**NEW SUMMIT COLLEGE**

*(Affiliated with Tribhuvan University)*



Final Report

on

**“FOODIE HEAVEN”**

***In Partial Fulfillment of Requirements for the bachelor’s degree in computer application***

Submitted by:

Anish Chauhan

Department of bachelor’s in computer application

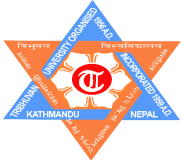
New Summit College

Under the Supervision of

**Prawesh Dhungana**

Date

December 2024



**Tribhuvan University**

**Faculty of Humanities and Social Sciences**

**New Summit College**

# Supervisor’s Recommendation

I hereby recommend that this project prepared under my supervision by ANISH CHAUHAN entitled “**FOODIE HEAVEN”** in partial fulfillment of the requirements for the degree of Bachelor of Computer Application is recommended for the final evaluation.

-----------------------

**SIGNATURE**

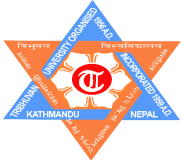
Prawesh Dhungana

**SUPERVISOR**

Department of Computer Application

New Summit College

Shantinagar, Kathmandu, Nepal



**Tribhuvan University**

**Faculty of Humanities and Social Sciences**

**New Summit College**

# LETTER OF APPROVAL

This is to certify that this project prepared by ANISH CHAUHAN entitled “**FOODIE HEAVEN”** in partial fulfillment of the requirements for the degree of bachelor’s in computer application has been evaluated. In our opinion it is satisfactory in the scope and quality as a project for the required degree.

|  |  |
| --- | --- |
| ---------------------------  **SIGNATURE**  Prawesh Dhungana  Supervisor  Department of Computer Application  Shantinagar, Kathmandu, Nepal | -------------------------  **SIGNATURE**  Chok Raj Dawadi  Principal  New Summit College  Shantinagar, Kathmandu, Nepal |
| -------------------------  **SIGNATURE**  Internal Examiner | --------------------------  **SIGNATURE**  External Examiner |

# ABSTRACT

Foodie heaven App is a mobile application designed to revolutionize the way users order food from various restaurants. Developed using Android Studio, the app utilizes Kotlin for frontend development and Firebase for backend services, including data storage and real-time synchronization. The primary objective of the app is to offer a seamless, efficient, and user-friendly platform that allows users to browse, select, and order meals with ease.

This project addresses common challenges in the traditional food ordering process, such as order inaccuracies, and limited payment options. Key features include an intuitive and visually appealing user interface,, and personalized meal recommendations based on user preferences and order history. Furthermore, the app provides functionalities for user profile management, saving Favourite orders, and offering feedback to improve service quality. The Foodie Heaven App aims to enhance the overall dining experience by providing convenience, efficiency, and a broad range of dining choices. Anticipated benefits include improved customer satisfaction, increased visibility for participating restaurants, and streamlined operations for food service providers. The project follows a waterfall model, allowing for continuous improvements and the integration of new features based on user feedback.

**Keywords**: *Mobile Application, Food Ordering, Android Studio, Kotlin, Firebase, Real-time Data, Waterfall Model*

# ACKNOWLEDGEMENT

It is a great pleasure to have the opportunity to extend our heartfelt gratitude to everyone who helped us throughout this project. We are profoundly grateful to our supervisor, **Mr. Prawesh Dhungana**, for his expert guidance, continuous encouragement and ever willingness to spare time from his otherwise busy schedule for the project’s progress reviews. His continuous inspiration has made us complete this project and achieve its target.

We would also like to express our deepest appreciation to **Mr. Chok Raj Dawadi**, Principal, New Summit College, for his constant motivation, support and for providing us with a suitable working environment.

We sincerely acknowledge direct and indirect help, suggestions and feedback offered by our colleagues before, during and after the development and implementation of the project.

At last, our special thanks go to all staff members of the BCA department at New Summit College who kindly extended their hands in making this project work a success.

Table of Contents

[Supervisor’s Recommendation ii](#_Toc184668343)

[LETTER OF APPROVAL iii](#_Toc184668344)

[ABSTRACT iv](#_Toc184668345)

[ACKNOWLEDGEMENT v](#_Toc184668346)

[LIST OF FIGURES viii](#_Toc184668347)

[LIST OF TABLES ix](#_Toc184668348)

[LIST OF ABBREVATIONS x](#_Toc184668349)

[CHAPTER 1 11](#_Toc184668350)

[INTRODUCTION 11](#_Toc184668351)

[1.1. Introduction 11](#_Toc184668352)

[1.2. Problem Statement 11](#_Toc184668353)

[1.3. Objectives 1](#_Toc184668354)

[1.4. Scope and Limitation 1](#_Toc184668355)

[1.4.1. Scope 1](#_Toc184668356)

[1.4.2. Limitation 1](#_Toc184668357)

[1.5. Development Methodology 2](#_Toc184668358)

[1.6. Report Organization 3](#_Toc184668359)

[CHAPTER 2 4](#_Toc184668360)

[BACKGROUND STUDY AND LITERATURE REVIEW 4](#_Toc184668361)

[2.1. Background study 4](#_Toc184668362)

[2.2. Literature Review 5](#_Toc184668363)

[CHAPTER 3 6](#_Toc184668364)

[SYSTEM ANALYSIS AND DESIGN 6](#_Toc184668365)

[3.1. System Analysis 6](#_Toc184668366)

[3.1.1. Requirement Analysis 6](#_Toc184668367)

