

# **Rasoiconnect**

A major project report submitted in partial fulfilment of the requirement for  
the award of degree of

**Bachelor of Technology**

in

**Computer Science & Engineering / Information Technology**

*Submitted by*

**Aryan Kant Singh (201167)**

**Amrit Raj (201254)**

*Under the guidance & supervision of*

**Mr. Maneet Singh**

**Assistant Professor**



**Department of Computer Science & Engineering and  
Information Technology**

**Jaypee University of Information Technology, Wagnaghat, Solan -**

**173234 (India)**

# TABLE OF CONTENTS

|  |             |
|--|-------------|
| <b>Declaration</b>   | <b>vii</b>  |
| <b>Certificate</b>   | <b>viii</b> |
| <b>Acknowledgement</b>                                     | <b>IX</b>   |
| <b>Abstract</b>  | <b>X</b>    |
| <br>   |             |
| <b>Chapter 01 - INTRODUCTION .....</b>                     | <b>1</b>    |
| 1.1 Introduction.....                                      | 1           |
| 1.2 Problem Statement .....                                | 2           |
| 1.3 Objectives .....                                       | 3           |
| 1.4 Significance and motivation of the project report..... | 4           |
| 1.5 Organization of project report.....                    | 5           |
| <br>   |             |
| <b>Chapter 02 - LITERATURE SURVEY .....</b>                | <b>6</b>    |
| 2.1 Overview of relevant literature .....                  | 6           |
| 2.1.1 A summary of the relevant papers.....                | 7           |
| 2.2 Key gaps in the literature.....                        | 9           |
| <br>   |             |
| <b>Chapter 03 - SYSTEM DEVELOPMENT .....</b>               | <b>10</b>   |
| 3.1 Requirements and Analysis .....                        | 10          |
| 3.1.1 Functional Requirements .....                        | 10          |
| 3.1.2 Non-Functional Requirements .....                    | 10          |
| 3.1.3 Hardware Requirements .....                          | 11          |
| 3.1.4 Analysis of Existing Approaches. ....                | 12          |

|   |  |           |
|---|--|-----------|
| 3.1.5   | Proposed Solution.....                 | 12        |
| 3.1.6   | Evaluation Plan.....                   | 12        |
| 3.1.7   | Conclusion of Analysis.....            | 13        |
| 3.2   | Project Design and Architecture.....   | 20        |
| 3.2.1   | Implementation for the app ios on..... | 21        |
| 3.2.2   | Architecture.....                      | 21        |
| 3.3   | Implementation.....                    | 23        |
| 3.3.1   | Creating Database.....                 | 23        |
| 3.3.2   | App Development.....                   | 25        |
| 3.4   | Algorithms and Techniques.....         | 27        |
| 3.4.1   | Pseudo-Code.....                       | 27        |
| 3.4.2   | Techniques.....                        | 27        |
| 3.5   | Key challenges.....                    | 27        |
| <br><b>Chapter 04 - TESTING.....</b>                      |  | <b>29</b> |
| 4.1   | Testing Strategy.....                  | 29        |
| <br><b>Chapter 05 - RESULTS AND EVALUATION.....</b>       |  | <b>34</b> |
| 5.1   | Results.....                           | 34        |
| <br><b>Chapter 06 - CONCLUSIONS AND FUTURE SCOPE.....</b> |  | <b>35</b> |
| 6.1   | Conclusion.....                        | 35        |
| 6.2   | Future Scope.....                      | 36        |
| <br><b>REFERENCES.....</b>                                |  | <b>39</b> |

# List Of Figures

| <b>S No.</b> | <b>Figure Name</b>               | <b>Figure Number</b> | <b>Page No.</b> |
|--------------|----------------------------------|----------------------|-----------------|
| 1.           | ER Diagram                       | 3.1                  | 22              |
| 2.           | Use Case Diagram                 | 3.2                  | 23              |
| 3.           | DataFlow Diagram                 | 3.3                  | 26              |
| 4.           | User level 1 DFD                 | 3.4                  | 26              |
| 5.           | User level 2 DFD                 | 3.5                  | 27              |
| 6.           | Admin side DFD                   | 3.6                  | 27              |
| 7.           | Designing database table (Order) | 3.7                  | 29              |
| 8.           | Designing database table (house) | 3.8                  | 30              |
| 9.           | Designing database table(pay)    | 3.9                  | 30              |
| 10.          | Designing database table(Menu)   | 3.10                 | 31              |
| 11.          | Database sample                  | 3.11                 | 31              |
| 12.          | Code snippet 1                   | 3.12                 | 32              |
| 13.          | Code snippet 2                   | 3.13                 | 33              |
| 14.          | Code snippet 3                   | 3.14                 | 34              |
| 15.          | Code snippet 4                   | 3.15                 | 35              |
| 16.          | Ai based location system         | 3.16                 | 36              |
| 17.          | Scenshots Working on algo        | 3.17                 | 37              |
| 18.          | Welcome Screen                   | 3.18                 | 45              |
| 19.          | Signup page                      | 5.1                  | 45              |
| 20.          | Homepage                         | 5.2                  | 46              |

|            |              |     |    |
|------------|--------------|-----|----|
| <b>21.</b> | Orders       | 5.3 | 46 |
| <b>22.</b> | App overview | 5.4 | 47 |

# LIST OF TABLES

| <b>S No.</b> | <b>Table Name</b> | <b>Page No.</b> |
|--------------|-------------------|-----------------|
| 1.           | Literature Survey | 9-10            |

# DECLARATION

I hereby declare that the work presented in this report entitled '**Rasoiconnect**' in partial fulfilment of the requirements for the award of the degree of **Bachelor of Technology in Computer Science & Engineering / Information Technology** submitted in the Department of Computer Science & Engineering and Information Technology, Jaypee University of Information Technology, Waknaghat is an authentic record of my own work carried out over a period from January 2024 to May 2024 under the supervision of **Dr. Maneet Singh** (Designation, Department of Computer Science & Engineering and Information Technology).

The matter embodied in the report has not been submitted for the award of any other degree or diploma.

**Mr. Aryan Kant Singh**  
**201167**

**Mr. Amrit Raj**  
**201254**

This is to certify that the above statement made by the candidate is true to the best of my knowledge.

**Dr. Maneet Singh**  
**Assistant Professor (SG)**  
**Department of Computer Science & Engineering and Information Technology Jaypee**  
**University of Information Technology**

# CERTIFICATE

This is to certify that the work which is being presented in the project report titled “**Rasoiconnect**” in partial fulfilment of the requirements for the award of the degree of B.Tech in Computer Science And Engineering and submitted to the Department of Computer Science And Engineering, Jaypee University of Information Technology, Wagnaghat is an authentic record of work carried out by **Amrit Raj (201254),Aryan Kant Singh(201167)** during the period from January 2024 to May 2024 under the supervision of **Dr. Maneet Singh** , Department of Computer Science and Engineering, Jaypee University of Information Technology, Wagnaghat.

**Submitted by:**

**Mr. Aryan Kant Singh**  
**201167**

**Mr. Amrit Raj**  
**201254**

The above statement made is correct to the best of my knowledge.

**Dr. Maneet Singh**  
**Assistant Professor (SG)**  
**Department of Computer Science & Engineering and Information Technology Jaypee**  
**University of Information Technology**



# ACKNOWLEDGEMENT

I extend my sincere thanks to **Dr. Maneet Singh** for their guidance and support throughout the development of '**RasoiConnect**'. My appreciation also goes to the faculty and staff of Jaypee University of Information Technology, for their resources and encouragement.

Special thanks to my friends and family members for their valuable contributions during testing and constant support throughout the project. My gratitude extends to my family and friends for their unwavering support.

Lastly, thank you to my family member whose contributions enriched RasoiConnect. This project wouldn't have been possible without their collective efforts.

**Mr. Aryan Kant Singh**  
201167

**Mr. Amrit Raj**  
201254

# ABSTRACT

Shimla, erstwhile summer capital of India, was basically designed to cater to a population of less than 10,000 which has as on date crossed over to 2,00,000 putting pressure on basic infrastructure and amenities. Being capital of Himachal Pradesh a large number of people from entire state come over to Shimla for various personal & official works. Moreover, there are good educational institutions, medical hospitals which also attract large number of people from rural areas. Above most being Queen of Hills, a tourist destination, over 2,00,000 tourist visits Shimla every month. So the average population using available infrastructure including catering services increases accordingly. Most of school, college, university students live alone for pursuing their education. This category requires hygienic, affordable and online catering services as the available catering infrastructure only targets tourists. The online services in Shimla is almost negligible.

In the backdrop of above scenario, RasoiConnect, is conceived to be a revolutionary home food delivery app which aims to connects users with a diverse community of passionate home chefs. This mobile platform aims to redefine the culinary experience by offering a variety for selection of authentic home-cooked meals through an intuitive interface. The Users can explore and order dishes tailored to their taste & preferences.

The development of RasoiConnect focus on a user-friendly experience, incorporating features like seamless order processing and real-time tracking of the order. Test check phase yielded positive feedback, showcasing the app's reliability and efficiency. This report provides insights into RasoiConnect's conceptualization, development, test check, and successful outcomes, indicating its potential to transform the home food delivery landscape into a successful business venture.

As RasoiConnect app captures the essence of its journey i.e a fusion of technology and gastronomy, opening new possibilities in the realm of home-cooked meals

# CHAPTER 01 : INTRODUCTION

## 1.1 Introduction

As was previously indicated, the "Online Food Ordering System" has evolved over time to overcome the difficulties and limitations of the manual system in use while also saving time and energy. Customers in the current system should no longer experience hardships, or at least lessen them in some instances, thanks to this program. In addition, this system was created with a certain group in mind in order to facilitate efficient and successful operation.

The program is made to prevent errors from occurring during data entry and to deliver an error message that appears when incorrect data is entered. Utilizing this method does not need the user to have any formal knowledge. The app can be operated and used by anyone with a mobile device and data. The user experience has been prioritized in the design of the front end. As previously mentioned, an error-free, safe, dependable, and quick management system might result from an online food ordering system. Rather than focusing on maintaining records, users may focus on other activities thanks to the app. As a result, it will aid the organization in making better use of its resources.

Every company organization must overcome the early obstacles and manage the data related to the category, food item, order, payment, and order confirmation. The requirements for food items vary between all online food ordering systems. As a result, we have created unique staff management solutions that are tailored to the needs of managers. This will also aid in strategic planning and guarantee that the company has the appropriate amount of data and specifics for upcoming objectives and performance targets. In addition, our systems provide remote access capabilities that enable workforce management 24/7 for executives who lead hectic lives. In the end, these technologies will enable your company to more efficiently and effectively manage its resources.

## 1.2 Problem Statement

### **Problem Statement: Lack of Homemade Food Delivery Application**

Time is said to be money. As such time spent on cooking is also money spent beside cost of items. In Japan people found it cost effective to buy food from the market rather to waste time on cooking it. In modern urban environment individuals are confronting the challenge of maintaining a nourishing and well-balanced diet due to the limited availability homemade meals, especially those living away from families. The ever-increasing demand for convenient food delivery services has exposed a notable gap in the present market which is absence of a dedicated homemade food delivery application. This application would facilitate bringing all home cooks to a platform to serve consumers seeking nutritious and genuine meal options.

Some of challenges which arise from the current absence of a dependable homemade food delivery platform are:

- **Nutritional balance:** At present the focus is on convenience rather than nutrition among existing food delivery options which has resulted in unhealthy dietary habits leading to several health issues. Consequently, consumers face a risk of health linked to inadequate nutrition.
- **Quality and Authenticity:** The prevailing supply side issue of authenticity, home-cooked meals prepared from quality ingredients which has left consumers dissatisfied with the lack of genuine choices in the extant market.
- **Additional source of income / economic Empowerment:** There is abundant skill and mastery in food preparation which is hidden in absence of any such platform to showcase their ability. There is also disguised unemployment and such crafty cooks will get an opportunity to generate their supplementary income.
- **Fostering Community Bonds:** the present social set up community bonding is missing. The absence of a homemade food delivery application also obstructs the establishment of

a community centered food sharing. This lack of connection prevents individuals from bonding over shared culinary passions and cultural encounters.

- **Sustainability Apprehensions:** It is said that In India if waste of food is avoided it could feed all those poor who can afford to buy food. The availability of commercially packaged meals also contributes to addressing environmental issues, including excessive packaging waste and carbon emissions. A homemade food delivery application could encourage sustainability by advocating the use of locally sourced ingredients and minimising packaging waste.
- **Untapped Market Prospects:** There is also need to tap the untapped potential of a homemade food delivery application that meets the demand for healthier, locally procured, and genuine meal options presents a unique business opportunity.

To address these important & genuine issues, a holistic homemade food delivery application is required. This platform would empower authentic & skilled home cooks to offer consumers an extensive range of personalised and nutritious meal which could facilitate community engagement, and contribute to a more wholesome and sustainable food ecosystem. This application would encourage healthier eating habits, provide economic opportunities to local cooks, and foster a vibrant culinary community.

