relationships between them. Below is a simplified example of an ER diagram for a food ordering application:

A) Entities:

User:

Attributes: UserID (Primary Key), Username, Password, Email, Phone, Address.

Restaurant:

Attributes: RestaurantID (Primary Key), Name, Cuisine, Location, Contact.
Menu Item:

Attributes: ItemID (Primary Key), Name, Description, Price.
Order:

Attributes: OrderID (Primary Key), UserID (Foreign Key), RestaurantID (Foreign Key),
OrderDate, Status.

Order Item:

Attributes: OrderItemID (Primary Key), OrderID (Foreign Key), ItemID (Foreign Key), Quantity.

B). Relationships:

User-Order Relationship:

Each User can place multiple Orders.
Each Order is placed by exactly one User.
Relationship Type: One-to-Many.

User-Restaurant Relationship:

Each User can have multiple favorite Restaurants.
Each Restaurant can be a favorite for multiple Users.
Relationship Type: Many-to-Many.

Restaurant-Menu Item Relationship:

Each Restaurant can have multiple Menu Items.
Each Menu Item belongs to exactly one Restaurant.
Relationship Type: One-to-Many.

Order-Order Item Relationship:

Each Order can have multiple Order Items.
Each Order Item belongs to exactly one Order.

Relationship Type: One-to-Many.

This is a basic representation, and the actual requirements of your food ordering application may necessitate additional entities, attributes, and relationships. Consider refining the diagram based on the specific features and functionalities your application needs to support.

The notation used here includes the entity names, attribute names, primary keys, and foreign keys. Lines connecting entities indicate relationships, and the "crow's foot" notation represents the multiplicity of the relationship (e.g., one-to-many).

Keep in mind that the actual design may vary based on the complexity and specific requirements of your food ordering application.

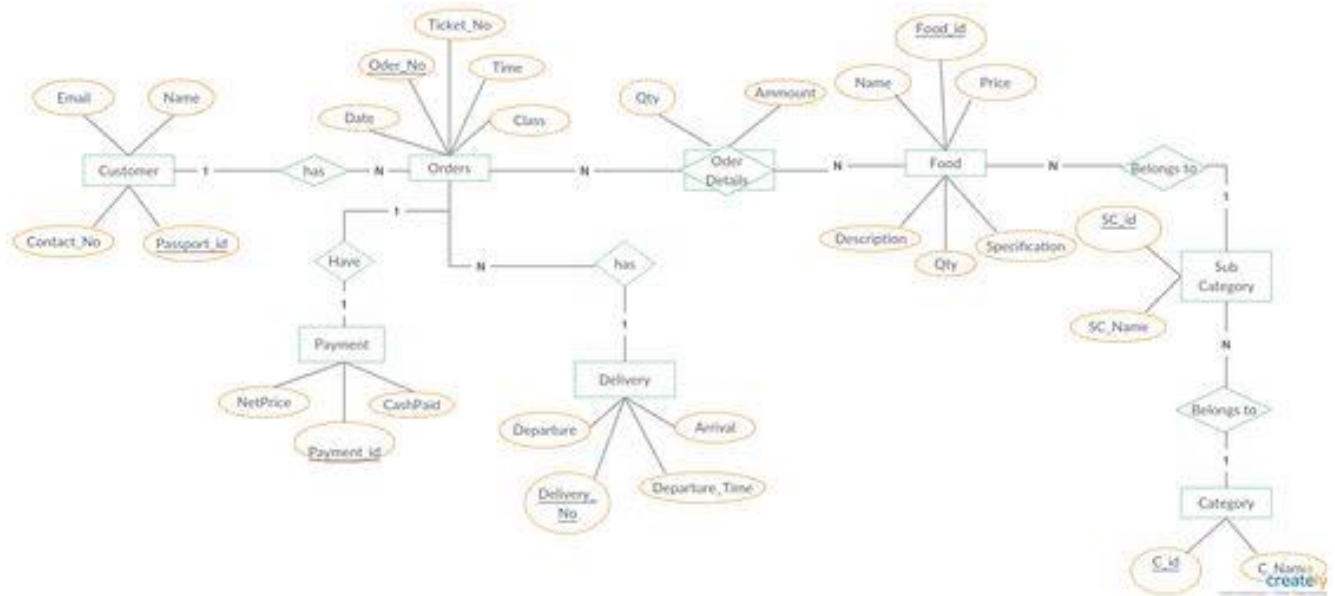


Figure 3.2 ER Diagram of Food Ordering Android Application

3.3 Use Case Diagram

A Use Case Diagram for a food ordering Android application illustrates the interactions between different actors (users and systems) and the various use cases (functionalities) provided by the application. Here's a simplified example:

A). Actors:

User: Interacts with the application to browse, order, and manage their profile.

Restaurant Owner: Manages the restaurant's menu, receives orders, and updates the order status.

B). Delivery Person: Responsible for delivering orders and updating delivery status.

Admin: Manages the overall system, monitors orders, and resolves issues.

C). Use Cases:

Browse Restaurants: Allows users to discover available restaurants.

View Menu: Enables users to see the menu of a selected restaurant.

Place Order: Users can place an order for food items.