[3.1.2. Feasibility Analysis 8](#_Toc184668368)

[3.1.3. Class Diagram 10](#_Toc184668369)

[3.1.4. Sequence Diagram 11](#_Toc184668370)

[3.1.5. Process modeling 12](#_Toc184668371)

[3.2. System Design 15](#_Toc184668372)

[3.2.1. Component Diagram 15](#_Toc184668373)

[3.2.2. Deployment Diagram 16](#_Toc184668374)

[3.3. Algorithm Used 17](#_Toc184668375)

[CHAPTER 4 19](#_Toc184668376)

[IMPLEMENTATION AND TESTING 19](#_Toc184668377)

[4.1. Implementation 19](#_Toc184668378)

[4.1.1. Tools Used 19](#_Toc184668379)

[4.2. Testing 20](#_Toc184668380)

[4.2.1. Test cases for Unit Testing 20](#_Toc184668381)

[4.2.2. Test cases for Integration Testing 22](#_Toc184668382)

[4.2.3. Test cases for System Testing 23](#_Toc184668383)

[4.2.4. Test cases for User Acceptance Testing 24](#_Toc184668384)

[CHAPTER 5 26](#_Toc184668385)

[CONCLUSION AND FUTURE RECOMMENDATION 26](#_Toc184668386)

[5.1. Conclusion 26](#_Toc184668387)

[5.2. Lesson Outcomes 26](#_Toc184668388)

[5.3. Future Recommendation 27](#_Toc184668389)

[References 28](#_Toc184668390)

[APPENDICES 29](#_Toc184668391)

[APPENDICES A: Screen Shots 29](#_Toc184668392)

[APPENDICES B: Source Code 33](#_Toc184668393)

# LIST OF FIGURES

[Figure 1.1: Iterative Waterfall Model 2](#_Toc183541037)

[Figure 3.1: Use Case Diagram of Foodie Heaven 7](#_Toc183541038)

[Figure 3.2: Class Diagram of Foodie Heaven 10](#_Toc183541039)

[Figure 3.3: User Sequence Diagram 11](#_Toc183541040)

[Figure 3.4: Activity Diagram 12](#_Toc183541041)

[Figure 3.4: Activity Diagram 13](#_Toc183541041)

[Figure 3.6: E-R Diagram 14](#_Toc183541041)

[Figure 3.7: Component Diagram 15](#_Toc183541042)

[Figure 3.8: Deployment Diagram 16](#_Toc183541043)

# LIST OF TABLES

[Table 3.1: Gantt Chart 9](#_Toc183540959)

[Table 4.1: Unit Testing 20](#_Toc183540960)

[Table 4.2: Integration Testing 21](#_Toc183540961)

[Table 4.3: System Testing 22](#_Toc183540962)

[Table 4.4: User Acceptance Testing 24](#_Toc183540963)

# LIST OF ABBREVATIONS

API : Application Programming Interface

PDA : Personal Digital Assistant

TAM: Technology Acceptance Model

ICT: Information and Communication Technology

LAN : Local Area Network

URL: Uniform Resource Locator

# CHAPTER 1

# INTRODUCTION

## Introduction

Foodie Heaven is a modern approach to purchasing meals and groceries through digital platforms. This process involves selecting food items from a website or mobile application, which can include prepared meals ready for consumption (such as dishes from a restaurant) or raw food products (like farm-fresh vegetables or frozen meats). The primary objective of creating a Foodie Heaven system is to digitize and streamline the traditional manual method of placing orders, enhancing efficiency and accuracy in order management.

The implementation of the Foodie Heaven system has vast potential. It provides a versatile solution for restaurants, fast food chains, and home kitchens to manage customer orders effectively. This system ensures a quick, accurate, and user-friendly experience, reducing the need for physical storage space due to its digital nature. By utilizing Firebase for backend services, the system guarantees robust data security and minimizes the risk of data loss, ensuring reliability and trustworthiness.

Foodie Heaven process starts with customers selecting their favourite restaurant through the app. They can browse the menu, pick their items, and choose between delivery or pickup. This system enhances customer convenience and transparency while helping restaurants streamline operations, reduce errors, and boost efficiency. It transforms traditional food ordering into a seamless digital experience tailored to modern consumer needs.

## Problem Statement

* In the fast-paced world we live in, people often lack the time to prepare meals at home and rely heavily on food delivery services.
* Traditional methods of ordering food can be inefficient and time-consuming, leading to frustration for both customers and food service providers.
* There is a need for a streamlined, user-friendly mobile application that allows users to order food conveniently and specify delivery times according to their schedules.

## Objectives

The objectives of the system are as follows:

* + - To Develop a mobile application with an intuitive and easy-to-navigate interface.
    - To Enable users to browse menus, select items, and place orders efficiently.
    - To Provide basic search and filtering options to help users find dishes.
    - Keep a simple record of users’ past orders for easy reordering.

## Scope and Limitation

### Scope

The project focuses on the core functionalities of food ordering. It includes:

* User registration and login
* Browsing restaurant menus with pictures and descriptions
* The app will use a lightweight database (e.g. Firebase) for managing local data.
* A simple user profile will store preferences and order history.