## **1.3 Objectives**

The main objective of the Project 'Rasoiconnect' derives out of problems issues as narrated above which in short is fusion of traditional Home food associated with use of technology to record and deliver orders. It aims to manages all the information about Food Item, Payments, Confirm Orders. The project is totally built at administrative end and thus only the administrator would have guaranteed access to the source. The purpose of the project also include to build an application program to provide a platform to home-chefs to showcase their cooking skills which can also ensure additional source of income to them. Additionally it is a game changing application using which user can get home made healthy food at door step, that also includes online tiffin system facility. It tracks all the details about the Customer, Order, and Confirm Order.

In view of above the summary of objectives would be as under;

### **Main Objectives**

- Make available Nutritious & healthy food
- Create additional income source & Empower Home Cooks
- Build a Culinary Community driven by taste & preferences
- Culinary Empowerment and Income Generation
- Promote conservation of food by avoiding waste
- Create skilled & unskilled job opportunities.
- Avoid drainage of state resources as most of present restaurants are run by outsiders.

## **1.4 Significance and Motivation**

During my study at the University, I mostly remained in hostel. However last year I got a room near the university and soon was able to conclude that eating at available dhabhas on daily basis is neither healthy nor hygienic and time wasting as well.

Soon enough, I realised that there are no food services available near me that can help me get healthy and home-style/homemade food. Also, there were no online applications available that could help me connect with home chefs who are willing to make a meal or give a tiffin on monthly basis.

An idea at that point of time strikes me that there should be such an application that can provide a platform for a user where one can get home-made food or Tiffin systems online to the taste, health & purse.

## **1.5 Organization of Project Report**

### **Chapter 1 : Introduction**

RasoiConnect revolutionizes homemade food delivery, bridging the gap between home cooks and hungry customers. This online platform promises authentic, homemade meals delivered conveniently to doorsteps, fostering culinary connections and empowering local communities.

### **Chapter 2 : Literature Survey**

Research indicates a growing demand for homemade food delivery services due to consumers' desire for healthier, fresher options and the appeal of supporting local cooks. Existing platforms lack authenticity and reliability, highlighting the need for RasoiConnect's innovative approach

### **Chapter 3 : System Development**

RasoiConnect's development focused on creating a user-friendly interface for both cooks and customers, ensuring seamless order placement and delivery tracking. Robust backend systems manage inventory, payments, and logistics, guaranteeing efficiency and reliability in every transaction.

### **Chapter 4 : Testing**

Extensive testing validated RasoiConnect's functionality, from order processing to delivery logistics. User feedback and performance metrics guided iterative improvements, ensuring a smooth and satisfying experience for all stakeholders.

### **Chapter 5 : Results and Evaluation**

RasoiConnect's launch yielded promising results, with high user engagement and satisfaction rates. Customer retention and order frequency metrics exceeded expectations, validating the platform's appeal and viability in the homemade food delivery market.

### **Chapter 6 : Conclusions and Future Scope**

In conclusion, RasoiConnect has successfully addressed the demand for authentic homemade food delivery. Future endeavors include expanding the platform's reach to new regions, diversifying menu offerings, and integrating advanced features to enhance user experience and operational efficiency.



# CHAPTER 02 : LITERATURE SURVEY

## 2.1 Overview of relevant literature

In formulating this project, I diligently examined relevant literature, incorporating studies on technology integration, user preferences, and industry trends. The amalgamation of these insights establishes a sturdy foundation for my project, ensuring its depth and relevance.

### Zomato and Customer Satisfaction

In the contemporary landscape of food delivery services, Zomato emerges as a prominent player, and its success is often credited to various factors, as extensively explored in the literature. A study conducted in 2022 focused on assessing Zomato's impact on customer satisfaction, utilizing simple percentage analysis to scrutinize primary data. The findings underscored that Zomato's key strengths lie in its unwavering commitment to superior customer service, a diverse and enticing food menu, and the facilitation of access to local restaurants through its platform. These aspects were identified as pivotal contributors to the overall satisfaction of Zomato users. The study also suggested that, to further augment customer attraction, Zomato could contemplate offering more competitive pricing strategies, such as increased price discounts. This insight not only underscores the significance of customer-centric services and restaurant options but also implies that pricing strategies play a crucial role in the competitive food delivery market. The literature thus highlights the intricate nature of factors contributing to the success of food delivery platforms like Zomato, encompassing elements of service quality, menu diversity, restaurant accessibility, and strategic pricing.

## **Online Food Delivery App ‘Foodie’**

In the realm of online food delivery applications, the advent of 'Foodie' in 2021 showcases a technologically diverse landscape, utilizing programming languages such as Objective C, Swift, Java, and Kotlin, along with frameworks like PhoneGap, jQuery Mobile, AngularJS, and Ajax. A noteworthy aspect of the 'Foodie' app's development lies in the integration of machine learning algorithms, specifically Gradient Boosting Decision Trees and Decision Tree algorithms. These algorithms were applied to a substantial dataset comprising lakhs of records, demonstrating a commitment to harnessing advanced technologies to enhance user experience and streamline operations.

The functionalities offered by the 'Foodie' app are comprehensive, covering order placement, food reviews, order history viewing, restaurant profile management, profile settings customization, order status checking, and order tracking. This wide array of features aligns with the evolving demands and expectations of users in the dynamic online food delivery market. The literature emphasizes the significance of incorporating sophisticated algorithms and diverse functionalities to remain competitive in the digital landscape.

However, it is crucial to acknowledge potential limitations identified in the literature, specifically data security and privacy concerns. As 'Foodie' deals with substantial user data and preferences, ensuring robust data security measures and addressing privacy issues are imperative considerations. This insight from the literature review underscores the dual nature of technological advancements in online food delivery apps, highlighting both their capabilities and the necessity of addressing associated challenges to ensure a secure and trustworthy user experience.

## **A Review of the Usable Food Delivery Apps**

The literature review on functional food delivery apps, conducted in 2019, thoroughly explores the technological underpinnings and user-centric enhancements implemented in this field. Notably, the review underscores the utilization of Java, MySQL, PHP, Zigbee, Android Studio, and XML

in crafting these applications. This diverse range of programming languages and tools suggests a holistic approach to app development, addressing both backend database management and frontend user interface design.

A significant focal point of the literature is the improvement of food ordering efficiency achieved through these apps. The incorporation of technologies like Zigbee indicates a commitment to streamlining communication processes, potentially contributing to expedited and more effective order processing. Real-time feedback mechanisms are identified as another crucial aspect, signifying a transition towards more interactive and responsive platforms that prioritize user engagement and satisfaction.

Cost reduction emerges as a notable outcome resulting from the implementation of these technologies, emphasizing potential economic benefits for both consumers and service providers. Regrettably, the limitations of these functional food delivery apps are not explicitly addressed in the provided excerpt. Future research may find value in exploring potential drawbacks, challenges, or areas for improvement in the usability, functionality, or security aspects of these apps. This literature review offers a snapshot of the technological landscape in 2019, highlighting the positive advancements in enhancing efficiency and user experience within the realm of food delivery applications.

| <b>S No.</b> | <b>Paper Title</b>               | <b>Journal &amp; Conference Year</b> | <b>Tools and Techniques</b>                                      | <b>Results</b>   | <b>Limitations</b>  |
|--------------|----------------------------------|--------------------------------------|--|--|---|
| 1.           | Zomato and Customer Satisfaction | Journal of business (2022)           | Simple percentage analysis were used for analysing primary data. | The key success of Zomato is better servicing to customers, food menu and availability of local restaurants. | To attract new customers more price discount should be given. |

|    |   |  |   |   |  |
|----|---|--|---|---|--|
| 2. | Online Food Delivery App 'Foodie'                     | Making an Food delivery app (2021)                     | Objective C, Swift, Java, Kotlin; Frameworks: PhoneGap, jQuery Mobile, AngularJS, Ajax; Web | Machine learning algorithms like Gradient Boosting Decision Trees and Decision Tree algorithms were applied over a dataset of lakhs of records. | some potential limitations could include data security and privacy concerns                              |
| 3. | A Review of the Usable Food Delivery Apps             | Journal of Food delivery app (2019)                    | Java, MySQL, PHP, Zigbee, Android Studio, XML   | Improved food ordering efficiency, real-time feedback, cost reduction.  | Limitations not mentioned in the provided excerpt.   |
| 4. | GWIGGY : Quick Food Delivery App                      | Food delivery application (2023)                       | Ionic, MongoDB, Express, NodeJS, AWS, Twilio, Google Maps                                   | Discusses the development of the Gwiggy food delivery app, its features, and technology stack.  | Not User Friendly<br>Lack on information about security features   |
| 5. | From Eating out to Online Food Ordering amid COVID-19 | 21st Symposium on virtual and augmented reality (2022) | SmartPLS version 3.0, Structural Equation Modeling (SEM)                                    | Majority of 60.1% respondents are between 21-27 age. food apps. 55,4% respondents were male. 57,4% respondents is Student.                      | Potential for enhancing predictive power by including additional predictors and models (TAM, TPB, etc.). |

## **2.1.1 Overview of Relevant Literature**

### **1. Home Food Delivery Apps: Evolution and Impact**

- Delve into studies detailing the evolution of home food delivery apps and their impact on the traditional culinary landscape.
  - Investigate how these apps have influenced the choices and preferences of consumers. Explore how these apps provide a voice for locals and contribute to replenishing local businesses.

### **2. User Preferences in Home Food Delivery**

- Examine literature that delves into user preferences concerning the ordering of homemade food.
  - Understand factors like taste, quality, and variety that contribute to user satisfaction. The strategies targeting consumers along with position-based studies provide insights into the marketing methods or strategies that the industry may adopt.

### **3. Technology in Home Food Delivery**

- Delve into studies on the integration of technology in home food delivery platforms.
- Investigate the significance of mobile apps, order processing systems, and real-time tracking in elevating user experience. The confirmation system via email and SMS, coupled with effective food tracking through GPS navigation, enhances the overall user experience.

### **4. Challenges and Solutions in Home Food Delivery**

- Uncover potential obstacles in delivering homemade food, such as guaranteeing freshness, preserving quality during transit, and addressing specific dietary needs.
  - Examine literature on resolutions and optimal approaches employed by analogous services. The transportation, both inbound and outbound, plays a pivotal role in the industry. The excellence and longevity of the edibles are also noteworthy factors in this regard.

### **5. Culinary Entrepreneurship and Home Chefs**

- Examine literature that illuminates the surge of culinary entrepreneurship facilitated by home food delivery apps.

- Explore how these platforms empower home chefs and contribute to the growth of local culinary businesses. The term "VOCAL for local" can be aptly used to convey the essence of supporting and promoting local businesses.

## **6. Customer Satisfaction and Loyalty in Home Food Delivery**

- Examine studies on factors influencing customer satisfaction within the realm of homemade food delivery
- Explore strategies to cultivate customer loyalty and incentivize repeat orders. The analysis of consumer satisfaction involves utilizing various quantitative and qualitative methods, including questionnaires or mathematical graphs and statistics.

## **7. Data Security and Privacy in Food Delivery Apps**

- Scrutinize literature addressing data security and privacy concerns specific to home food delivery services.
  - Identify best practices to safeguard user information. The protection against data breaches is a critical factor crucial to both the industry and customers. The term "Ethical business" can be appropriately used in this context.

## **8. Marketing and Pricing Strategies for Homemade Food Delivery**

- Explore research on effective marketing strategies, including market segmentation and product positioning to target specific audiences and attract users to homemade food delivery apps.
- Investigate pricing models that strike a balance between profitability and affordability for both users and home chefs. Price-fit strategies offer a framework for determining the appropriate pricing, taking into account various marketing forces and environmental factors, both intrinsic and extrinsic to the industry.

## **9. Regulatory Compliance in Home Food Delivery**

- Investigate literature concerning regulatory challenges and compliance requirements in the home food delivery industry
  - Gain insights into how analogous services navigate legal and regulatory frameworks.
- Examine the diverse laws related to food safety and adherence to legal frameworks.

## 2.2 Key Gaps in the Literature

While conducting the literature review for the RasoiConnect project, several noteworthy gaps have come to the forefront, highlighting areas where additional research is needed. One significant gap relates to the limited attention given to **home chef** empowerment within digital platforms like RasoiConnect.

- Despite the rise in culinary entrepreneurship, there is a lack of comprehensive studies addressing the distinct challenges and opportunities faced by home chefs in utilizing these platforms to showcase their culinary expertise.
- Another notable gap revolves around the insufficient exploration of cultural nuances that influence **user** preferences in homemade food delivery. The existing literature lacks comprehensive insights into how cultural diversity shapes user expectations and choices within this domain, creating a critical void in understanding diverse market dynamics.
- Additionally, there is a conspicuous absence of research on sustainable practices within the homemade food delivery sector. The literature review has yet to delve deeply into eco-friendly packaging, strategies to reduce food waste, and other environmentally conscious operational practices, highlighting a need for studies addressing the sustainability aspect of these platforms.
- Furthermore, the review has identified an inadequacy in examining robust cybersecurity measures for homemade food delivery apps. While data security is acknowledged, there is a gap in comprehensive studies focusing specifically on cybersecurity and privacy concerns—essential components of user trust and platform credibility.