Track Order: Provides the ability to track the status of an order.

Manage Profile: Users can update their personal information.

Add/Remove Favorites: Users can add or remove favorite restaurants.

Manage Menu (Owner): Restaurant owners can add, update, or remove menu items.

D). Confirm Delivery: Confirmation of successful delivery by the delivery person.

Manage Orders (Admin): Admins can view and manage all orders in the system.

Include Relationships:

The diagram includes relationships denoted by "<<include>>" to indicate that certain use cases (e.g., Manage Menu) are included in others (e.g., Restaurant Owner).

This is a high-level representation, and you can expand it based on the specific features and functionalities your food ordering Android application needs to support. Additionally, you might further detail each use case with specific scenarios or interactions.

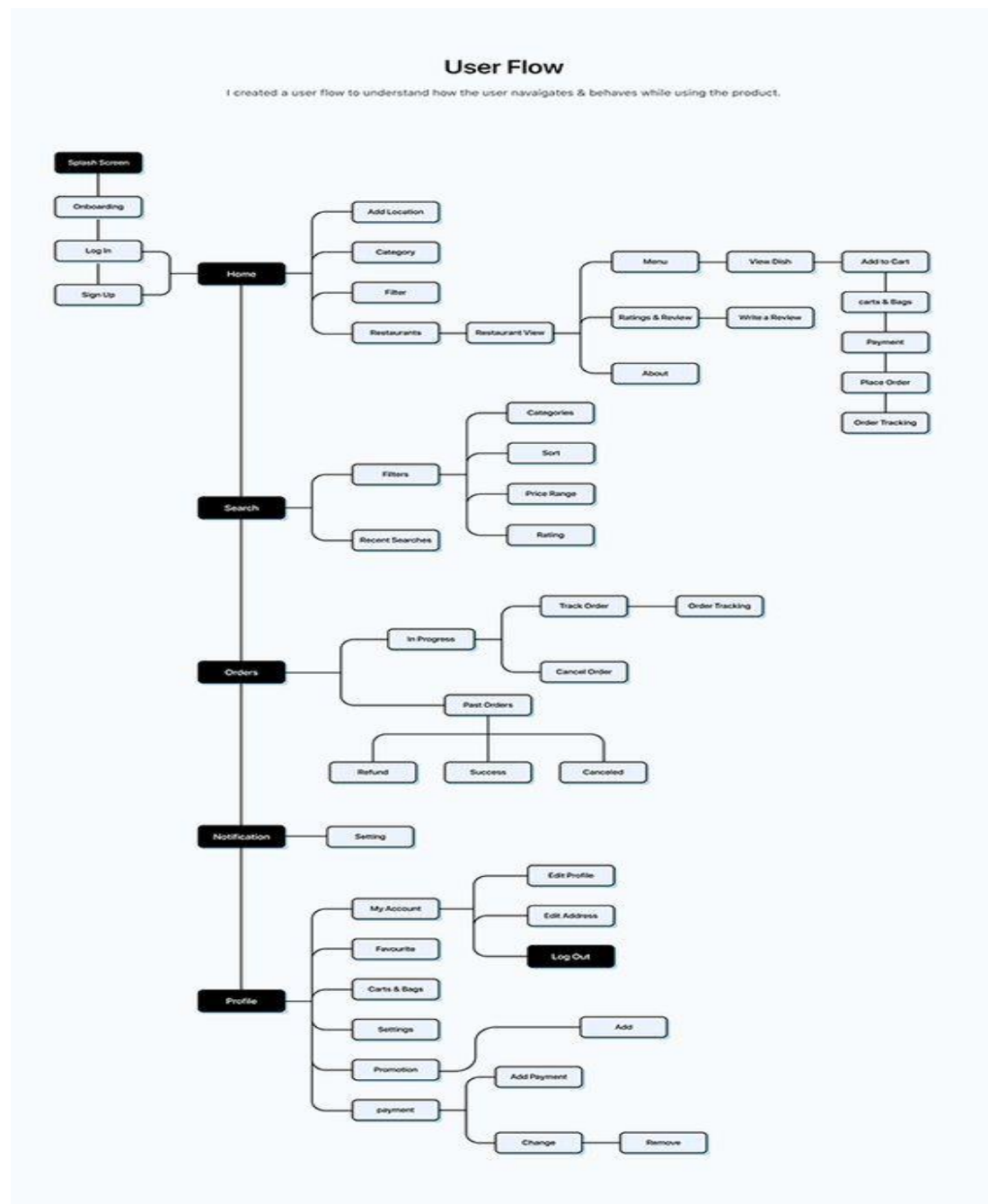


Figure 3.3 Use CASE Diagram

3.5 Activity Diagram

An activity diagram for a food ordering system provides a visual representation of the workflow and interactions between different activities or actions within the system. Below is a simplified example of an activity diagram for a food ordering system:

Explanation:

A). Start:

The starting point of the activity diagram.

B). Select Restaurant:

Users select a restaurant from the available options.

C). Browse Menu:

Users browse the menu of the selected restaurant.

D). Add Items to Cart:

Users add food items to their shopping cart.

E). View Cart:

Users review the items in their cart.