### Limitation

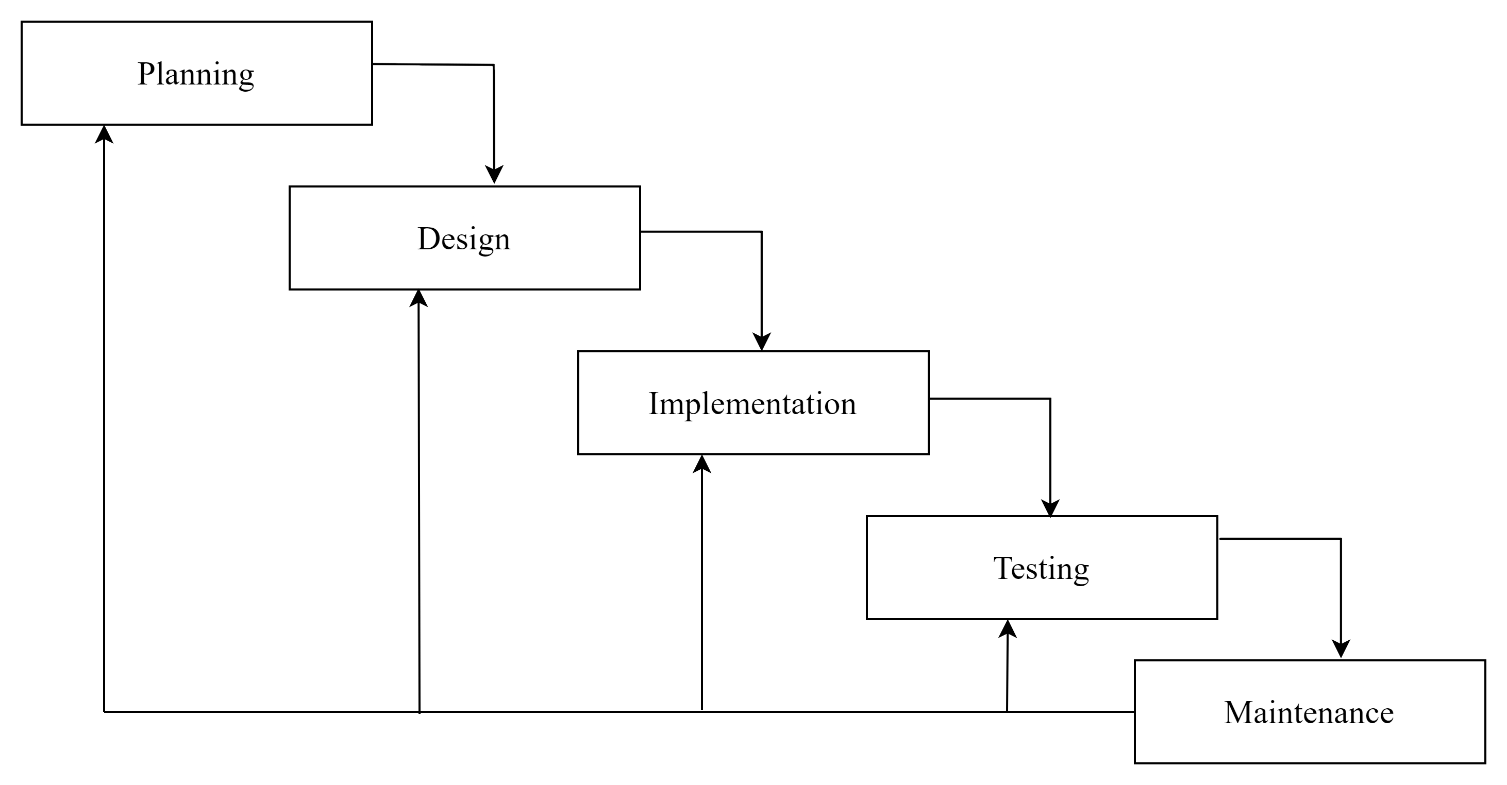
The limitations of the project are:

* + - * The app may require an active internet connection for features like browsing updated menus or placing orders.
* The app does not provide real-time updates on the status of orders (e.g., order confirmation, preparation, or delivery tracking).
* The app does not support online payment options; all payments must be made offline (e.g., cash on delivery).

## Development Methodology

Iterative Waterfall Model

For this project we have chosen the Waterfall model. It is a traditional and widely used methodology for software development. It is a linear and sequential approach, where each phase of the development process must be completed before the next phase begins. Using the in the project provides a clear, structured, and manageable approach to software development. With well-defined phases, thorough documentation, and a focus on stable requirements, the Waterfall Model ensures that each aspect of the project is meticulously planned and executed, leading to a high-quality and reliable final product



**Figure 1.1: Iterative Waterfall Model**

## Report Organization

This report is organized into 5 chapters:

* **Chapter 1: “Introduction”-** This chapter introduces the problem statement, objectives and limitations of the project.
* **Chapter 2: “Requirement and Feasibility Analysis”-**This chapter describes about the functional and non-functional requirements, economic feasibility, technical feasibility, operational feasibility and scheduling feasibility.
* **Chapter 3: “System Design”-**This chapter introduces about the system and interface design of the project app.
* **Chapter 4: “Implementation and Testing-** This chapter clearly illustrates the methods and tools used to implement the project.
* **Chapter 5: “Conclusion and Future Works”-** This is the final chapter that concludes the project and talks about our future plans with the project.

# CHAPTER 2

# BACKGROUND STUDY AND LITERATURE REVIEW

## 2.1. Background study

Our analysis drew insights from several research papers pivotal to shaping the development of the Foodie Heaven project:

In one study, a wireless meal ordering system was crafted to incorporate real-time customer feedback within restaurant environments. This innovative system leverages smartphone technology in Wi-Fi settings, allowing restaurant operators to dynamically adjust menu presentations and engage directly with patrons [1].