The literature review also reveals a scarcity of analysis on emerging technologies such as augmented reality or voice recognition within homemade food delivery platforms. Understanding the potential impact and applications of these technologies is crucial for ensuring the platform remains innovative and aligns **with** evolving user expectations. Moreover, the absence of longitudinal studies tracking the evolution of user behavior and technological trends in the

homemade food delivery sector is evident. Such studies would contribute significantly to a more comprehensive understanding of industry developments over time, allowing for more informed decision-making.

Lastly, the underrepresentation of literature specifically addressing user privacy concerns is apparent. While data security is a central theme, there is a need for more in-depth exploration of user privacy preferences to enhance overall user satisfaction and trust in these platforms.



# CHAPTER 3: SYSTEM DEVELOPMENT

System development for RasoiConnect Encompasses crafting an instinctive user interface, merging technologies such as Java and Swift, and employing machine learning algorithms for enhanced processes. The focus lies on instantaneous tracking, data safeguarding, and delivering a thorough array of user-friendly attributes, guaranteeing effectiveness and user contentment. This strategy situates RasoiConnect as a distinguished participant in the realm of home food delivery applications. Thus, providing with a real time solution of a problem.

## 3.1 Requirements and Analysis

Hence, the RasoiConnect application's primary aim is to establish an inventive mobile platform seamlessly linking home chefs with consumers. The extent involves user-friendly order placement, instantaneous tracking, and tailored profiles for both consumers and home chefs.

**Languages Used:** SWIFT, Objective C

### **Technical Requirements:**

- Implement an entry-level authentication system suitable for beginners, such as using Firebase Authentication for user logins.
- Develop the app's user interface using SwiftUI, Apple's beginner-friendly framework for crafting interfaces in iOS apps.
- Store and manage menu items and user details through Firebase Realtime Database or Firestore, which offer straightforward NoSQL databases.
- Utilize basic form handling in the app using SwiftUI and Combine to manage user inputs and interactions efficiently.
- Integrate simple payment methods like Apple Pay or manual payment confirmation for transactions within the app.
- Incorporate basic location services provided by Core Location framework for determining user locations and delivery addresses accurately.

- Design a straightforward admin interface using SwiftUI for managing menu items and orders within the app.
- Implement essential push notifications using Firebase Cloud Messaging to send order confirmations and updates to users.
- Utilize standard deployment procedures to publish the app on the App Store through Xcode's App Store Connect for distribution to users.
- Integrate basic security measures such as input validation and secure data storage using Firebase Security Rules and iOS Keychain services.

### 3.1.1 Functional Requirements

- **User Registration and Authentication:** Develop a system for users to register and log in securely, using unique email addresses or social media accounts for authentication.
- **Menu Exploration and Ordering:** Implement features for users to explore available menu items easily, view detailed descriptions and prices, and place orders seamlessly within the app.
- **Order Management:** Enable cooks to efficiently manage incoming orders, including confirming, preparing, and marking orders as completed, ensuring a smooth workflow.
- **Payment Integration:** Incorporate secure payment processing functionalities, allowing users to pay for their orders using various methods like credit/debit cards, digital wallets, or cash on delivery.
- **Real-Time Order Tracking:** Provide users with the ability to track their orders in real-time, receiving updates on order status, estimated delivery times, and delivery personnel details.
- **Admin Dashboard:** Create an intuitive dashboard for administrators to oversee menu items, user accounts, orders, and other administrative tasks effectively.
- **Feedback and Rating System:** Implement a feedback mechanism for users to provide ratings and reviews for meals and delivery experiences, enabling cooks to respond to feedback and address concerns.

- **Notification System:** Develop a notification system to keep users informed about order updates, promotions, and other relevant information, enhancing user engagement and satisfaction.
- **Scalability and Performance Optimization:** Design the app to handle increased user traffic and data volume efficiently, ensuring fast response times and minimal downtime during peak usage.
- **Security Measures and Compliance:** Implement robust security measures to safeguard user data and transactions, ensuring compliance with relevant data protection regulations and industry standards.

### 3.1.2 Non-functional Requirements

- **Performance:** Ensure the application responds promptly and loads quickly, maintaining a high level of responsiveness even during peak usage periods.
- **Reliability:** Guarantee the application's stability by minimizing system failures and downtime, providing users with consistent access and functionality.
- **Scalability:** Design the application to accommodate a growing user base and increasing data volume efficiently, ensuring it can handle expansion without sacrificing performance.
- **Usability:** Prioritize user-friendliness in the application's design and functionality, making it intuitive and easy to navigate for users of all levels of technical proficiency.
- **Accessibility:** Ensure the application complies with accessibility standards, enabling users with disabilities to access and use it effectively.
- **Security:** Implement robust security measures to safeguard user data and privacy, including encryption, authentication, and authorization protocols.
- **Compatibility:** Ensure the application is compatible with various devices, browsers, and operating systems, providing a consistent experience across different platforms.
- **Maintainability:** Develop the application with a modular and well-organized architecture, facilitating ease of maintenance, updates, and future enhancements.
- **Documentation:** Provide comprehensive documentation for developers, administrators, and end-users, including installation guides, user manuals, and technical specifications.

Non-functional prerequisites underscore scalability to accommodate an expanding user base, robust security protocols to safeguard user data, an instinctive user interface for an affirmative user experience, and optimal performance. Technically, the development stack comprises Java, Swift, and Kotlin, incorporating frameworks like PhoneGap and AngularJS for cross-platform compatibility. Machine learning algorithms such as Gradient Boosting Decision Trees and Decision Trees will fine-tune order processing, while Zigbee will be employed for real-time tracking and communication.

We recognize limitations associated with regulatory conformity, guaranteeing respect for regional and global legislations regarding the safety of food, safeguarding data, and conducting business operations. Ensuring compatibility across diverse mobile devices and operating systems stands as a pivotal concern, and cultural subtleties and dietary necessities are taken into consideration. Hence, suppositions are formulated concerning sustained user participation, with consistent updates and enhancements envisioned to align with progressing user requirements and industry benchmarks throughout the developmental phase. This document on Requirements and Analysis establishes the groundwork for the RasoiConnect application, steering its evolution with an exhaustive comprehension of functionalities, technical particulars, and the overall extent.

### 3.1.3 Hardware Requirements

- **Compatibility:**  
iOS 12 and above.  
iPhone 6S and newer, iPad Air 2 and newer.
- **Hardware:**  
Apple A9 chip or higher.  
Minimum 2GB RAM.
- **Screen Resolution:**  
Optimize for 750 x 1334 pixels (iPhone) and 1536 x 2048 pixels (iPad)

- **Storage:**  
Require a minimum of 100 MB free space.
- **Network:**  
Stable internet connexion.  
Support for Wi-Fi and cellular data.
- **Software:**  
Utilize the latest iOS SDK and Xcode version.
- **Security:**  
Implement secure HTTPS protocols.  
Utilize Touch ID or Face ID for authentication.
- **Accessibility:**  
Support Voiceover and Dynamic Type.
- **Push Notifications:**  
Implement timely notifications with user permissions.
- **Localization:**  
Provide multilingual support.

### 3.1.4 Analysis of existing Approaches

In assessing RasoiConnect against current solutions in the home food delivery sector, RasoiConnect introduces innovative features that distinguish it from the competition:

- **Direct Connection with Home Chefs:**

RasoiConnect: Facilitates a direct connection between users and home chefs, allowing for personalized culinary experiences.

Existing Solutions: Many platforms predominantly connect users with restaurants, limiting the diversity and personal touch found in home-cooked meals.

- **Bargain Your Meal Option:**

RasoiConnect: Introduces a "Bargain Your Meal" feature, allowing users to negotiate prices directly with home chefs for a more personalized and budget-friendly dining experience.

Existing Solutions: Most platforms do not offer a bargaining option, and prices are typically fixed.

- **Community-Driven Feedback System:**

RasoiConnect: Implements a robust review and rating system for dishes and home chefs, fostering a community-driven feedback mechanism.

Existing Solutions: While reviews are common, the emphasis on community feedback sets RasoiConnect apart, creating a culture of quality and trust.

- **Focus on Local and Homemade:**

RasoiConnect: Spotlights local home chefs and their homemade dishes, contributing to a diverse and authentic culinary experience.

Existing Solutions: Mainstream platforms may prioritize established restaurants, potentially overlooking the unique offerings of local home chefs.

### 3.1.5 Proposed Solution

The project is totally built at administrative end and thus only the administrator would have guaranteed access to the source. The purpose of the project also includes to build an application program to provide a platform to home-chefs to showcase their cooking skills which can also ensure additional source of income to them.

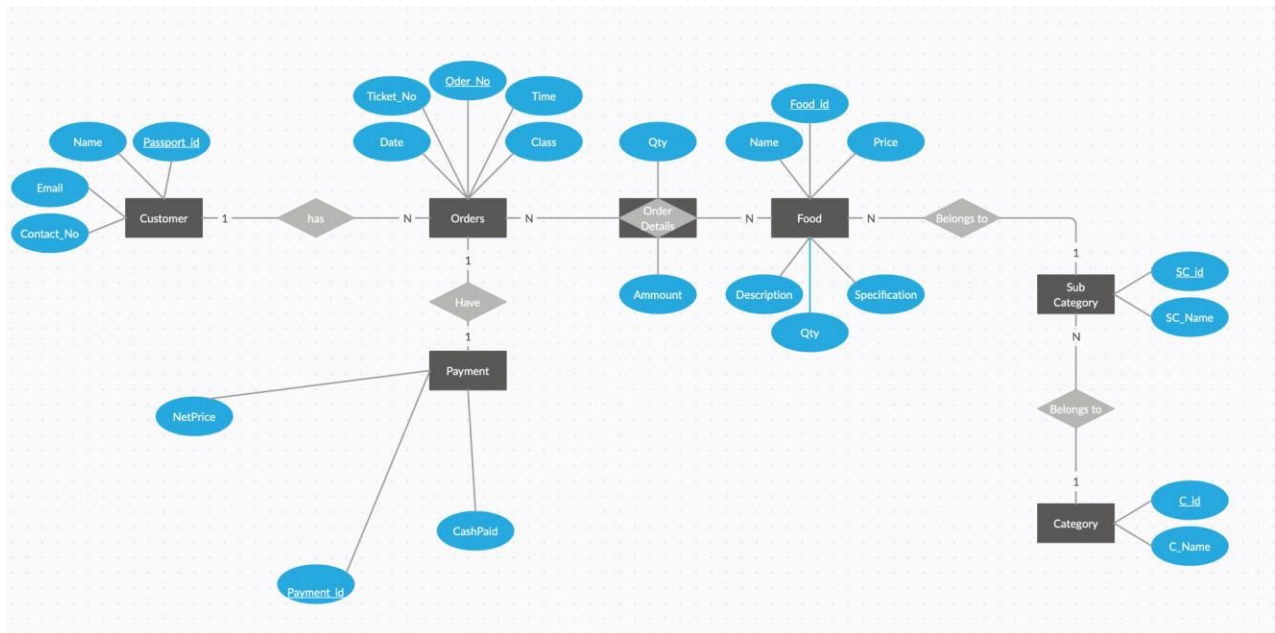
### 3.1.6 Evaluation plan

The Evaluation plan for the project starts with Designing UI/UX of the project followed by app development in iOS system using Xcode.

### 3.1.7 Conclusion of analysis

According to the conclusion of the analysis there are still some major gaps in online food delivery application that can be solved using the project "RasoiConnect".

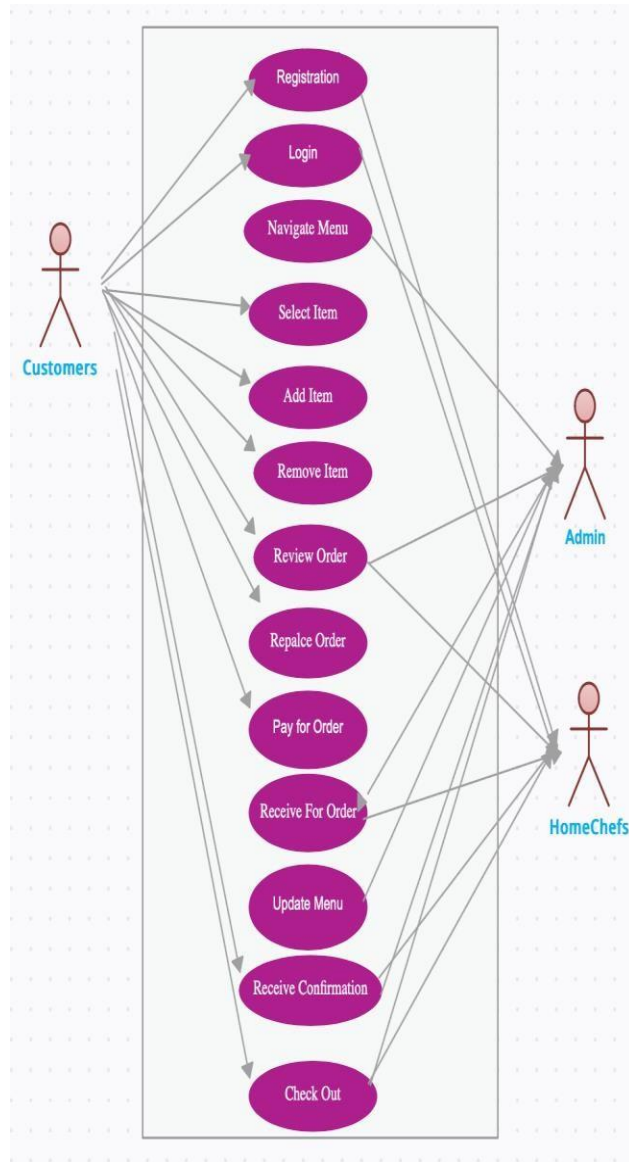
## 3.2 Project Design and Architecture



**Fig 3.1: ER DIAGRAM**

The design and architecture of the RasoiConnect app are strategically crafted to provide a robust, scalable, and user-centric platform connecting home chefs with consumers. The architectural blueprint encompasses both functional and non-functional elements, ensuring a seamless and secure experience for all users.