F). Place Order:

Users initiate the order placement.

G). Confirm Order:

Confirmation step to verify the order details.

Payment Process:

Users go through the payment process for the order.

Notify Restaurant:

The system notifies the selected restaurant about the new order.

H). Track Order:

Users can track the status of their order.

End:

The end point of the activity diagram.

This activity diagram provides an overview of the major steps involved in the food ordering process. Depending on the complexity of your system, you may need to include additional details, decision points, or parallel activities in the diagram. The goal is to represent the flow of activities in a clear and understandable manner.

3.5.1 State Diagram

Creating a state diagram for a food ordering Android application involves illustrating the various states the application can be in and the transitions between these states. Below is a simplified example of a state diagram for a food ordering app:

A). Start State:

Description: Initial state when the application is launched.

Transitions: Move to the "Login" state if the user needs to log in or to the "Menu" state if the user is already logged in.

Login State:

B). Description: User is required to log in.

Transitions:

Successful login transitions to the "Menu" state.

Unsuccessful login may return to the "Start" state or remain in the "Login" state.

Menu State:

C). Transitions:

Selecting a restaurant transition to the "Restaurant Details" state.

Choosing a food item transitions to the "Order Details" state.

Going to the "Cart" state to review and modify the order.

Logout transitions to the "Start" state.

Restaurant Details State:

D). Description: Display details about the selected restaurant.

Transitions:

Going back to the "Menu" state.

Selecting a food item transitions to the "Order Details" state.

Order Details State:

Description: Display details about the selected food item.

Transitions:

Adding the item to the cart transitions to the "Cart" state.

Going back to the "Menu" state.

Cart State:

E). Description: Display the items in the user's shopping cart.

Transitions:

Proceeding to checkout transitions to the "Checkout" state.

Removing items from the cart.

Going back to the "Menu" state to continue browsing.

Checkout State:

F). Description: Finalize the order and provide payment details.

Transitions:

Successful payment transitions to the "Order Confirmation" state.
 Cancelling the order returns to the "Cart" state.
 Order Confirmation State:

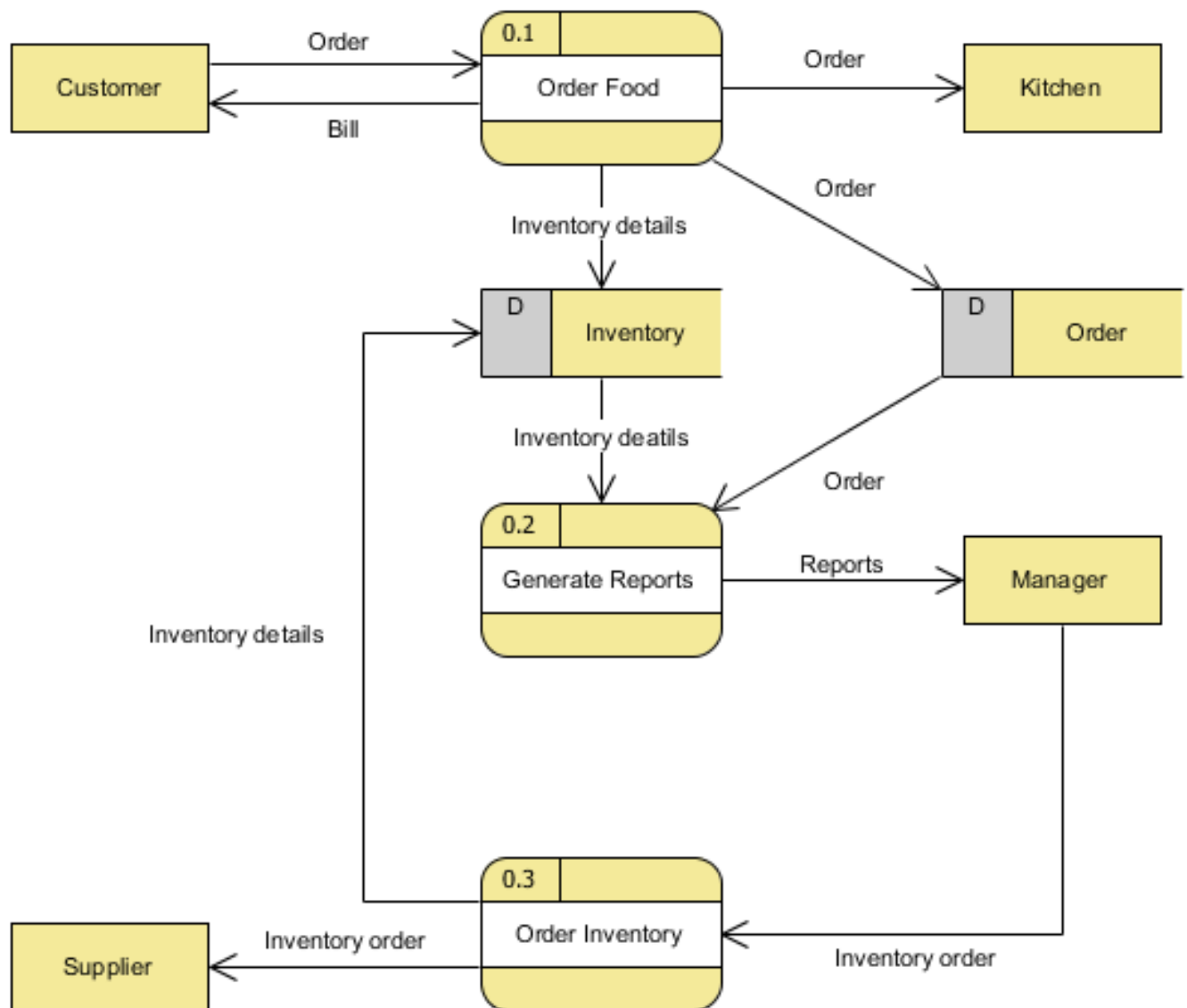


Figure 3.4 State Diagram

3.6 Component Diagram

The food ordering system comprises key components: the Client App with Order Management and Authentication, the Restaurant System handling menus and orders, a Payment Gateway for secure transactions, User Authentication for access control, and Order Management for workflow oversight.

Notifications are managed by the Notification component, while the Database component stores

user, order, and menu data. External Services interact with external entities, and Logging and Monitoring components ensure system integrity. The Feedback and Rating component enhances user experience.

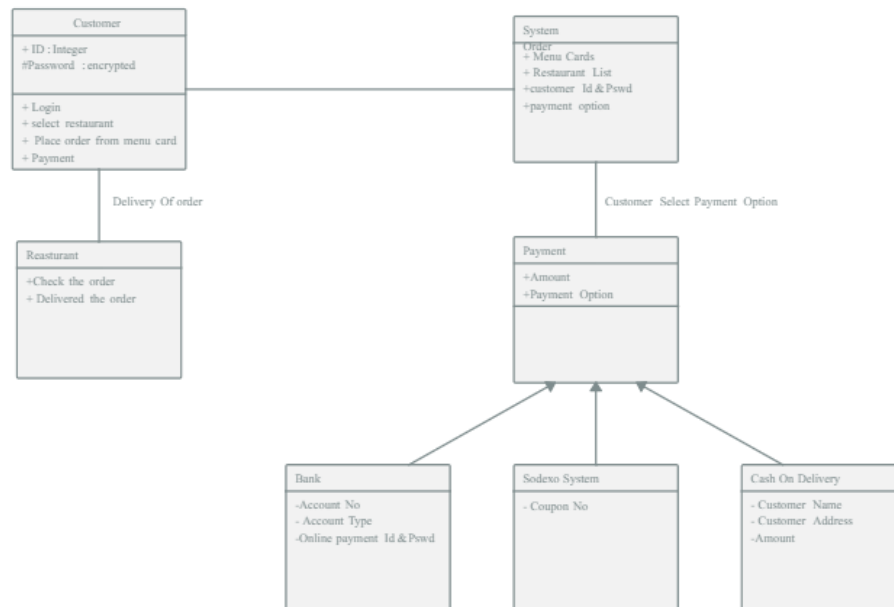


Figure 3.5 Use case diagram for Firebase backend

3.7 Deployment Diagram

Deployment Diagram For Food Order System

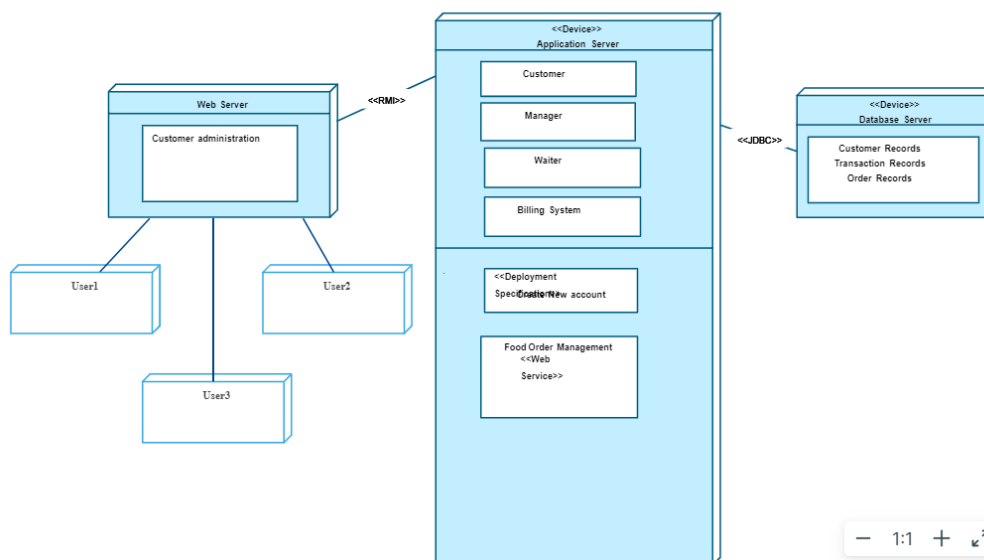


Figure 3.6 Deployment Diagram

The food ordering Android app deployment diagram features client devices running the app, communicating with an Application Server hosting business logic, User Authentication, and Order and Menu Management. A Database Server stores user and order data, while a Payment Gateway Server handles transactions. Notifications are managed by a dedicated server. External Services or APIs may be integrated, and an optional Web Server caters to web components. This visual representation illustrates the physical deployment, emphasizing the interconnectedness of user devices and backend servers for effective and seamless food ordering processes