Another research focused on understanding the factors influencing internet users' perceptions of online food ordering, particularly among university students in Turkey. Applying Davis’s Technology Acceptance Model (TAM) alongside additional factors like Trust, Innovation, and External Influences provided a comprehensive analysis of adoption behaviors in web-based food ordering [1].

Furthermore, efforts to automate restaurant meal ordering processes were explored through the development of a mobile Android application. This system facilitates seamless wireless communication between customer devices and restaurant servers, ensuring swift updates to a central database for efficient management and timely menu adjustments.

Additionally, restaurant owners' initiatives to enhance dining experiences using ICTs such as PDAs and wireless LANs were investigated. A cost-effective touch screen-based restaurant management system, proposed as an alternative to traditional paper-based and PDA-based systems, aimed to streamline operations and improve customer service efficiency [2].

Lastly, a user-centered design approach was emphasized in the development of an online food ordering system, addressing critical service issues and enhancing user experience. This system simplifies order placement, provides comprehensive information to customers, manages order data effectively, and supports administrative tasks for food service operations.

## 2.2. Literature Review

The shift towards computerized systems has revolutionized management practices, particularly in the domain of online food delivery services. Unlike traditional methods reliant on physical records vulnerable to loss, modern systems leverage robust databases for secure and efficient data storage. This advancement not only ensures reliability but also enhances user-friendliness, effectively resolving challenges inherent in manual management systems.

**Foodmandu;**

Established in 2010 by Manohar Adhikari, FoodMandu is a pioneering online food delivery service in Nepal. It revolutionizes the dining experience by connecting customers with a diverse range of restaurants through its web and mobile platforms. Foodmandu offers a seamless ordering process, allowing users to browse menus, place orders, and track deliveries in real-time. With a commitment to customer satisfaction, Foodmandu ensures prompt and reliable delivery services, making it a preferred choice for food enthusiasts across Nepal.

**Bhojdeal;**

Bhojdeal now known as Bhoj, is a popular Nepali online food ordering app focused on the Kathmandu area. It offered a convenient way for users to order food delivery from a variety of restaurants. You could browse menus with pictures of dishes, place orders, and even track your food in real-time. Bhoj currently does not offer scheduled delivery. The ordering process is simple and intuitive, allowing users to place orders swiftly and track their food delivery in real time, ensuring transparency throughout the process.

# CHAPTER 3

# SYSTEM ANALYSIS AND DESIGN

## 3.1. System Analysis

The online food delivery software is designed to facilitate seamless ordering and delivery processes for customers and restaurant administrators. This system aims to streamline the food ordering experience by providing a user-friendly interface and efficient backend management. Restaurant administrators can manage orders, menus, and customer feedback effortlessly. The system ensures real-time updates and secure handling of customer information.

### 3.1.1. Requirement Analysis

The term requirements determination explains the overall things that can be done within the

system in simpler manner. The requirements for the system can be termed as functional and

no- functional requirements.

#### Functional Requirements

* The customers should be able to register and login into the system.
* Users should be able to browse restaurants, view menus, and place orders.
* Users can rate their experience and provide feedback.
* The admin shall be able to update the order status.
* The admin shall be able to edit the order status.
* The admin shall be able to view the order status.

**Use Case Diagram**

A diagram of a structure

Description automatically generated

**Figure 3.1: Use Case Diagram of foodie Heaven**

#### ii. Non-Functional Requirement

* + Security: The system is secure which lets only authentic users.
  + Appearance: It has simple, user-friendly, and attractive to users.
  + Architecture: The architecture of this system is highly optimized and well managed.
  + Speed: It is a light application that uses minimum bandwidth and provides high-speed accessibility.

### 3.1.2. Feasibility Analysis

Once the problem is identified, the next step is to conduct a feasibility study, which is a high- level capsule version of the entered systems and design process. The objective is to determine whether the proposed system is feasible. The three tests of feasibility have been carried out.

#### Technical Feasibility

The proposed "Online Food Delivery" app will be developed using Java and Kotlin languages for Android development, with Firebase backend services for data storage and synchronization. Both Java and Kotlin are widely supported in the Android development community, ensuring that existing technologies can adequately support the app's development. Required hardware and software, including Android Studio and Firebase tools, are readily available and capable of supporting the app's development and implementation. Therefore, the solution is technically feasible.

#### Operational Feasibility

The app is designed to be user-friendly, with a straightforward interface for both customers and restaurant administrators. Minimal training will be required for users to navigate the app effectively, ensuring operational feasibility. The system's interoperability and ease of use are expected to enhance user satisfaction and service quality, contributing to its operational feasibility.

#### iii. Economic Feasibility

The "Online Food Delivery" app is being developed as part of a college project, which means there are no direct costs associated with software acquisition. Development costs are primarily related to time and effort invested by the development team. The benefits of the app include enhancing user convenience and operational efficiency. Given the project's educational context, where resources are already available, the app is economically feasible.

#### iv. Schedule Feasibility

The project will follow a phased approach with clear milestones for each stage: planning, design, development, testing, and maintenance. Using Iterative Waterfall methodology allows for iterative development and continuous integration of user feedback, ensuring timely progress and flexibility to adapt to changes.

