

A Project Report On
SMART RESTAURANT MANAGEMENT SYSTEM (SRMS)

*Submitted in partial fulfilment of the requirements for the award of
degree
of
Bachelor of Technology
in
INFORMATION TECHNOLOGY
by*

Student Name	Student ID
1. N.Pavithra	19K61A1238
2. B.Sarika	19K61A1210
3. P.Hemanth	19K61A1243
4. B.Sai Kiran	19K61A1211

Under the esteemed Supervision of
Dr.S Phani Kumar

Associate Professor, Department of IT



Department of Information Technology



INSTITUTE OF
TECHNOLOGY &
ENGINEERING
TADEPALLIGUDEM
West Godavari District



NBA
NATIONAL BOARD OF
ACCREDITATION

KADAKATLA, TADEPALLIGUDEM -534101,

WEST GODAVARI, ANDHRA PRADESH.

Academic Year : 2022-2023



INSTITUTE OF
TECHNOLOGY &
ENGINEERING
TADEPALLIGudem
West Godavari District



Department of Information Technology

CERTIFICATE

*This is to certify that the project work entitled "**SMART RESTAURANT MANAGEMENT SYSTEM**"(SRMS) is being submitted by **N.Pavithra (19K61A1238), B.Sarika (19K61A1210), P.Hemanth (19K61A1243), B.Sai Kiran (19K61A1211** in partial fulfilment for the award of the degree of **BACHELOR OF TECHNOLOGY** in **Information Technology** affiliated to Jawaharlal Nehru Technological University, Kakinada during the academic year 2022 to 2023 is a record of bonafide work carried out by them under my guidance and supervision. The results presented in this thesis have been verified and are found to be satisfactory. The results embodied in this thesis have not been submitted to any other University or Institute for the award of any other degree or diploma.*

Project Guide

Dr. S.Phani Kumar
Associate Professor
Department of IT

Head of Department

Dr.K.Subhash Bhagavan
Associate Professor & HOD
Department of IT

External Examiner



INSTITUTE OF
TECHNOLOGY &
ENGINEERING
TADEPALLIGudem
West Godavari District



Department of Information Technology

DECLARATION BY THE CANDIDATES

We N.Pavithra (19K61A1238), B.Sarika (19K61A1210), P.Hemanth (19K61A1243), B.Sai Kiran (19K61A1211) here by declare the project report entitled "**SMART RESTAURANT MANAGEMENT SYSTEM" (SRMS)**" under esteemed supervision of **Dr S.Phani Kumar**, is submitted in partial fulfilment of the requirements for the award of the degree of Bachelor of Technology in Information Technology. This is a record of work carried out by us and the results embodied in this project have not been reproduced or copied from any source. The results embodied in this project report have not been submitted to any other University or Institute for the award of any other degree or diploma.

N.Pavithra (19K61A1238)

B.Sarika (19K61A1210)

P.Hemanth (19K61A1243)

B.Sai Kiran(19K61A1211)

ACKNOWLEDGEMENT

We pay obeisance to the dynamic Chairman of Sasi Educational Society, **Sri Burugupalli Venu Gopala Krishna**, for his inspiring presence, which has always been the principle driving force behind all over endeavors.

First of all, we would like to thank **Mr. Meka Narendra Krishna**, Vice-Chairman, Sasi Institute of Technology and Engineering, for his everlasting support.

We would like to thank **Dr. K Bhanu Prasad**, Director, Sasi Institute of Technology and Engineering, for his support.

We would like to thank **Dr. Mohammed Ismail**, Principal , Sasi Institute of Technology and Engineering, for his support.

We would like to thank **Dr.T J V Subrahmaneswara Rao**, Dean Students Affairs, Sasi Institute of Technology and Engineering, for his support.

We would like to thank **Dr. K Subhash Bhagavan**, Associate Professor and Head of the Department, Information Technology, Sasi Institute of Technology and Engineering for providing us with invaluable feedback on our work, which allowed us to constantly improve.

We are greatful to our project coordinator and also Guide **Dr.S Phani Kumar**, Associate Professor and thanks to all teaching and non teaching staff members those who contributed for the successful completion of our project work.