Screenshots

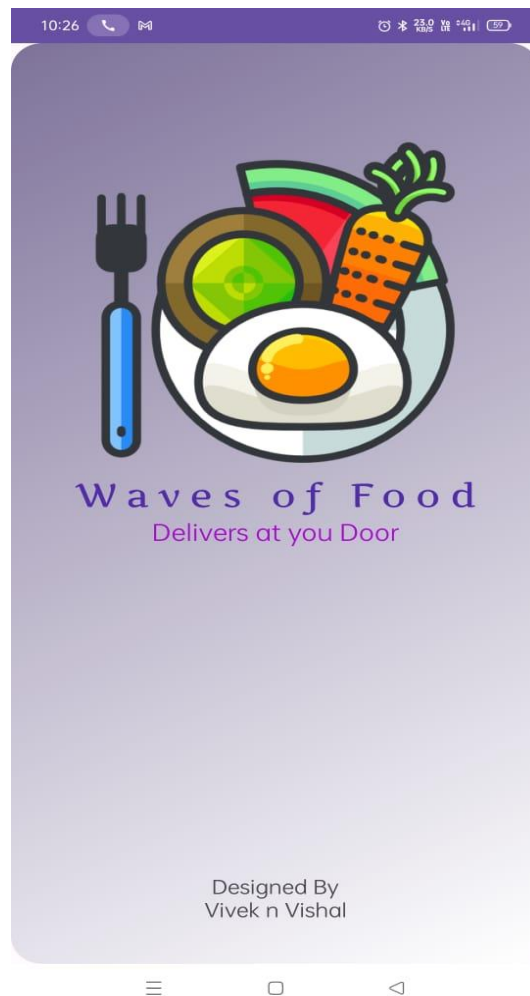


Figure 1: Splash screen



Enjoy Restaurant Quality at Home

NEXT

Designed By
Vivek n Vishal



Figure 2: First page

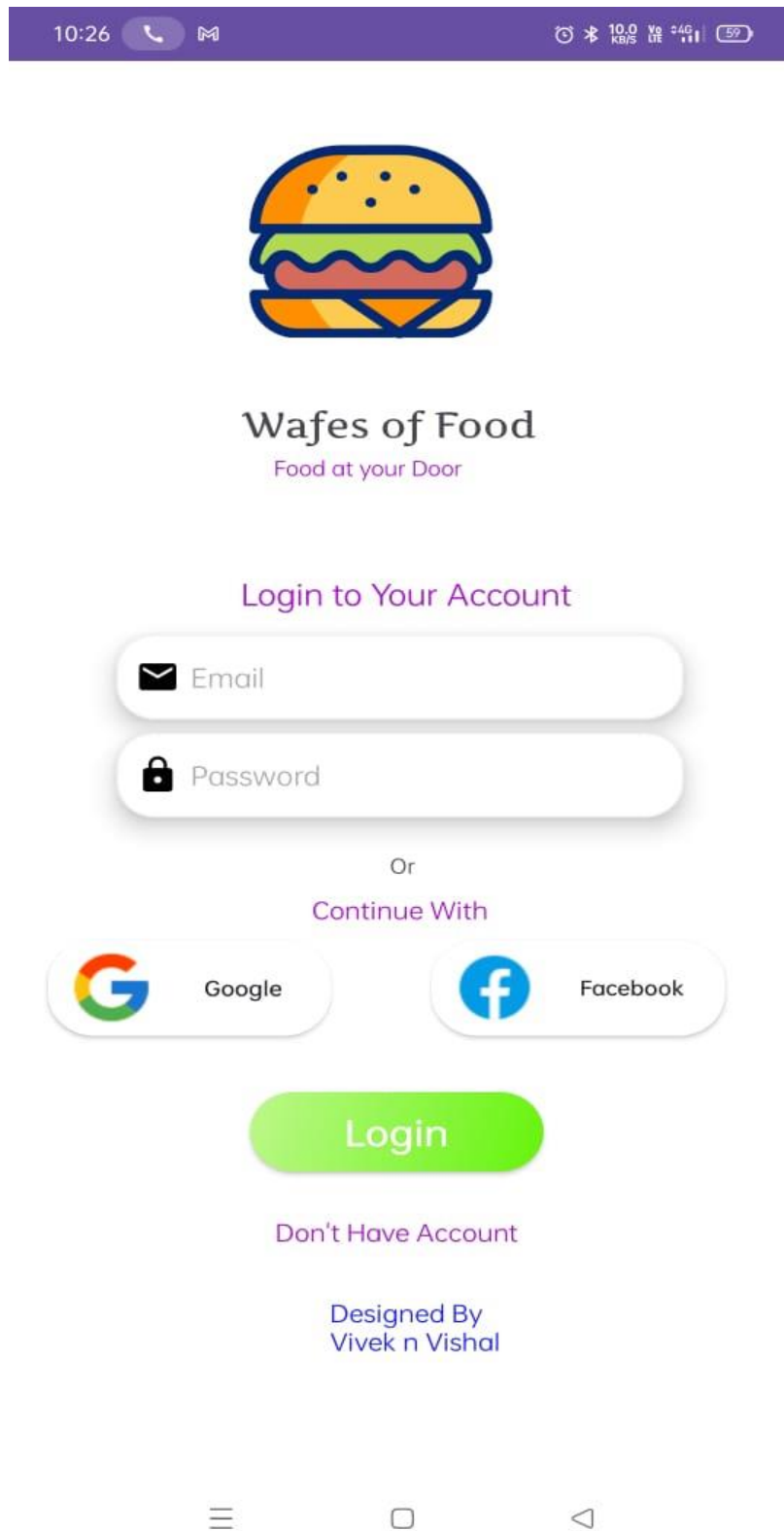


Figure 3: Login page



Waves of Food

Food at your Door

Sign Up Here



Name



Email or Phone Number



Password

Or

Sign Up With



Button



Button

Create Account

Already have an Account ?