**Table 3.1: Gantt Chart**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Working Time | 15th July | 19th Aug | 25th Sept | 15th Oct | 8th Nov | 28th Nov |
| Requirements |  |  |  |  |  |  |
| System Design |  |  |  |  |  |  |
| Coding |  |  |  |  |  |  |
| Implementation |  |  |  |  |  |  |
| Testing |  |  |  |  |  |  |
| Documentation |  |  |  |  |  |  |

### 3.1.3. Class Diagram

A black background with white text

Description automatically generatedThe main purpose of class diagram for our project is to build a static view of an application. It is a blueprint of a system as it visualizes the particular working functionality of our system. The UML diagram of FOODIE HEAVEN System only depicts the abstract view of the diagram.

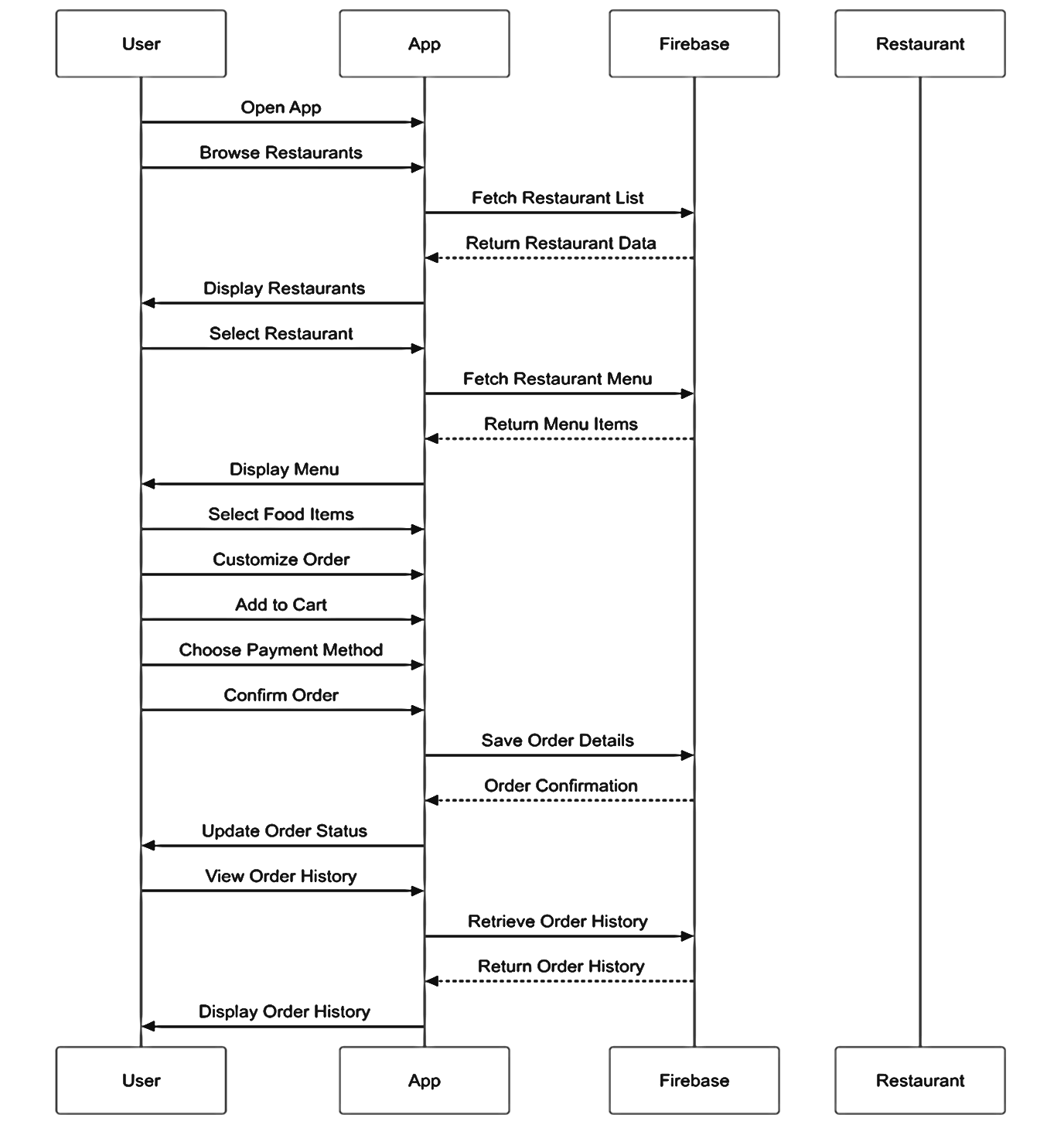
**Figure 3.2: Class Diagram of Foodie Heaven**

**Figure 3.1.Class Diagram**

### 3.1.4. Sequence Diagram

In the FOODIE HEAVEN app, the sequence diagram dynamically models the interactions between various objects within the system in a sequential order.

**User Sequence Diagram**



**Figure 3.3: User Sequence Diagram**

### Process modeling

While developing a system to ensure that all requirements have been met by the system and the users are satisfied with how the system is being implemented, we’ve done process modeling. This modeling is supposed to help the system is created in the most logical manner and will be able to be used for a long time.