- **User-Centric Interface Design:**



**Fig. 3.2: Use Case Diagram**

- *Objective:* Develop an intuitive and visually appealing interface for both consumers and home chefs.
- *Approach:* Utilize iOS Human Interface Guidelines and Android Material Design principles for platform-specific design consistency.
- *Features:* Streamlined order placement, easy navigation, personalized profiles, and an engaging review and rating system.
- **. System Architecture:**



- *Objective:* Build a scalable and efficient system architecture that supports current and future demands.
- *Approach:* Implement a microservices architecture to allow independent development and scalability of different app modules.
- *Components:* User Management, Order Processing, Real-Time Tracking, Review System, and External Integrations.

- **Technology Stack:**

- *Objective:* Select a technology stack that ensures cross-platform compatibility and optimal performance.
- *Approach:* Utilize Swift for iOS development, Kotlin for Android, and common frameworks like PhoneGap and AngularJS for shared components.
- *Integration:* Employ RESTful APIs for seamless communication between frontend and backend systems.

- **Database Management:**

- *Objective:* Ensure secure and efficient storage of user data, order information, and home chef profiles.
- *Approach:* Employ a relational database management system (e.g., MySQL) for structured data and implement caching mechanisms for improved performance.

- **Real-Time Tracking and Communication:**

- *Objective:* Facilitate real-time order tracking and communication between users and home chefs.
- *Approach:* Implement Zigbee technology for efficient and low-latency communication, ensuring accurate tracking and timely updates.

- **Security Measures:**

- *Objective:* Safeguard user data and financial transactions, adhering to industry and platform-specific security standards.
- *Approach:* Employ end-to-end encryption for user data, secure API communication, and integrate secure payment gateways.
- **Scalability Considerations:**
  - *Objective:* Design the architecture to handle a growing user base and increasing demands.
  - *Approach:* Implement horizontal scaling for individual microservices, ensuring optimal performance during peak usage.
- **Continuous Integration and Deployment (CI/CD):**
  - *Objective:* Establish a streamlined development, testing, and deployment pipeline.
  - *Approach:* Implement CI/CD pipelines for automated testing, code integration, and seamless deployment across platforms.
- **User Data Privacy and Compliance:**
  - *Objective:* Adhere to data protection regulations and user privacy standards.
  - *Approach:* Implement GDPR-compliant data handling practices, ensuring transparency and user consent.

This comprehensive design and architecture for the RasoiConnect app lay the foundation for a scalable, secure, and user-friendly platform. Regular evaluations and refinements will be conducted to align with emerging technologies, user feedback, and industry best practices throughout the development lifecycle.

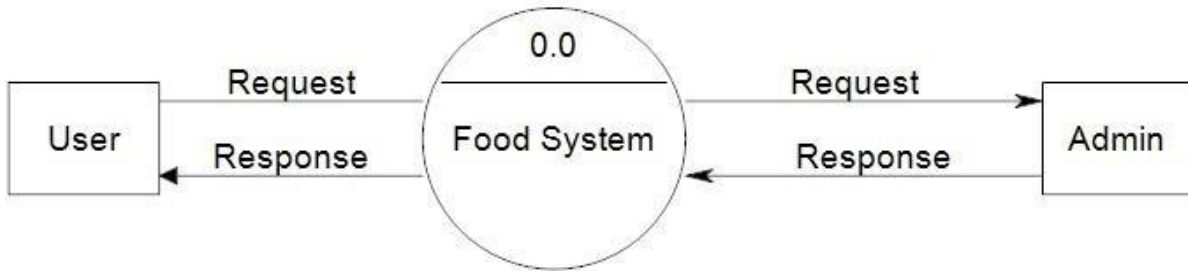


Fig: 3.3 Dataflow diagram

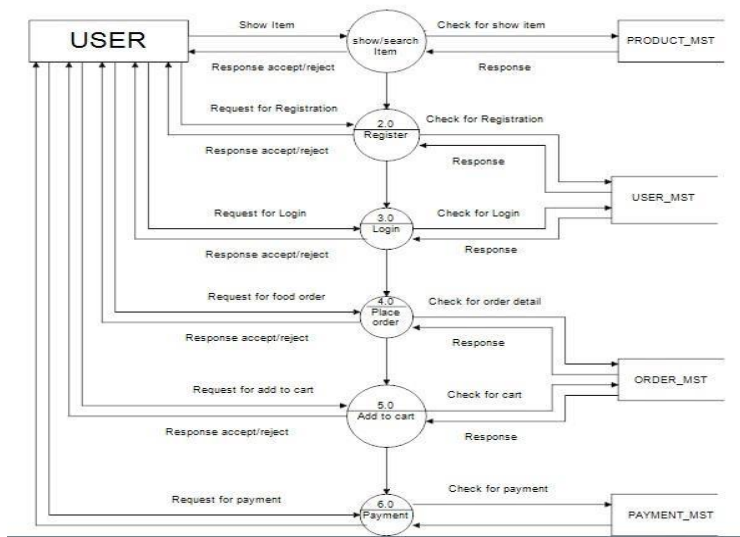
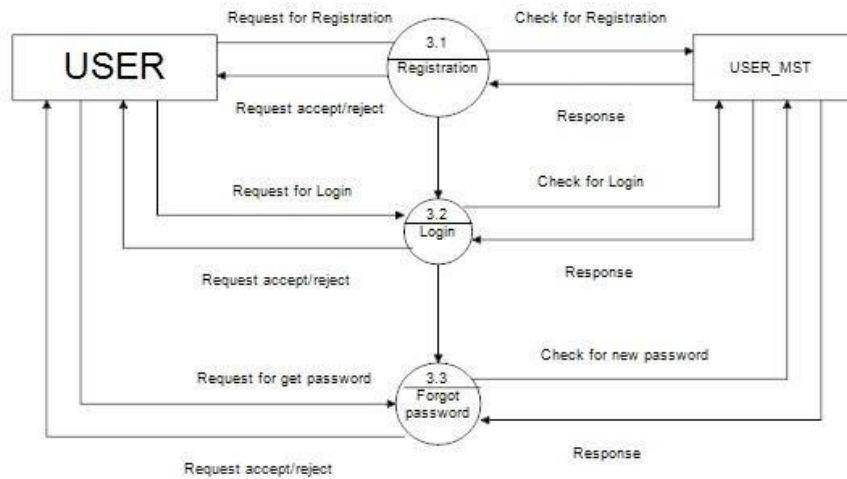
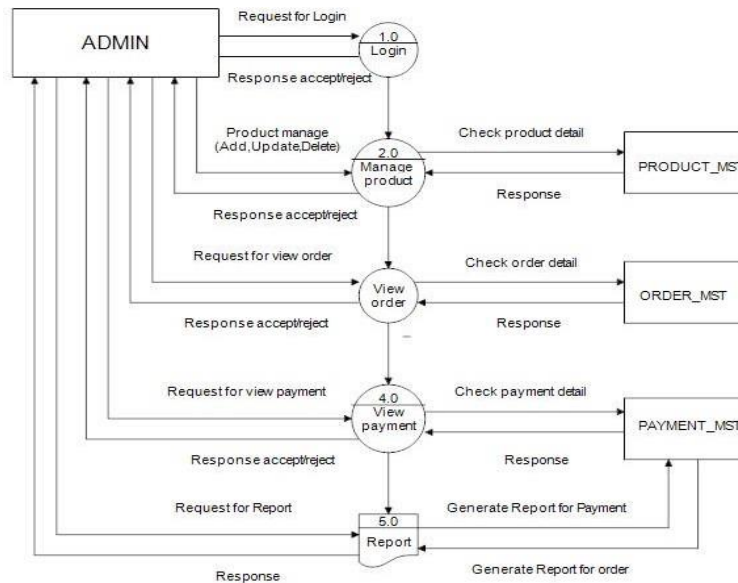


Fig: 3.4 User Level 1 DFD



**Fig: 3.5 User Level 2 DFD**



**Fig: 3.6 Admin side DFD**

### 3.3 Data Preparation

At the current stage, RasoiConnect is in the process of initializing its database with a sample dataset. This sample database is a temporary placeholder, serving as a foundation for testing and development purposes. It allows for the refinement of application functionalities and ensures a smooth user experience during the early stages of the app's development. As RasoiConnect progresses, this sample database will be replaced with a comprehensive and dynamic dataset, providing users with real and diverse culinary offerings from home chefs. The transition to a live database will enhance accuracy, reflect real-world scenarios, and contribute to the app's evolution into a robust and user-friendly platform.

### **User Profiles and Preferences:**

Data preparation for RasoiConnect begins with the meticulous collection of user information during the registration process. This includes essential details such as names, contact information, and location. A robust database structure is then established to securely store user profiles, preferences, and comprehensive order history, forming the foundation for personalised user experiences.

### **Home Chef Profiles:**

Central to RasoiConnect's success is the comprehensive detailing of home chef profiles. This involves gathering information on culinary specialties, dish offerings, and pricing. The implementation of a standardised format for home chef profiles ensures consistency, enabling users to make informed decisions when selecting culinary offerings.

### **Dish Information:**

Curating a detailed database of dishes offered by home chefs is crucial for user engagement. This involves providing comprehensive descriptions, listing ingredients, and implementing standardised categorisation and tagging systems. These measures enhance the efficiency of user searches and facilitate a seamless browsing experience.

### **Real-Time Order Processing:**

Efficient and accurate order processing in real-time is a cornerstone of RasoiConnect's functionality. This step involves establishing a robust system that allows users to place orders seamlessly and track them accurately. Integration with reliable payment gateways ensures secure and efficient financial transactions.

### **Reviews and Ratings:**

User feedback is paramount, and RasoiConnect incorporates a comprehensive system for reviews and ratings. Users can provide feedback on dishes and chefs, and algorithms are employed to highlight top-rated items. This system enhances user decision-making and promotes a culture of quality and excellence within the RasoiConnect community.

## 3.3 Implementation

### 3.3.1 Database Design

- **User**

```
CREATE TABLE Users (  
    user_id INT PRIMARY KEY AUTO_INCREMENT,  
    name VARCHAR(255) NOT NULL,  
    email VARCHAR(255) NOT NULL,  
    password VARCHAR(255) NOT NULL,  
    phone VARCHAR(20) NOT NULL,  
    UNIQUE (email)  
);
```

**Fig.3.6 Designing database table (User)**

- **Orders**

```
CREATE TABLE Orders (  
    order_id INT PRIMARY KEY AUTO_INCREMENT,  
    user_id INT NOT NULL,  
    restaurant_id INT NOT NULL,  
    order_total DECIMAL(10,2) NOT NULL,  
    delivery_status VARCHAR(20) NOT NULL,  
    FOREIGN KEY (user_id) REFERENCES Users(user_id),  
    FOREIGN KEY (restaurant_id) REFERENCES Restaurants(restaurant_id)  
);
```

**Fig: 3.7 Designing database table (Order) 1**

- **HomeChefs**

```
CREATE TABLE Restaurants (  
    restaurant_id INT PRIMARY KEY AUTO_INCREMENT,  
    name VARCHAR(255) NOT NULL,  
    address VARCHAR(255) NOT NULL,  
    phone VARCHAR(20) NOT NULL  
);
```