Designed By
Vivek n Vishal



Figure 4: Sign up page

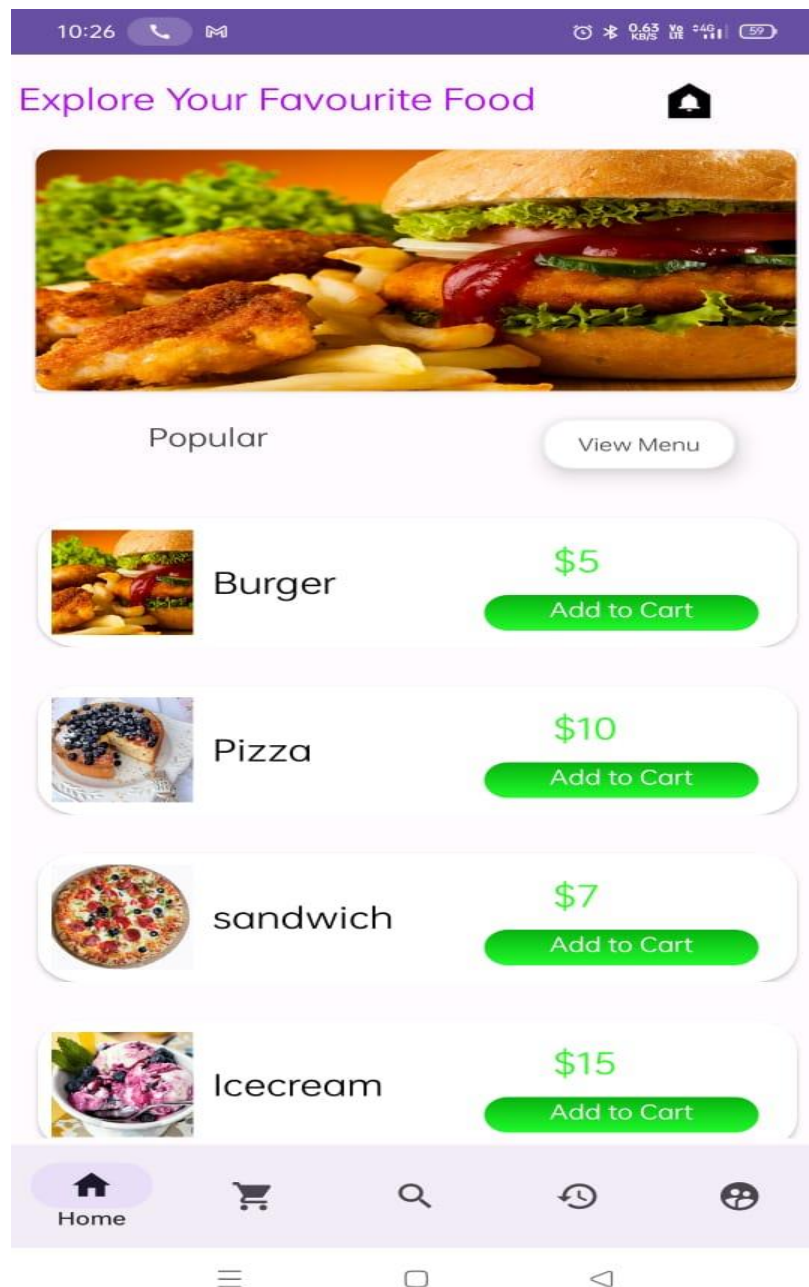


Figure 5: Menu page

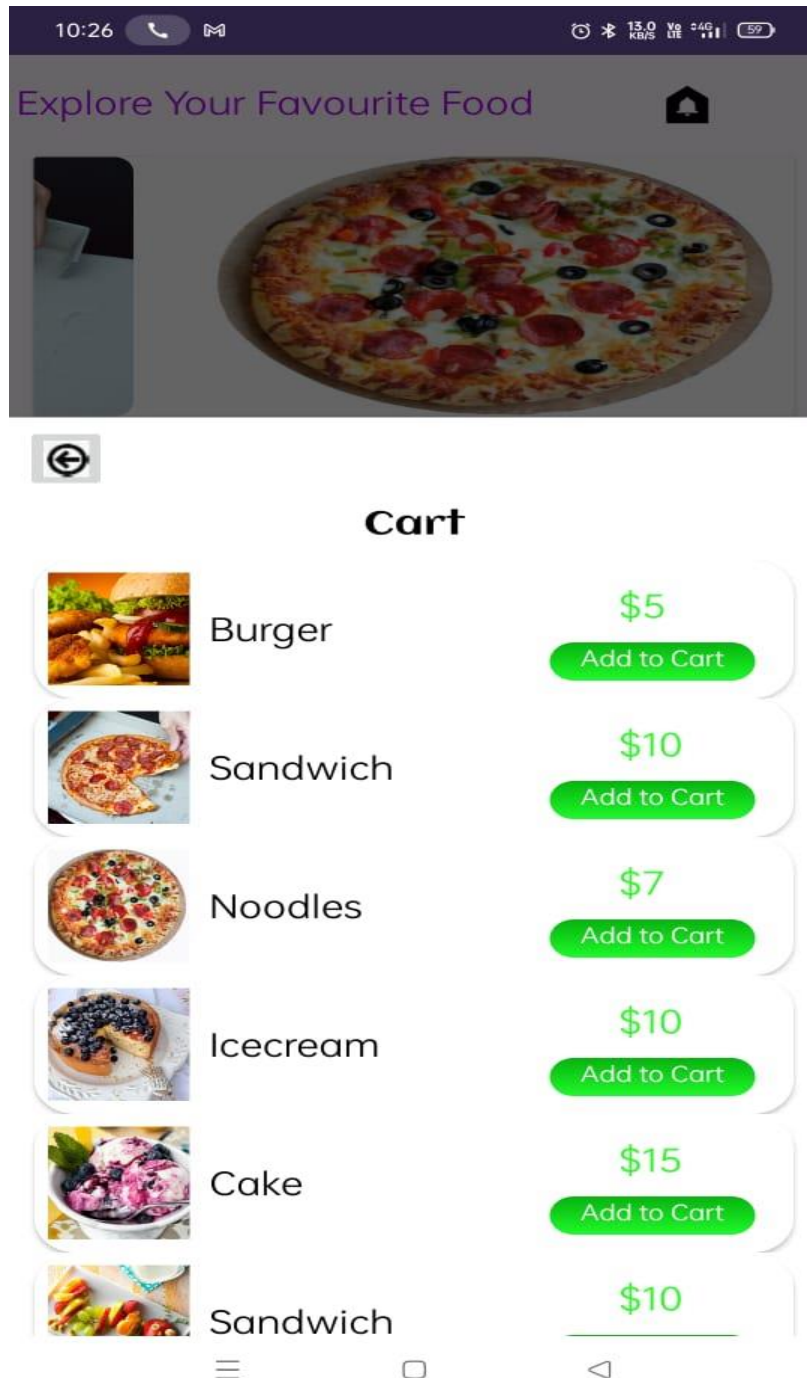


Figure 6: Menu detail page

Explore Your Favourite Food



Cart



Burger

\$5



1



Sandwich

\$10



1



Noodles

\$7



1



Icecream

\$10



1



Cake

Proceed




Cart



Figure 7: Cart page



Edit

Name	Name
Address	Name
Phone	9878787878
Payment Method	
Total Amount	\$25

Place My Order



Figure 8: Place order

ORDER PAGE

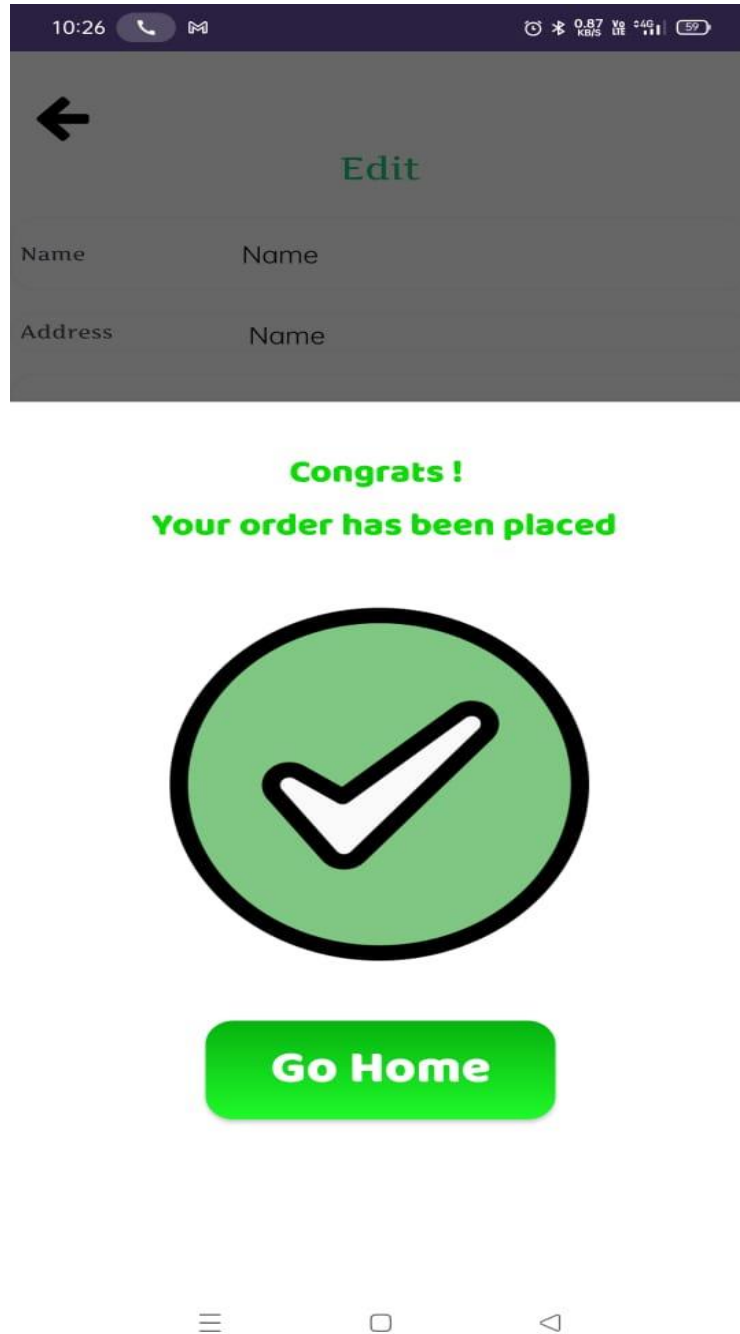


Figure 9: Success Order

Explore Your Favourite Food



Recent Buy



FoodName

\$5



Recent Buy



Food1

\$5

Buy Again



Food2

\$10

Buy Again



Food3

\$15

Buy Again



Food4

\$10

Buy Again



History



Figure 9: History page

Explore Your Favourite Food



What do you want to eat.. ?

Menu



Burger

\$5

Add to Cart



Sandwich

\$10

Add to Cart



Noodles

\$7

Add to Cart



mango

\$10

Add to Cart



Cake

\$15

Add to Cart



Sandwich

\$10

Add to Cart



\$7



Search



Figure 10: Search page

Explore Your Favourite Food



Name Admin

Address Ghaziabad

Email admin@gmail.com

Phone 9898978976

Save Information



Profile



Figure 11: Profile page

CONCLUSION

In conclusion, our endeavor to develop an online food ordering Android system represents a humble yet comprehensive effort to meet the evolving needs of users seeking a convenient and efficient way to manage their food orders. By incorporating user-friendly coding practices, we aimed to create a powerful package that addresses the specific requirements of users, particularly within the context of schools.