**Activity Diagram**

A black background with a black square

Description automatically generated with medium confidence

**Figure 3.4: Admin Activity Diagram**

Initially to visit the food categories or food menu, users don’t need to login/register an account. After checking out the categories and menu items, if the user finds his/her desired menu and if they want to order that item, they can go to the order page. During placing any order, the customer needs to provide his/her required information mentioned the order section

A black background with a black square

Description automatically generated with medium confidence

**Figure 3.5: Customer Activity Diagram**

**E-R Diagram**

A black background with a black square

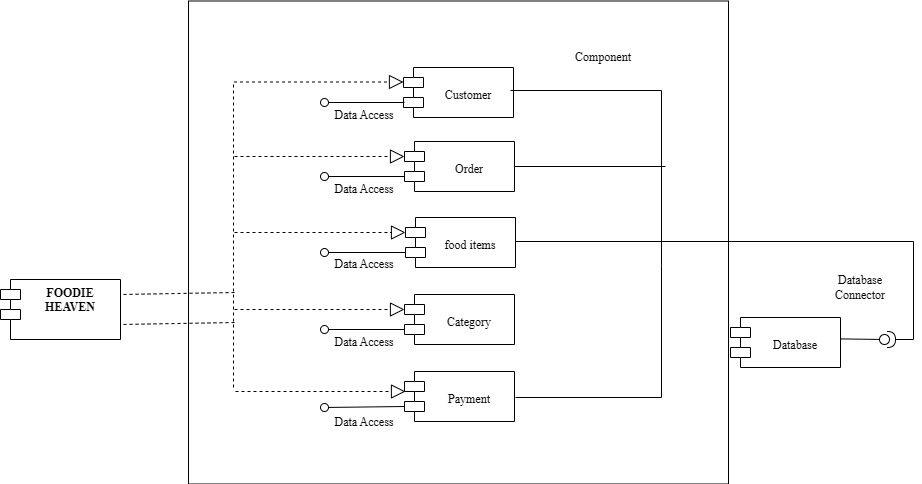
Description automatically generated with medium confidence

**Figure 3.6: E-R Diagram**

## System Design

### Component Diagram

The Foodie Heaven app's component diagram reveals a sophisticated architectural design with key components strategically interconnected. Critical modules include User Interface, User Management, Restaurant and Menu Management, Order Processing, Recommendation Engine, and Data Storage. Each component serves a specific function, from managing user interactions and authentication to processing orders, providing personalized recommendations, and maintaining robust data infrastructure. The modular architecture ensures seamless communication between components, enabling a comprehensive and efficient food ordering experience that prioritizes user convenience, personalization, and system reliability.



**Figure 3.7: Component Diagram**

### Deployment Diagram

The deployment diagram for FOODIE HEAVEN visualizes the physical deployment of the app’s software components across different nodes in the network. It shows how the application is distributed across client devices and server infrastructures. For instance, the user interface and front-end components are deployed on user devices, while the backend services like the Content Retrieval Service, User Management Service, and Database Server are hosted on cloud servers. This diagram also depicts the connections between these nodes, highlighting the communication pathways and network configurations, ensuring efficient deployment and scalability of the app.

A white cube with black text

Description automatically generated

**Figure 3.8: Deployment Diagram**

A deployment diagram helps by providing a clear picture of how the system is structured in terms of its physical components. It shows how the “User Site", "Admin site", "server" and "Firebase Server" are interconnected, helping developers and stakeholders understand how data flows between them and how the system operates in a real-world environment. This visualization aids in system design, troubleshooting, and scalability planning, ensuring that the Foodie Heaven can effectively meet its goals while maintaining security and performance.

## Algorithm Used

**RECOMMENDATION ALGORITHM**

The **Recommendation Algorithm** in the FOODIE HEAVEN App helps provide personalized and meaningful suggestions to enhance the user experience. It analyzes user input, behavior, and patterns to deliver accurate and relevant recommendations.

**CONTENT-** **BASED FILTERING ALGORITHM FOR FOODIE HEAVEN APP**

Content-Based Filtering (CBF) is a recommendation algorithm that personalizes food suggestions by analyzing the attributes of food items and matching them to a user's preferences. In the context of the Foodie Heaven app, each food item can be represented by attributes such as cuisine type, ingredients, spice level, dietary restrictions, and restaurant origin.

The algorithm builds a user profile based on dishes the user has previously ordered or interacted with. By identifying and analyzing the attributes of these dishes, the system calculates the similarity between these dishes and others in the database. It then recommends food items with the most similar attributes. This makes CBF highly effective for providing tailored recommendations, especially when a user shows specific interests like searching for dishes from a particular cuisine or with specific dietary requirement**s.**

**COSINE SIMILARITY**

Cosine similarity is a metric used to measure the similarity between two non-zero vectors in an n-dimensional space. It is often employed in various applications such as information retrieval, document similarity analysis, and recommendation systems. The cosine similarity algorithm calculates the cosine of the angle between two vectors and yields a value between -1 and 1, where a higher value indicates greater similarity.

similarity = dot\_product(user\_profile, book\_features) / (

magnitude(user\_profile) \* magnitude(book\_features)

)

Where:

* dot\_product combines weighted features (food \* 4.0 + meat \* 2.5 + course \* 2.0)
* magnitude normalizes the vectors to ensure fair comparison regardless of vector size

1. Diversity Enhancement:

* Applies a diversity factor (0.7) for repeated authors
* Filters recommendations with similar scores > 0.1
* Returns top 10 most similar books

For example:

• When a user interacts with "COLLEEN HOOVER" books:

User Vector = {

writer\_colleen\_hoover: 4.0 \* decay \* boost,

genre\_romance: 2.5 \* decay \* boost,

course\_literature: 2.0 \* decay \* boost

}

The cosine similarity then finds books with the most similar feature vectors.

This cosine similarity approach ensures that recommendations are:

* Normalized across different feature scales
* Independent of the magnitude of user interactions
* Mathematically sound in measuring content similarity

The implementation balances mathematical precision with practical considerations for generating relevant and diverse recommendations.