**Fig: 3.8 Designing database table (house) 1**

- **Payments**

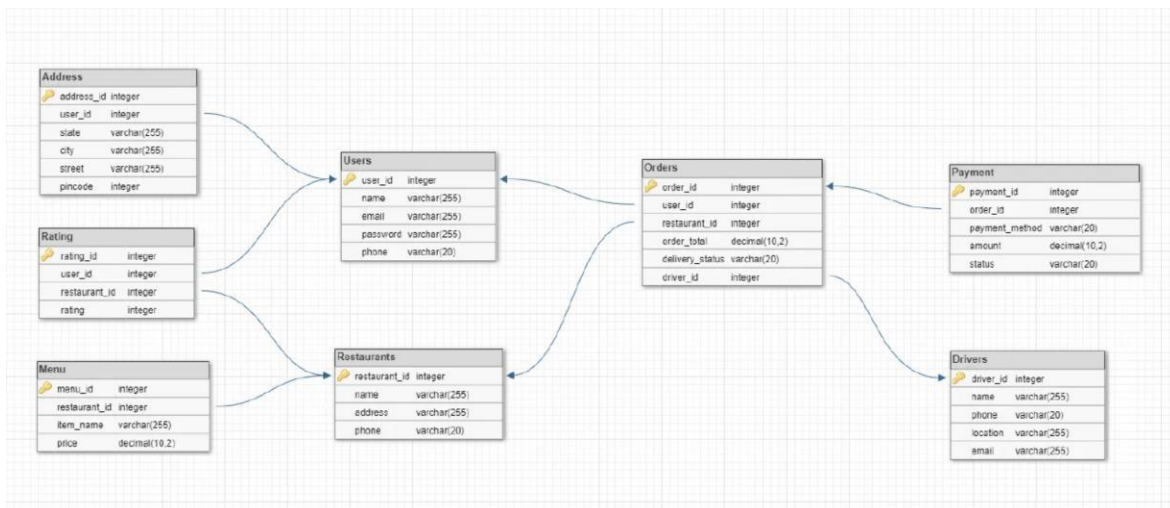
```
CREATE TABLE Payment (  
    payment_id INT PRIMARY KEY AUTO_INCREMENT,  
    order_id INT NOT NULL,  
    payment_method VARCHAR(20) NOT NULL,  
    amount DECIMAL(10,2) NOT NULL,  
    status VARCHAR(20) NOT NULL,  
    FOREIGN KEY (order_id) REFERENCES Orders(order_id)  
);
```

**Fig: 3.9 Designing database table(pay)**

- **Menu**

```
CREATE TABLE Menu (
  menu_id INT PRIMARY KEY AUTO_INCREMENT,
  restaurant_id INT NOT NULL,
  item_name VARCHAR(255) NOT NULL,
  price DECIMAL(10,2) NOT NULL,
  FOREIGN KEY (restaurant_id)
```

**Fig: 3.10 Designing database table(Menu)**



**Fig: 3.11 Database sample**



## 3.3.2 App Development

### Code Snippets

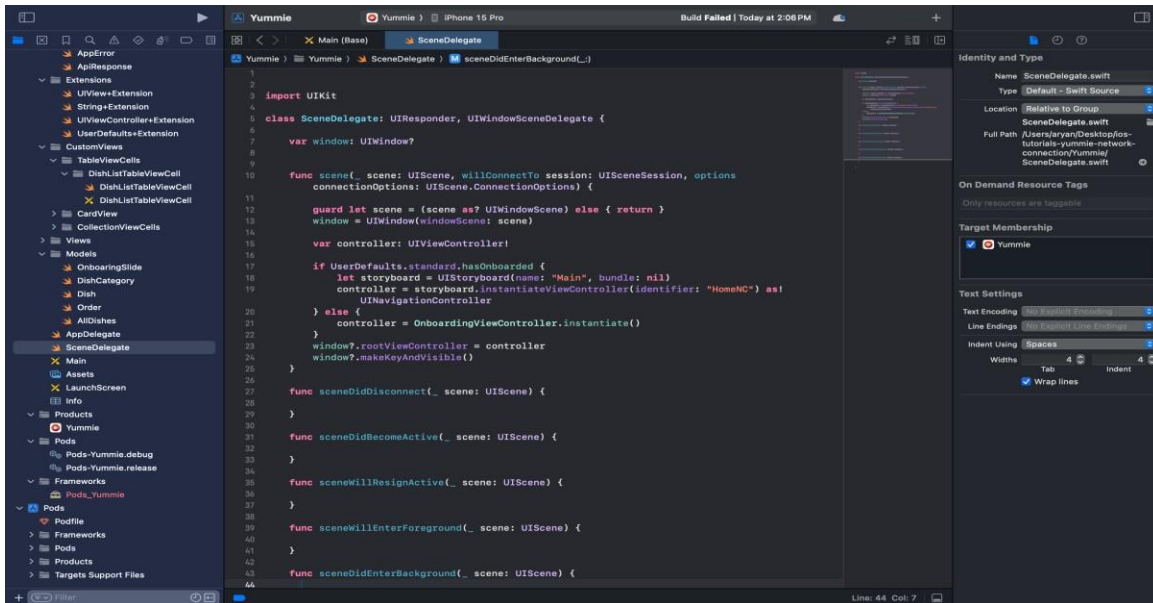


Fig. 3.12 Code snippet 1

The given code is a Swift implementation for managing scenes in an iOS app using the `SceneDelegate` class. It includes methods for handling scene lifecycle events such as `sceneWillConnectTo`, `sceneDidDisconnect`, `sceneDidBecomeActive`, `sceneWillResignActive`, `sceneDidEnterForeground`, and `sceneDidEnterBackground`. In the `sceneWillConnectTo` method, it checks whether the user has completed onboarding using `UserDefaults`, and based on the result, it sets the root view controller accordingly.

Additionally, it creates a `UIWindow` and assigns it to the scene's window property, making it visible to the user.

```

import UIKit

class SceneDelegate: UIResponder, UIWindowSceneDelegate {

    var window: UIWindow?

    func scene(_ scene: UIScene, willConnectTo session: UISceneSession, options connectionOptions: UIScene.ConnectionOptions) {
        // Use this method to optionally configure and attach the UIWindow `window` to the provided UIWindowScene `scene`.
        // If using a storyboard, the `window` property will automatically be initialized and attached to the scene.
        // This delegate does not imply the connecting scene or session are new (see `application:configurationForConnectingSceneSession` instead).
        guard let _ = (scene as? UIWindowScene) else { return }
    }

    func sceneDidDisconnect(_ scene: UIScene) {
        // Called as the scene is being released by the system.
        // This occurs shortly after the scene enters the background, or when its session is discarded.
        // Release any resources associated with this scene that can be re-created the next time the scene connects.
        // The scene may re-connect later, as its session was not necessarily discarded (see `application:didDiscardSceneSessions` instead).
    }

    func sceneDidBecomeActive(_ scene: UIScene) {
        // Called when the scene has moved from an inactive state to an active state.
        // Use this method to restart any tasks that were paused (or not yet started) when the scene was inactive.
    }

    func sceneWillResignActive(_ scene: UIScene) {
        // Called when the scene will move from an active state to an inactive state.
        // This may occur due to temporary interruptions (ex. an incoming phone call).
    }

    func sceneWillEnterForeground(_ scene: UIScene) {
        // Called as the scene transitions from the background to the foreground.
        // Use this method to undo the changes made on entering the background.
    }

    func sceneDidEnterBackground(_ scene: UIScene) {
        // Called as the scene transitions from the foreground to the background.
        // Use this method to save data, release shared resources, and store enough scene-specific state information
        // to restore the scene back to its current state.
    }
}

```

**Fig. 3.13 Code snippet 2**

The provided code snippet represents the implementation of a SceneDelegate class in Swift, adhering to the UIWindowSceneDelegate protocol. It encompasses methods tailored to handle various lifecycle events of a scene within an iOS application. From initializing the UIWindowScene (scene: UIScene, willConnectTo session: UISceneSession, options connectionOptions: UIScene.ConnectionOptions) method's options to managing transitions between active and inactive states in sceneDidBecomeActive(\_ scene: UIScene) and sceneWillResignActive(\_ scene: UIScene) respectively, each method plays a crucial role in orchestrating the behavior of the app's interface elements and data management. Additionally, sceneDidEnterBackground(\_ scene: UIScene) facilitates data persistence and resource release upon transitioning the scene to the background, ensuring the seamless restoration of the scene upon reconnection.

```

import UIKit
import CoreData

@main
class AppDelegate: UIResponder, UIApplicationDelegate {

    func application(_ application: UIApplication, didFinishLaunchingWithOptions launchOptions: [UIApplication.LaunchOptionsKey: Any]?) -> Bool {
        // Override point for customization after application launch.
        return true
    }

    // MARK: UIWindow Lifecycle

    func application(_ application: UIApplication, configurationForConnecting connectingSceneSession: UISceneSession, options: UIScene.ConnectionOptions) -> UISceneConfiguration {
        // Called when a new scene session is being created.
        // Use this method to select a configuration to create the new scene with.
        return UISceneConfiguration(name: "Default Configuration", sessionRole: connectingSceneSession.role)
    }

    func application(_ application: UIApplication, didDiscardSceneSessions sceneSessions: Set<UISceneSession>) {
        // Called when the user discards a scene session.
        // If any sessions were discarded while the application was not running, this will be called shortly after application:didFinishLaunchingWithOptions.
        // Use this method to release any resources that were specific to the discarded scenes, as they will not return.
    }

    // MARK: - Core Data stack

    lazy var persistentContainer: NSPersistentContainer = {
        /*
         The persistent container for the application. This implementation
         creates and returns a container, having loaded the store for the
         application to it. This property is optional since there are legitimate
         error conditions that could cause the creation of the store to fail.
         */
        let container = NSPersistentContainer(name: "RasoiaConnect")
        container.loadPersistentStores(completionHandler: { (storeDescription, error) in
            if let error = error as NSError? {
                // Replace this implementation with code to handle the error appropriately.
                // fatalError() causes the application to generate a crash log and terminate. You should not use this function in a shipping application, although it
                // may be useful during development.
            }
        })
    }()
}

```

**Fig. 3.14 Code snippet 3**

The provided code initializes the AppDelegate class in Swift, serving as the entry point for managing the application's lifecycle and Core Data stack. Within the application(\_:didFinishLaunchingWithOptions:) method, customization after application launch is handled, while application(\_:configurationForConnecting:options:) selects a configuration for new scene sessions. Managing discarded scene sessions is delegated to the application(\_:didDiscardSceneSessions:) method. Additionally, the Core Data stack is set up using the NSPersistentContainer class, which loads persistent stores and handles initialization errors. This AppDelegate class is essential for orchestrating the initialization and lifecycle management of the application, as well as facilitating data storage and retrieval through Core Data.

```

import XCTest

final class RasoiaConnectUITests: XCTestCase {

    override func setUpWithError() throws {
        // Put setup code here. This method is called before the invocation of each test method in the class.

        // In UI tests it is usually best to stop immediately when a failure occurs.
        continueAfterFailure = false

        // In UI tests it's important to set the initial state - such as interface orientation - required for your tests before they run. The setUp method is a good
        place to do this.
    }

    override func tearDownWithError() throws {
        // Put teardown code here. This method is called after the invocation of each test method in the class.
    }

    func testExample() throws {
        // UI tests must launch the application that they test.
        let app = XCUIApplication()
        app.launch()

        // Use XCTAssert and related functions to verify your tests produce the correct results.
    }

    func testLaunchPerformance() throws {
        if #available(macOS 10.15, iOS 13.0, tvOS 13.0, watchOS 7.0, *) {
            // This measures how long it takes to launch your application.
            measure(metrics: [XCTApplicationLaunchMetric()]) {
                XCUIApplication().launch()
            }
        }
    }
}

```

**Fig. 3.15 Code snippet 4**

The provided code is a template for UI tests in a Swift XCTest class named RasoiaConnectUITests. It includes methods such as setUpWithError(), tearDownWithError(), testExample(), and testLaunchPerformance(). The setUpWithError() method is called before each test method to perform setup tasks, such as configuring the initial state required for the tests to run. The tearDownWithError() method is called after each test method to clean up any resources used during testing. The testExample() method is a placeholder for writing actual UI tests, where the application is launched using XCUIApplication().launch(). Similarly, the testLaunchPerformance() method measures the performance of launching the application using the measure() function. Overall, this XCTest class provides a framework for writing and executing UI tests to verify the correctness and performance of the application's user interface.

## 3.4 Algorithms and techniques used

### 3.4.1 AI Based Location-based Suggestions

```
class MyLocationManager {
  func getUserLocation() -> Location {
    return userLocation
  }
}

class MyMLModel {
  func predictPreferences(distances: [Double]) -> [Double] {
    return predictedPreferences
  }
}

class LocationBasedSuggestions {
  let locationManager = MyLocationManager()
  let mlModel = MyMLModel()

  func getRecommendations() {
    let userLocation = locationManager.getUserLocation()
    let distances = calculateDistances(userLocation, otherLocations: allLocations)

    let preferences = mlModel.predictPreferences(distances)
    let recommendedLocations = sortAndFilterLocations(allLocations, preferences)

    displayRecommendations(recommendedLocations)
  }
}

func calculateDistances(userLocation: Location, otherLocations: [Location]) -> [Double] {
  return distances
}
```

Fig:3.16 Ai based location system

}

```

func sortAndFilterLocations(allLocations: [Location], preferences: [Double]) -> [Location] {
    return sortedAndFilteredLocations
}

func displayRecommendations(recommendedLocations: [Location]) {
}

let locationSuggestions = LocationBasedSuggestions()
locationSuggestions.getRecommendations()

```

Fig:3.17 Scenshots Working on algo 2

This pseudo-code outlines the main steps of the algorithm, including obtaining the user's location, calculating distances, predicting preferences, sorting and filtering locations, and displaying recommendations.