1. Throughout the project, we focused on several key aspects:
2. Background and Context: We provided a detailed description of the background and context of our project, highlighting its relevance to existing work in the area of online food ordering systems.
3. Aims and Objectives: Clear statements were made regarding the aims and objectives of our project, setting a framework for the development process.
4. Purpose, Scope, and Applicability: We defined the purpose and scope of our online food ordering system, outlining its applicability to the targeted user base.
5. Problem Definition: We carefully defined the problem our system addresses, establishing a foundation for the subsequent development stages.
6. Requirement Specifications: Detailed specifications of system requirements were outlined, encompassing the actions that users can perform within the system.
7. System Model: We produced a comprehensive model of the system, providing insights into the operations that can be executed, thereby enhancing our understanding of the problem domain.
8. Features and Operations: The inclusion of features and operations was meticulously detailed, accompanied by screen layouts, ensuring a user-friendly and intuitive interface.
9. User Interface Design and Security: We dedicated attention to the design of the user interface, addressing security issues to guarantee a secure and reliable user experience.
10. Implementation and Testing: The final stages of the project involved the implementation of the system, adhering to the defined requirements, and rigorous testing according to predetermined test cases.

FUTURE SCOPE OF PROJECT

- We can give more advance software for Online Food Ordering Android Application including more facilities
- We will host the platform on online servers to make it accessible worldwide. Integrate multiple load balancers to distribute the loads of the system
- Include offline chatting within a particular range
- Implement the backup mechanism for taking backup of codebase and database on regular basis on different servers

BIBLIOGRAPHY

- [1] C# 8.0 and .NET Core 3.0 – Modern Cross-Platform Development: Build applications with C#, .NET Core, Entity Framework Core, ASP.NET Core, and ML.NET using Visual Studio Code, 4th Edition
- [2] C. J. Date, A. Kannan and S. Swamynathan, An Introduction to Database Systems, Pearson Education, Eighth Edition, 2009.
- [3] Abraham Silberschatz, Henry F. Korth and S. Sudarshan, Database System Concepts, McGraw-Hill Education (Asia), Fifth Edition, 2006.
- [4] Pressman, Somerville, Software Engineering: Design, Implementation, and Management.
- [5] <https://www.tutorialspoint.com>
- [6] <https://www.javatpoint.com>
- [7] <https://www.w3school.com>
- [8] <https://www.youtube.com>
- [9] <https://www.google.com>
- [10] <https://www.wikipedia.com>
- [11] <https://www.geeksforgeeks.com>
- [12] <https://www.studocu.com>