# CHAPTER 4

# IMPLEMENTATION AND TESTING

## 4.1. Implementation

Software implementation is the process of converting the designed system into operational programs. This process includes not only writing the code but also preparing requirements and objectives, designing what is to be coded, and confirming that what is developed meets predefined objectives.

### 4.1.1. Tools Used

**Android Studio**

It provides the development environment for building Android apps. Android Studio is the primary recommended platform for Android app development, offering features that streamline the entire process from coding to publishing apps on the Google Play Store.

**Firebase**

Firebase was implemented to handle secure user authentication and real-time database management, offering reliable backend services for managing user data. Its integration helped streamline the user login process and ensure data consistency across multiple devices. However, due to the premium cost of Firebase Storage, alternative solutions were adopted for storing media files while retaining Firebase for critical functionalities.

**Kotlin**

Kotlin are programming languages used for Android app development**.** Kotlin is designed to be more expressive and safer than Java, with features like null safety, extension functions, and more compact syntax. It's particularly popular for Android app development, server-side development, and has growing support in web and multiplatform scenarios**.**

**Imgur:**

Imgur was used as an alternative platform for storing and managing image files, providing a cost-effective solution compared to Firebase Storage. It enabled quick uploads and retrieval of images via URLs, ensuring seamless integration of media files into the application while maintaining a smooth user experience.

**Draw.io**

Draw.io is free online diagram software for making Class diagram, Sequence diagram, Component and Deployment diagram. It is an open-source technology stack & most widely used browser-based end-user diagramming application.

## 4.2. Testing

Once source code has been generated, software must be tested to correct as many errors as possible before delivery to customer. Our goal is to design a series of test cases that have a high likelihood of finding errors. Following testing techniques are well known and the same strategy is adopted during this project testing.

**Unit Testing**

Unit testing focuses on verifying the smallest unit of software design - the individual components or modules. It is white-box oriented, meaning tests are based on the internal workings of the application. For the FOODIE HEAVEN app, unit tests are implemented for each module by providing correct manual inputs. These tests ensure that data is stored in the database and retrieved correctly. If a module requires user input or produces output, any errors encountered during this process are handled by specific error handlers, which provide detailed information about the type and cause of the error. This ensures that each component functions correctly in isolation, facilitating easier identification and fixing of bugs.

### 4.2.1. Test cases for Unit Testing

**Table 4.1: Unit Testing**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **S.N.** | **Test Case**  **ID** | **Test Description** | **Steps Executed** | **Expected**  **Results** | **Actual Results** | **Pass/**  **Fail** |
| i) | TC-01 | Login Functionality Test | 1. Create a Login button.  2. Bind the login Button and add setOnClickListener. | Login should be successful with proper validation and go to Main Activity. | Login was successful, with Home Page open.  . | Pass |
| ii) | TC-02 | Logout | Click the  Account | Should be redirected to the login page | Was  redirected to the login page. | Pass |
| iii) | TC-03 | Search Functionality | 1. Retrieve menu items from database. | Returns the search food items. | Retrieve menu items from the database. | Pass |
| iv) | TC-04 | Purchase History | 1. Create listOfOrderItem ().  2. Call retrieveBuyHistory (). | Returns recent buy food items. | Display recent buy food item. | Pass |

**Integration Testing**

Integration testing focuses on verifying the interactions between integrated modules. In the FOODIE HEAVEN app, once individual modules pass unit testing, they are integrated to work together. Tests are conducted to ensure that data flows correctly between components like the user interface, content retrieval system, translation service, and database. These tests identify issues that might arise from the integration of different modules, such as data misalignment or communication errors. By simulating real-world use cases, integration testing ensures that combined components work together seamlessly.

### 4.2.2. Test cases for Integration Testing

**Table 4.2: Integration Testing**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **S.N.** | **Test Case**  **ID** | **Test Description** | **Steps Executed** | **Expected**  **Results** | **Actual Results** | **Pass/**  **Fail** |
| i) | TC-01 | User registration and Login | 1. Navigate to login activity.  2. Enter valid credentials.  3. Click "Login". | Login should be successful and go to the Main Activity. | Login was successful, and the  Main Activity entered. | Pass |
| ii) | TC-02 | Home Page to Search Functionality | 1. Navigate to Home Page  2.Search results . | Search results matching the term should be displayed. | Search results displayed successfully. | Pass |
| iii) | TC-03 | Search to Add to Cart Workflow | 1.Perform search.  2. Select food item and add to cart. | Selected food item should be added to the cart successfully. | Selected food item added to the cart. | Pass |
| iv) | TC-04 | Purchase History Validation | 1.Complete an order.  2.Navigate to the history  . | Recent Purchased should appear in history. | Recent Purchased displayed in the history | Pass |

**System Testing**

System testing involves testing the complete and integrated application to verify that it meets the specified requirements. For FOODIE HEAVEN, this includes testing the entire application workflow, from user authentication and content retrieval to word searches, translations, and quizzes. System testing ensures that all components function together as a single system and that the application performs as expected under various conditions. This phase also includes performance testing to verify that the app can handle high loads and stress testing to identify its breaking points.