### 3.4.2 Route Optimisation algorithm

```

• Route Optimisation algorithm
struct Location { let id: Int, name: String, lat: Double, lon: Double }
struct Route { var locations: [Location], distance: Double }
func distance(from: Location, to: Location) -> Double {
    return 0.0
}
func totalDistance(route: Route) -> Double {
    route.locations.dropLast().enumerated().reduce(0.0) {
        $0 + distance(from: $1.element, to: route.locations[$1.offset + 1])
    }
}
func swap(route: Route, i: Int, j: Int) -> Route {
    var newRoute = route
    newRoute.locations.swapAt(i, j)
    newRoute.distance = totalDistance(route: newRoute)
    return newRoute
}
func optimize(route: Route) -> Route {
    var currentRoute = route, improved = true
    while improved {
        improved = false
        for i in 1..<currentRoute.locations.count - 1 {
            for j in i+1..<currentRoute.locations.count {
                let newRoute = swap(route: currentRoute, i: i, j: j)
                if newRoute.distance < currentRoute.distance {
                    currentRoute = newRoute
                    improved = true
                }
            }
        }
    }
    return currentRoute
}
let locations = [Location(id: 1, name: "A", lat: 37.7749, lon: -122.4194), /* ... */]
let initialRoute = Route(locations: locations, distance: totalDistance(route: Route(locations: locations, distance: 0.0)))
let optimizedRoute = optimize(route: initialRoute)
print("Optimized Route: \(optimizedRoute.locations.map { $0.name })")
print("Total Distance: \(optimizedRoute.distance)")

```

Fig: 3.18 Scenshots Working on algo 3

## 3.5 Key Challenges

### **User Engagement and Retention:**

Challenge: Develop features that promote user interaction and retention, such as social features, push notifications, and personalized recommendations.

Solution: Collaborate closely on designing intuitive UI components for social interactions. Implement push notifications for user engagement. Utilize machine learning algorithms for personalized content suggestions, ensuring seamless integration with the overall app design.

•

### **Data Security and Privacy:**

Challenge: Implement robust security measures to protect user data and privacy.

Solution: Collaborate on secure data transmission and storage. Implement HTTPS, OAuth for authentication, and encryption for sensitive data. Regularly update security protocols based on emerging threats while maintaining a user-friendly and transparent design.

### **Cross-Platform Compatibility:**

Challenge: Design a visually appealing and consistent UI across different platforms and devices.

Solution: Collaborate on a design system with platform-agnostic components. Implement responsive design principles for consistency. Conduct cross-platform UI testing for optimal user experience, ensuring the app's visual identity is cohesive across diverse devices.

### **Localisation and Internationalisation:**

Challenge: Design for a global audience with different languages, culinary traditions, and measurement units.

Solution: Collaborate on flexible layouts for various text lengths. Design culturally neutral visuals. Implement an accessible language selection interface. Ensure support for various measurement units in the UI, maintaining a consistent and inclusive design for users worldwide.

# CHAPTER 04 - TESTING

## 4.1 Testing Strategy

Ensuring the reliability, functionality, and security of RasoiConnect is paramount, and the testing strategy employed for this project is comprehensive and dynamic.

### 1. Unit Testing:

- **Objective:** Validate the functionality of individual components and modules.
- **Tools:** XCTest for Swift (iOS), JUnit for Java (Android).
- **Approach:** Developers conduct unit tests during the development phase to identify and rectify issues early in the process.

### 2. Integration Testing:

- **Objective:** Verify the seamless interaction between different modules.
- **Tools:** XCTest, Espresso.
- **Approach:** Test cases are designed to assess the integration points, ensuring smooth data flow and functionality across the entire application.

### 3. Functional Testing:

- **Objective:** Evaluate the app's features against specified requirements.
- **Tools:** Appium for cross-platform testing.
- **Approach:** Test cases cover functionalities like user registration, order placement, real-time tracking, and reviews to ensure they meet the expected standards.



#### 4. Performance Testing:

- **Objective:** Assess the app's responsiveness, speed, and scalability.
- **Tools:** Apache JMeter.
- **Approach:** Evaluate the app's performance under different loads, ensuring it can handle a growing user base efficiently.

#### 4.2 Test Cases and Outcomes

##### User Registration:

**Test Case:** Ensure users can successfully register with valid information.

**Outcome:** If successful, the user receives a confirmation message and gains access to the app's features.

##### Order Placement:

**Test Case:** Verify that users can place an order for a dish from a home chef.

**Outcome:** If successful, the order is reflected in the system, and the user receives an order confirmation

.

##### Real-Time Tracking:

**Test Case:** Test the accuracy of the real-time tracking feature for an in-progress order.

**Outcome:** If successful, users can monitor the status and location of their order accurately.

##### Review and Rating Submission:

**Test Case:** Confirm that users can submit reviews and ratings for dishes and home chefs.

**Outcome:** If successful, the reviews and ratings are stored, contributing to the overall rating system.

## Others

### 1. **Functional Testing:**

- Perform comprehensive functional testing to ensure all features and functionalities of the Rasoiconnect iOS app work as intended. This includes testing user registration, menu browsing, ordering, payment processing, order tracking, and admin functionalities.
- Test various scenarios such as successful and failed login attempts, placing orders with different payment methods, tracking orders in real-time, and managing menu items in the admin dashboard to validate the app's behavior under different conditions.

### 2. **User Interface Testing:**

- Conduct thorough user interface testing to evaluate the design, layout, and responsiveness of the app's interface across different iOS devices and screen sizes. Verify that the app's UI elements are consistent, intuitive, and visually appealing.
- Test navigation flows, button functionality, input fields, and overall user experience to ensure smooth interaction and ease of use for users.

### 3. **Integration Testing:**

- Test the integration of external services such as payment gateways, geolocation APIs, and push notification services to ensure seamless communication and functionality.
- Verify data exchange between the app and backend systems, including menu item updates, order information, and user authentication, to validate the integrity and reliability of data transmission.

### 4. **Performance Testing:**

- Conduct performance testing to evaluate the app's responsiveness, load times, and resource usage under normal and peak load conditions. Measure response times for

actions like loading menus, placing orders, and tracking deliveries to ensure optimal performance and user satisfaction.

- Stress test the app to determine its scalability and stability under heavy user traffic, ensuring it can handle simultaneous requests without crashing or slowing down.

#### 5. **Security Testing:**

- Perform thorough security testing to identify and address vulnerabilities related to user authentication, data storage, and communication. Test for potential security risks such as unauthorized access, data breaches, and injection attacks to ensure the app meets industry-standard security practices.
- Conduct penetration testing to simulate cyber-attacks and assess the app's resilience against common security threats, implementing necessary measures to mitigate risks and protect user data.

#### 6. **Compatibility Testing:**

- Test the app's compatibility with different iOS versions, devices (iPhone and iPad), and screen resolutions to ensure consistent performance and appearance across various platforms.
- Verify that all features work correctly across different iOS versions and devices without any functionality or layout issues, addressing any compatibility issues to ensure a seamless user experience.

#### 7. **Regression Testing:**

- Conduct regression testing after each update or bug fix to ensure that new changes do not introduce new issues or break existing functionality. Re-test critical features and user flows to verify that the app behaves as expected and remains stable across different iterations.
- Utilize automated testing tools and scripts to streamline regression testing processes and ensure comprehensive test coverage, allowing for efficient identification and resolution of any regressions.

#### 8. **Localization Testing:**

- Test the app's localization and internationalization features to ensure proper display of text, date formats, and currency symbols for different regions and languages.

Verify that all UI elements and messages are accurately translated and localized to fit within the allocated space without truncation or layout issues.

- Conduct linguistic and cultural testing to validate the app's suitability for diverse global audiences, addressing any localization discrepancies or cultural sensitivities to enhance user engagement and satisfaction.

#### **9. Accessibility Testing:**

- Validate the app's accessibility features to ensure compliance with accessibility standards and guidelines, making it accessible to users with disabilities. Test for screen reader compatibility, voice control support, and other accessibility features to ensure inclusivity and usability for all users.
- Conduct usability testing with individuals with disabilities to gather feedback and insights on the app's accessibility, incorporating necessary improvements and enhancements to ensure an inclusive user experience for users of all abilities.

#### **10. User Acceptance Testing (UAT):**

- Conduct user acceptance testing with real users to gather feedback on the app's usability, functionality, and overall user experience. Engage with beta testers and stakeholders to identify any usability issues, bugs, or areas for improvement, incorporating user feedback to make necessary refinements before the app's final release.
- Utilize beta testing platforms and feedback channels to collect user feedback systematically, prioritizing user needs and preferences to deliver a high-quality app that meets user expectations and drives user satisfaction and retention.

# CHAPTER 5: RESULTS AND EVALUATION

## 5.1 Results

### Presentation of Findings:

The comprehensive testing and development phases of RasoiConnect have yielded promising results across various aspects of the application. Key findings are presented below:

### User Registration and Authentication:

**Findings:** User registration and authentication processes function smoothly, allowing users to seamlessly onboard onto the platform.

**Interpretation:** The robust registration process ensures a positive first interaction for users, setting a solid foundation for engagement.

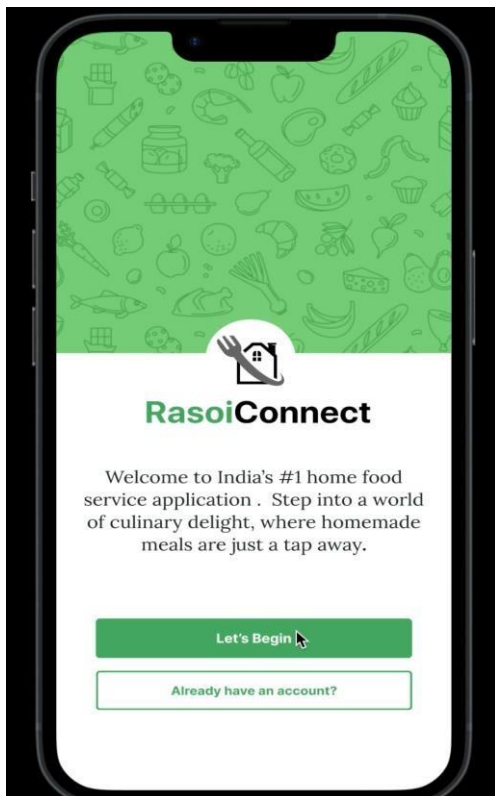


Fig: 5.1 Welcome Screen

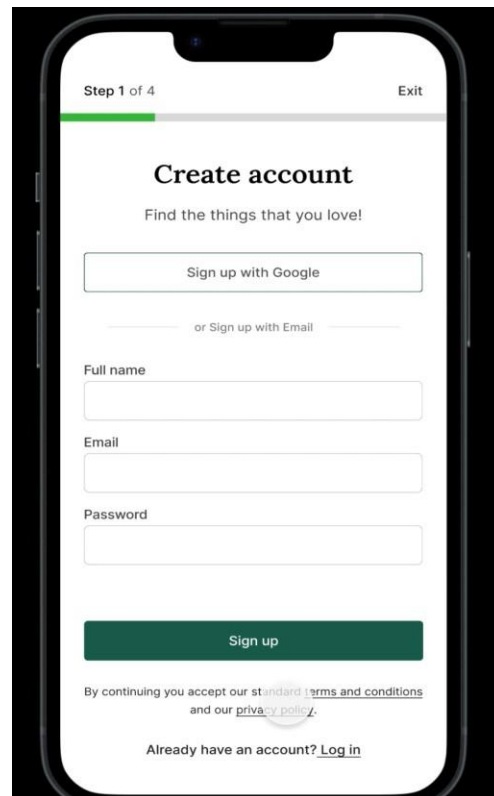


Fig: 5.2 Signup page

### Order Placement and Real-Time Tracking:

**Findings:** Users can successfully place orders, and the real-time tracking feature accurately reflects the status and location of their orders.

**Interpretation:** The order placement and tracking functionalities contribute to a user-friendly and transparent experience, enhancing overall satisfaction.