### 4.2.3. Test cases for System Testing

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **S.N.** | **Test Case**  **ID** | **Test Description** | **Steps Executed** | **Expected**  **Results** | **Actual Results** | **Pass/**  **Fail** |
| i) | TC-01 | Complete user login Flow | 1. Launch app & enter credentials.  2. Click login.  3. verify Home Page display. | User should successfully log in and be navigated to the Home Page. | Successfully logged in and navigated to the Home Page. | Pass |
| ii) | TC-02 | Invalid Login Attempt | 1. Open app.  2. Enter invalid credentials.  3.Click Login. | App should display an error message “login failed” | Error message displayed correctly. | Pass |
| iii) | TC-03 | Search Food and display Results | 1. Log in.  2. Enter a search term in the search bar.  3. Verify result. | Search results should match the query and display correctly. | Search results displayed successfully. | Pass |
| iv) | TC-04 | Purchase History Display | 1. Complete an order.  2. Navigate to History section | Recent order should appear in the History list with accurate detail. | Recent order displayed correctly in History. | Pass |

**Table 4.3: System Testing**

**User Acceptance Testing**

User Acceptance Testing (UAT) is the final phase of testing, where the application is tested in a real-world environment by end users. For FOODIE HEAVEN, UAT involves providing the app to a group of users who interact with its features and provide feedback. This phase ensures that the app meets user expectations and requirements. UAT focuses on usability, functionality, and user experience. Any issues identified during this phase are addressed before the final deployment. This step ensures that the app is ready for widespread use and satisfies the needs of its intended audience.

### 4.2.4. Test cases for User Acceptance Testing

**Table 4.4: User Acceptance Testing**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **S.N.** | **Test Case**  **ID** | **Test Description** | **Steps Executed** | **Expected**  **Results** | **Actual Results** | **Pass/**  **Fail** |
| i) | i) | TC-01 | User Login and Navigation | 1. Open app.  2. Register a new user.  3. Login with the new credentials. | User should be able to register and log in successfully. | Pass |
| ii) | ii) | TC-02 | Search for Food Items | 1.Log in.  2. use search bar to look for specific items. | User should see a list of matching food items, with accurate detail. | Pass |
| iii) | iii) | TC-03 | 1. Add to Cart and View Cart | 1. Select food items from search results. 2. Add them to cart.   Open the cart. | Food items added should be displayed in the cart. | Pass |
| iv) | iv) | TC-04 | 1. Logout Functionality | 1. Log in. 2. Navigate to the Profile.   Click on Logout. | User should be redirected to the Login page, and no session data should be active. | Pass |

# CHAPTER 5

# CONCLUSION AND FUTURE RECOMMENDATION

## 5.1. Conclusion

To conclude, the development of our Foodie Heaven app using Android Studio, Kotlin, and Firebase marks a transformative approach to modern food ordering technology. The project was carefully crafted to meet user needs and overcome limitations in existing food ordering platforms. It demonstrates a dedicated effort to enhance user convenience, operational transparency, and system efficiency for both customers and restaurant management. The app represents an innovative solution that reimagines the food ordering experience through advanced technological integration, offering robust features that enable personalized interactions. By prioritizing user-centric design and cutting-edge technological implementation, Foodie Heaven has the potential to significantly improve how people engage with food ordering platforms.

## 5.2. Lesson Outcomes

Through the development of the FOODIE HEAVEN project, we have gained several key insights and outcomes. Technically, we have mastered the use of Kotlin to build applications, enabling us to deliver a seamless experience across multiple devices. We've learned the importance of a user-centered design approach, ensuring that the app is intuitive and meets the needs of its diverse audience. The project underscored the necessity of effective content management strategies to handle diverse data such as words, proverbs, and quizzes. We also recognized the value of implementing comprehensive testing methodologies, including unit, integration, system, and user acceptance testing, to ensure the app's functionality and reliability.

## 5.3. Future Recommendation

1. **Payment Integration:**

* Integrate popular Nepali payment gateways like **eSewa,** **Khalti,** and **IME Pay** for seamless online transactions.

1. **Enhanced Food search:**

* Implement **auto-complete text suggestions** in the search bar to help users find food items quickly.

1. **Google Maps Integration:**

* Use Google Maps API for precise location.
* Add features like delivery tracking to let users see the real-time location of the delivery personnel.

1. **User Reviews and Ratings:**

* Allow users to rate and review food items and restaurants.

1. **Loyalty Programs and Promotions:**

* Introduce a loyalty program to reward frequent customers with points redeemable for discounts.

1. **Security Features:**

* Ensure secure user authentication with two-factor authentication (2FA).

 Safeguard payment information using industry-standard encryption.

# References

|  |  |
| --- | --- |
| [1] | K. Acharya, "Online Food Ordering System," ResearchGate, kathmandu, 2022. |
| [2] | Ashutosh Bhargave, Niranjan Jadhav, Apurva Joshi, Prachi Oke, S. R Lahane, "Digital Ordering System for Restaurant Using Android," *International Journal of Scientific and Research,* 2013. |

# APPENDICES

## A screenshot of a food order Description automatically generatedA screenshot of a food login Description automatically generatedAPPENDICES A: Screen Shots

**Figure 1: Login of Foodie Heaven Figure 1: Home of Foodie Heaven**

**A screenshot of a phone

Description automatically generatedA screenshot of a menu

Description automatically generated**

**Figure 3: Cart Figure 3: Order**

**A screenshot of a phone

Description automatically generatedA screenshot of a phone

Description automatically generated**

**Figure 3: History Figure 3: Profile**

**A screenshot of a phone

Description automatically generatedA screenshot of a phone

Description automatically generated**

**Figure 4: Create admin account Figure 4: admin dashboard**

## APPENDICES B: Source Code