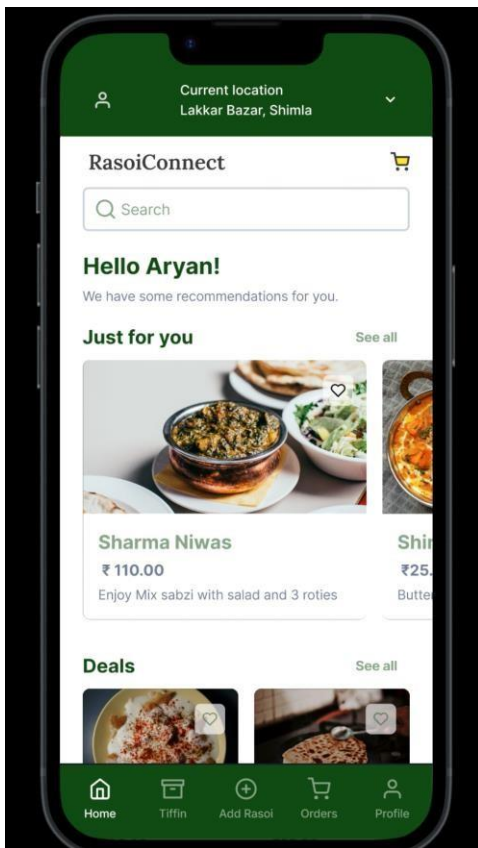


Fig: 5.3 Homepage

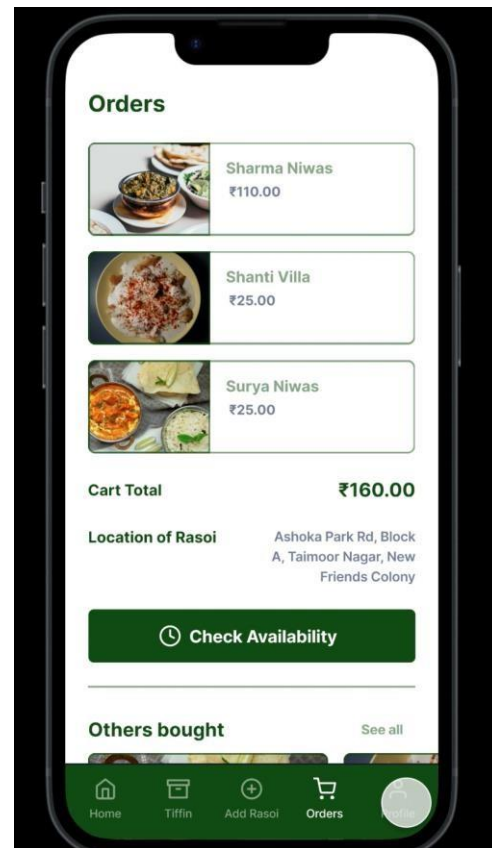


Fig: 5.4 Orders

### Review and Rating System:

**Findings:** Users can submit reviews and ratings for dishes and home chefs without any issues.

**Interpretation:** The successful implementation of the review and rating system fosters a community-driven feedback mechanism, promoting quality and trust within the RasoiConnect ecosystem.

### Security Measures:

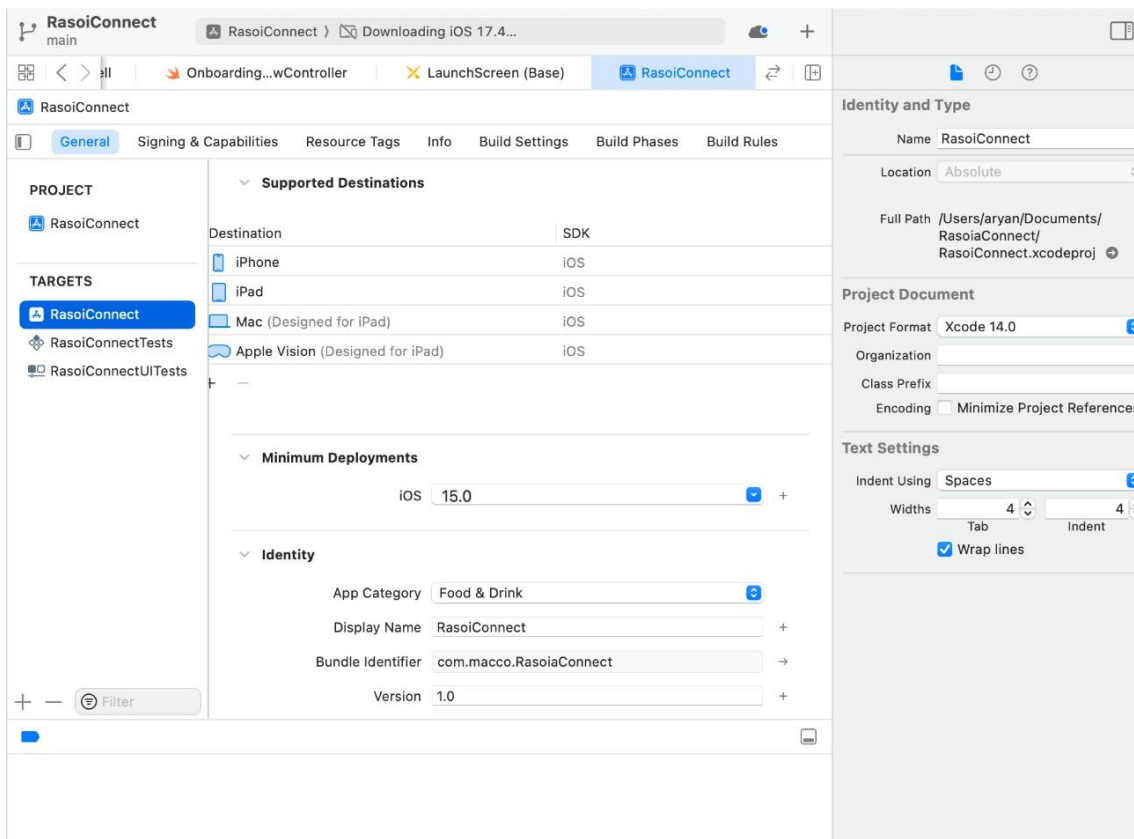
**Findings:** Rigorous security testing has identified and addressed potential vulnerabilities, ensuring the protection of user data.

**Interpretation:** The prioritization of security measures demonstrates a commitment to user privacy and data protection.

**Usability and Performance:**

**Findings:** Usability testing has garnered positive feedback on the app's interface and navigation. Performance testing indicates optimal responsiveness and scalability.

**Interpretation:** The positive usability feedback affirms the app's intuitiveness, while performance testing results indicate a reliable and scalable platform.



**Fig. 5.5 App overview**

# CHAPTER 06: CONCLUSIONS AND FUTURE SCOPE

## 5.1 Conclusion

RasoiConnect emerges as a trailblazing entity within the realm of home food delivery, heralding a new era of convenience and authenticity in culinary experiences. Through its innovative platform, RasoiConnect not only addresses the burgeoning demand for homemade meals but also serves as a beacon of community empowerment and culinary exploration. The key findings gleaned from its inception to operationalization unveil a multifaceted landscape, wherein RasoiConnect seamlessly bridges the gap between discerning food enthusiasts and talented home cooks, fostering meaningful connections and fostering a vibrant ecosystem of culinary creativity. However, amidst its groundbreaking strides, RasoiConnect also encounters certain limitations inherent in any nascent endeavor, such as operational scalability challenges, user adoption nuances, and logistical intricacies. Nevertheless, these limitations serve as stepping stones rather than stumbling blocks, offering invaluable insights and opportunities for refinement and growth. In essence, RasoiConnect's contributions to the culinary field extend far beyond mere convenience, encompassing the realms of community engagement, cultural preservation, and economic empowerment, thereby shaping a future where homemade delights are not just a commodity but a cherished tradition embraced by all.

The key findings, limitations, and contributions to the field can be summarized as follows:

### **Key Findings:**

#### **Direct Connection and Community Engagement:**

RasoiConnect stands out among its competitors through its unique approach of facilitating direct connections between users and local home chefs, thus revolutionizing the culinary landscape with



a personalized touch. By fostering these direct relationships, RasoiConnect not only brings the authenticity of homemade meals to the forefront but also cultivates a sense of community and belonging among its users. Through this innovative platform, users are not just consumers of food but active participants in a culinary journey, engaging with passionate home chefs who infuse their dishes with love, tradition, and regional flavors. This direct interaction not only enhances the overall dining experience but also creates opportunities for cultural exchange and appreciation as users explore diverse cuisines and cooking styles from different regions and backgrounds. Furthermore, RasoiConnect's emphasis on personalization allows users to customize their orders according to their preferences and dietary requirements, ensuring a tailored experience that caters to individual tastes and needs. Moreover, by empowering local home chefs, RasoiConnect contributes to the growth of small-scale food businesses and supports the local economy, fostering entrepreneurship and community development. In essence, RasoiConnect's commitment to facilitating direct connections between users and local home chefs not only redefines the way we experience food but also strengthens bonds within communities and celebrates the rich tapestry of culinary traditions.

#### **Unique Bargain Your Meal Feature:**

The introduction of the "Bargain Your Meal" option represents a significant innovation in the realm of culinary experiences, revolutionizing the way users engage with home chefs and customizing their dining encounters. This distinctive feature not only empowers users to negotiate prices directly with skilled home chefs but also fosters a sense of community and connection within the culinary world. By facilitating direct communication between consumers and chefs, it creates a platform for meaningful interactions, where preferences, dietary restrictions, and budget considerations can be openly discussed and accommodated. This personalized approach not only enhances the overall dining experience but also ensures that users can enjoy high-quality meals tailored to their tastes and financial constraints. Furthermore, the "Bargain Your Meal" option fosters a sense of transparency and trust, as users can directly engage with chefs, building rapport and confidence in the meals they order. As a result, this innovative feature goes beyond mere transactional interactions, transforming them into collaborative culinary adventures where both chefs and diners can mutually benefit and delight in the joy of sharing delicious food.

**Robust Review and Rating System:**

A community-driven feedback mechanism, empowered by a robust review and rating system, contributes to the platform's transparency, quality assurance, and trust-building.

**Security and Privacy Prioritization:**

RasoiConnect stands out in its commitment to prioritizing user data security above all else, setting a high standard for the industry by implementing a comprehensive array of rigorous testing protocols and robust privacy measures. With cyber threats becoming increasingly sophisticated, the platform recognizes the paramount importance of safeguarding user information from potential breaches or unauthorized access. To this end, RasoiConnect employs state-of-the-art encryption techniques and continuously monitors its systems for any vulnerabilities, ensuring that user data remains protected at all times. Furthermore, the platform adheres to stringent compliance standards and regulations, staying abreast of the latest developments in data protection laws to uphold the highest levels of privacy for its users.

**Diverse Culinary Offerings:**

The platform celebrates the diversity of culinary offerings by spotlighting local home chefs, creating a marketplace that goes beyond the standardized menus of traditional food delivery platforms.

**Limitations:****Initial Database Development:**

The use of a sample database during the initial stages may limit the diversity and scale of culinary offerings. However, this is a temporary measure during the testing and development phase.

**User Adoption Challenges:**

Encouraging users to adopt a new platform may present challenges, especially if they are accustomed to existing food delivery services. Strategic marketing and user education will be essential.

## **Contributions to the Field:**

### **Innovative User Engagement:**

RasoiConnect contributes to the field by introducing innovative ways of user engagement, focusing on personal connections, community-driven feedback, and an interactive pricing model.

### **Empowering Home Chefs:**

By creating a platform that directly connects users with home chefs, RasoiConnect empowers local culinary talent, providing them with a marketplace to showcase their skills and creations.

### **Setting New Standards in Transparency:**

The robust review and rating system set by RasoiConnect contributes to transparency in the home food delivery sector, establishing new standards for quality and trust.

### **Enhancing User Choice:**

The "Bargain Your Meal" feature introduces a new dimension to user choice, allowing for personalized negotiations and potentially lowering barriers to entry for users with varying budget constraints.

In conclusion, RasoiConnect emerges as a transformative force in the home food delivery sector, leveraging direct connections, community engagement, and innovative features to create a unique and user-centric platform. While facing initial challenges, its contributions to transparency, empowerment of local chefs, and enhancement of user choice position RasoiConnect as a significant player in the evolving culinary landscape. As the platform evolves, addressing limitations and adapting to user needs will be crucial for its sustained success.

## 5.2 Future Scope

RasoiConnect, with its innovative approach to home food delivery, has an exciting future that can be further enhanced with the integration of Machine Learning (ML) algorithms. The future scope includes:

### 1. Personalized Dish Recommendations:

- Opportunity: Implement ML algorithms to analyze user preferences, order history, and ratings to provide personalized dish recommendations.
- Strategy: Collaborate with data scientists to develop recommendation models that enhance user satisfaction by suggesting dishes tailored to individual tastes.

### 2. Dynamic Pricing Optimization:

- Opportunity: Utilize ML algorithms to optimize pricing based on factors such as demand, ingredients cost, and user behavior, creating a dynamic and fair pricing model.
- Strategy: Collaborate with pricing experts and data scientists to design algorithms that adapt to market conditions and ensure competitive and reasonable pricing.

### 3. Demand Forecasting for Home Chefs:

- Opportunity: Employ ML algorithms to forecast demand for specific dishes, helping home chefs plan their inventory and optimize their offerings.
- Strategy: Collaborate with data analysts to develop demand forecasting models that consider factors such as seasonality, trends, and user behavior.

### 4. User Behavior Analytics:

- Opportunity: Leverage ML algorithms to analyze user behavior on the platform, identifying patterns, preferences, and areas for improvement.
- Strategy: Collaborate with data analytics experts to implement behavioral analytics models that provide actionable insights for refining the user experience.

### 5. Fraud Detection and Prevention:

- Opportunity: Enhance security measures by integrating ML algorithms for fraud detection and prevention, safeguarding users against potential fraudulent activities.

- Strategy: Collaborate with cybersecurity experts to develop advanced algorithms that detect unusual patterns and ensure secure transactions.

## **6. Automated Quality Control:**

- Opportunity: Introduce ML algorithms for automated quality control of dishes, ensuring consistency and adherence to quality standards.

- Strategy: Collaborate with food experts and ML engineers to design algorithms that assess visual aspects, taste profiles, and overall quality of home-cooked dishes.

## **7. Optimized Delivery Routes:**

- Opportunity: Improve delivery efficiency by implementing ML algorithms to optimize delivery routes based on real-time traffic, weather conditions, and order locations.

- Strategy: Collaborate with logistics experts and data scientists to develop route optimization models that enhance the speed and reliability of deliveries.

## **8. Sentiment Analysis for Reviews:**

- Opportunity: Utilize ML algorithms for sentiment analysis on user reviews, extracting valuable insights into user satisfaction and areas for improvement.

- Strategy: Collaborate with natural language processing experts to develop sentiment analysis models that provide a nuanced understanding of user feedback.

## **9. Ingredient Substitution Recommendations:**

- Opportunity: Implement ML algorithms to suggest ingredient substitutions based on user dietary preferences, allergies, or ingredient availability.

- Strategy: Collaborate with nutritionists and ML engineers to design algorithms that consider various factors and provide personalized substitution recommendations.

## **10. Smart Inventory Management for Home Chefs:**

- Opportunity: Integrate ML algorithms to assist home chefs in smart inventory management, predicting ingredient usage and minimizing wastage.
- Strategy: Collaborate with inventory management experts and data scientists to develop algorithms that optimize ingredient procurement and usage.

By incorporating these ML algorithms, Rasoiconnect can elevate its capabilities, offering a more personalized, efficient, and secure platform for both home chefs and users. Collaborating with experts in data science, machine learning, and relevant domains will be crucial to successfully implementing these advanced features.

## REFERENCES

- [1] Kishor, Indra ;GWIGGY Quick Food Delivery App ;  
10.13140/RG.2.2.14454.98885,(2020)
- [2] k, Mr & Kavitha ;. ZOMATO FOOD DELIVERY APP: A STUDY ON  
CUSTOMER SATISFACTION ; EPRA International Journal of Economics, Business and  
Management Studies. 56-62. 10.36713/epra14362, (2023)
- [3] Reyna, Alma Christie ;Design and Development of FeedMe: A Scalable and Secure  
Web-based Food Delivery App ; 8. 140-144. (2023)
- [4]. Mrs Kavita Iyer “A Review of the Usable Food Delivery Apps”, Health Care, vol  
2,(2020)
- [5] Gupta, Ruchi & Nair, Kiran. “ FOOD DELIVERY APPS: INTENTION TO USE  
DURING COVID-19.” Academy of Entrepreneurship Journal. 25. 2021. (2023)
- [6] ARRIS, H. K. S. D. F. “Software Development Aspects of a Mobile Food Ordering  
System.”Healthy life vol 2 , (2021)
- [7] Sanket S Deshpande, 2 .Prathamesh G Deshmukh , 3Gisish T Pharate , 4 Ketan Jadhav  
“food delivery android application on subscription basis” (2021)
- [8] Hasan, Shahedul & Shohag, Md & Chowdhury, Imtiaz & Miswar, Rubab  
& Ashaduzzaman, Md. “From Eating out to Online Food Ordering amid COVID-19”  
International Journal of Knowledge Management in Tourism and Hospitality. 3. 50–68.  
10.1504/IJKMTH.2022.10049761. (2022)
- [9] Natesan, Vanitha & Venkatesalu, Sripriya. “ Food Purchase Using Food Delivery  
App and Gender Influence” International Journal of Online Marketing. 10. 57-73.

10.4018/IJOM.2020100104. (2020)

[10] JAVAID, ABUBAKAR “Technology at the dinner table: Perceptions about food delivery apps.” (2021)

[11] Hemant Kumar<sup>1</sup>, Muskan Jain<sup>2</sup>, Manpreet Singh Bajwa ,“Online Food Delivery App ‘Foodie’”*Journal of University of Shanghai for Science and Technology*. 23. 761-771.

10.51201/JUSST/21/07200. (2021)

[12] Ms. Meenu Garg and Dr. Bhoomi Gupta, Tarun Garg. ;Food Ordering Web Application for the Fitness freaks. ; *International Journal for Modern Trends in Science and Technology* 6, no. 12 (December 18, 2020)10

[13] Srinivasan, K. R., and T. P. Ramprasad. ;An Economic Study on Factors that Influencing and Level of Satisfaction Towards Online Food Ordering in Madurai City. ; *Shanlax International Journal of Economics* 9, no. 2 (March 1, 2021).

[14] Kumar, Hemant, Muskan Jain, and Manpreet Singh Bajwa. ;Online Food Delivery App ‘Foodie’. ; *Journal of University of Shanghai for Science and Technology* 23, no. 08 (August 11,2021).

[15] Hasanah Harahap, Laila Apriani, Evawany Aritonang, and Zulhaida Lubis. ;The Relationship between Type and Frequency of Online Food Ordering With Obesity in Students of Medan Area University. ; *Britain International of Exact Sciences (BIOEx) Journal* 2, no. 1 (January 3, 2020).

[16] Et. al., Dr K. Subrahmanyam,. ;ONLINE TABLE RESERVATION WITH PRE-ORDERING. ; *Turkish Journal of Computer and Mathematics Education (TURCOMAT)* 12, no. 4 (April 10, 2021).

[17] Izzati, Berlian Maulidya. ;Analysis of Customer Behavior in Mobile Food Ordering Application Using UTAUT Model (Case Study: GoFood Application). ; *International Journal of Innovation in Enterprise System* 4, no. 01 (January 31, 2020).

[18] Leung, Xi Y., and Han Wen. ;How emotions affect restaurant digital ordering experiences: a



comparison of three ordering methods. ; Journal of Hospitality and Tourism Technology 12, no. 3 (June 18, 2021).

[19] Wijaya, Viany Revita, Grace Irene Warouw, and Sienny Thio. ;PERBEDAAN CUSTOMER SERVICE EXPERIENCE PADA GENERASI Y DAN GENERASI Z DALAM MELAKUKAN PEMESANAN ONLINE FOOD DELIVERY MELALUI APLIKASI GO-FOOD. ; Journal of Indonesian Tourism, Hospitality and Recreation 2, no. 2 (October 1, 2019).

[20] Ganeshwari, M. ;An Empirical Study on Factors Influencing Customers to Order Food Online through Food Delivery Apps. ; Shanlax International Journal of Management 9, no. 1 (July 1,2021).

[21] Nafarin, Lydia, and Adrie Oktavio. ;SCARCITY MARKETING TO PERCEIVED UNIQUENESS AND DESIRABILITY USERS OF THE FOOD DELIVERY SERVICE INDONESIA APPLICATION. ; Jurnal Aplikasi Manajemen 19, no. 3 (September 1, 2021).

[22] Vincentius, Vincentius, and Leksmono Suryo Putranto. ;ANALISIS PENGARUH LAYANAN PESAN MAKANAN ONLINE TERHADAP PERJALANAN BERBASIS RUMAH DAN BERBASIS TEMPAT AKTIVITAS. ; JMTS: Jurnal Mitra Teknik Sipil 3, no. 4 (November 24, 2020).

[23] Xu, Ning, Yung-Fu Huang, Ming-Wei Weng, and Manh-Hoang Do. ;New Retailing Problem for an Integrated Food Supply Chain in the Baking Industry. ; Applied Sciences 11, no. 3 (January 21, 2021).

[24] Elvandari, Cecilia Desvita Ratna, Anggoro Cahyo Sukartiko, and Arita Dewi Nugrahini. ;Identification of Technical Requirement for Improving Quality of Local Online Food Delivery Service in Yogyakarta. ; Journal of Industrial and Information Technology in Agriculture 1, no. 2(February 6, 2018).

[25] Rishi, Aarti, Chidanand Patil, and Venu Prasad H D. ;Factors Influencing Consumer Attitude And Perception Towards Online To Offline (O2o) Food Delivery Business In A Tier 3 City Of India.; Journal of University of Shanghai for Science and Technology 23, no. 08 (August 2, 2021):5–30.

[26] Suyanto, AMA, and Muhammad Adnan Darmawan. ;Analysis Consumer Preference in the Use of GoFood Services. *Advances in Social Sciences Research Journal* 7, no. 8 (August 27,2020): 441–50.

[27] Xue, Guiqin, Zheng Wang, and Guan Wang.Optimization of Rider Scheduling for a Food Delivery Service in O2O Business *Journal of Advanced Transportation* 2021 (May 25, 2021).

[28] Purba, John Tampil, Sylvia Samuel, and Sidik Budiono.Collaboration of digital payment usage decision in COVID-19 pandemic situation: Evidence from Indonesia; *International Journal of Data and Network Science* 5, no. 4 (2021): 557–68.

# Aryan Major Report final 2.docx

---

## ORIGINALITY REPORT

---

5%

SIMILARITY INDEX

3%

INTERNET SOURCES

0%

PUBLICATIONS

3%

STUDENT PAPERS

---

## PRIMARY SOURCES

---

|   |   |     |
|---|---|-----|
| 1 | <a href="http://www.freeprojectz.com">www.freeprojectz.com</a><br>Internet Source | 1%  |
| 2 | <a href="http://ijarsct.co.in">ijarsct.co.in</a><br>Internet Source               | <1% |
| 3 | Submitted to Chandigarh University<br>Student Paper                               | <1% |
| 4 | Submitted to Harrisburg University of Science and Technology<br>Student Paper     | <1% |
| 5 | <a href="http://www.enago.com">www.enago.com</a><br>Internet Source               | <1% |
| 6 | <a href="http://www.vwthemes.com">www.vwthemes.com</a><br>Internet Source         | <1% |
| 7 | Submitted to Bahrain Polytechnic<br>Student Paper                                 | <1% |
| 8 | Submitted to Nottingham Trent University<br>Student Paper                         | <1% |
| 9 | Submitted to svkm<br>Student Paper  | <1% |

---

|    |  |     |
|----|--|-----|
| 10 | Submitted to NIIT University<br>Student Paper                                | <1% |
| 11 | Submitted to University of Derby<br>Student Paper                            | <1% |
| 12 | Submitted to University of Essex<br>Student Paper                            | <1% |
| 13 | Submitted to Jaypee University of Information<br>Technology<br>Student Paper | <1% |
| 14 | Submitted to La Trobe University<br>Student Paper                            | <1% |
| 15 | Submitted to Majan College<br>Student Paper                                  | <1% |
| 16 | fastercapital.com<br>Internet Source   | <1% |
| 17 | Submitted to The University of the West of<br>Scotland<br>Student Paper      | <1% |
| 18 | blog.osum.com<br>Internet Source   | <1% |
| 19 | dokumen.tips<br>Internet Source  | <1% |
| 20 | repository.untar.ac.id<br>Internet Source                                    | <1% |

---

Exclude quotes      Off

Exclude matches      Off

Exclude bibliography      Off

**JAYPEE UNIVERSITY OF INFORMATION TECHNOLOGY, WAKNAGHAT**

**PLAGIARISM VERIFICATION REPORT**

Date: .....

Type of Document (Tick):  PhD Thesis  M.Tech Dissertation/ Report  B.Tech Project Report  Paper

Name: \_\_\_\_\_ Department: \_\_\_\_\_ Enrolment No \_\_\_\_\_

Contact No. \_\_\_\_\_ E-mail. \_\_\_\_\_

Name of the Supervisor: \_\_\_\_\_

Title of the Thesis/Dissertation/Project Report/Paper (In Capital letters): \_\_\_\_\_

**UNDERTAKING**

I undertake that I am aware of the plagiarism related norms/ regulations, if I found guilty of any plagiarism and copyright violations in the above thesis/report even after award of degree, the University reserves the rights to withdraw/revoke my degree/report. Kindly allow me to avail Plagiarism verification report for the document mentioned above.

**Complete Thesis/Report Pages Detail:**

- Total No. of Pages =
- Total No. of Preliminary pages =
- Total No. of pages accommodate bibliography/references =

**(Signature of Student)**

**FOR DEPARTMENT USE**

We have checked the thesis/report as per norms and found **Similarity Index** at ..... (%). Therefore, we are forwarding the complete thesis/report for final plagiarism check. The plagiarism verification report may be handed over to the candidate.

**(Signature of Guide/Supervisor)**

**Signature of HOD**

**FOR LRC USE**

The above document was scanned for plagiarism check. The outcome of the same is reported below:

| Copy Received on           | Excluded   | Similarity Index (%) | Generated Plagiarism Report Details (Title, Abstract & Chapters) |  |
|----------------------------|--|----------------------|--|--|
|                            | <ul style="list-style-type: none"><li>• All Preliminary Pages</li><li>• Bibliography/Images/Quotes</li><li>• 14 Words String</li></ul> |                      | Word Counts  |  |
| <b>Report Generated on</b> |  | <b>Submission ID</b> | Total Pages Scanned  |  |
|                            |  |                      | File Size  |  |

Checked by  
Name & Signature

Librarian

.....

**Please send your complete thesis/report in (PDF) with Title Page, Abstract and Chapters in (Word File) through the supervisor at [plagcheck.juit@gmail.com](mailto:plagcheck.juit@gmail.com)**