We wish to thank my family and many other people whose names are not mentioned here but this does not mean that we have forgotten their help.

With gratitude,

N.Pavithra (19K61A1238)

B.Sarika (19K61A1210)

P.Hemanth (19K61A1243)

B.Sai Kiran(19K61A1211)

INSTITUTE VISION AND MISSION

Institute Vision

Aspire to be a leading institute in professional education by creating technocrats to propel societal transformations through inventions and innovations

Institute Mission

1. To impart technology integrated active learning environment that nurtures the technical and life skills
2. To enhance scientific temper through active research leading to innovations and sustainable environment.
3. To create responsible citizens with highest ethical standards.

DEPARTMENT VISION AND MISSION

Department Vision

To become recognized centre for excellence for quality Information Technology education and create professionals with ability to solve social needs.

Department Mission

1. Provide quality teaching learning environment oriented towards employability and career development.
2. Conduct training/events for overall development of stakeholders with collaborations.
3. Impart value base education to serve the society with high integrity and good character.
4. Provide state of the art facilities to enable innovation, student centric learning.



PROGRAM OUTCOMES (POs)

Students in the Information Technology program should, at the time of their graduation, be in possession of :

PO1. Engineering knowledge : Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.

PO2. Problem analysis : Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.

PO3. Design/development of solutions : Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.

PO4. Conduct investigations of complex problems: Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to valid conclusions.

PO5. Modern tool usage : Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.

PO6. The engineer and society : Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.

PO7. Environment and sustainability : Understand the impact of the professional engineering solutions in societal and environmental contexts and demonstrate knowledge of and need for sustainable development.

PO8. Ethics : Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.

PO9. Individual and Team Work : Function effectively as an individual, and as a member or leader in diverse teams and in multidisciplinary settings.

PO10. Communication : Communicate effectively on complex engineering activities with the engineering community and with society at large, such as being able to comprehend and write effective reports and design documentation, make effective presentations and give and receive clear instructions.

PO11. Project Management and Finance : Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.

PO12. Life-long Learning : Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadcast context of technological change.

PROGRAM SPECIFIC OUTCOMES (PSOs)

PSO1. Application Development : Develop risk free innovative IT applications or industrial needs.

PSO2. Successful Career and Entrepreneurship : Explore technical knowledge in diverse areas of IT and experience an environment conducive in cultivating skills for successful career, entrepreneurship and higher studies.

PROGRAM EDUCATIONAL OBJECTIVES (PEOs)

PEO1. Possesses strong knowledge about Information Technology applications and leadership qualities.

PEO2. Pursue successful career in Information Technology and allied industries, innovate and provide solutions for global needs.

PEO3. Have attitude towards life-long learning attitude and practice professional ethics.

PROJECT OUTCOMES (PROs)

PRO1. Identifying the problems by doing a thorough literature survey of the existing research related to smart restaurant management system and generate solutions to the same with innovative ideas

PRO2. Analyze, design and develop a solution for the problems that arises when restaurant activities done manually.

PRO3. Develop employability and the ability to work in a team following the best ethical practices with a spirit for life-long learning and sharpening communication and presentation skills for smart restaurant management system..

PRO4. Make use of appropriate tools or techniques for sustainable development of a solution.

PRO5. Create interest to research different sources to perform restaurant activities through on-line.

PRO6. Construct a platform that makes users to interact with proposed system.

EXPECTED OUTCOMES

1. PROGRAM OUTCOMES(POs)

- **PO1** : Engineering knowledge
- **PO2** : Problem analysis
- **PO3** : Design/development of solutions
- **PO4** : Conduct investigations of complex problems
- **PO5** : Modern tool usage
- **PO6** : The engineer and society
- **PO7** : Environment and sustainability
- **PO8** : Ethics
- **PO9** : Individual and team work
- **PO10** : Communication
- **PO11** : Project management and finance
- **PO12** : Life-long learning

2. PROGRAM SPECIFIC OUTCOMES(PSOs)

- **PSO1** : Application Development
- **PSO2** : Successful Career and Entrepreneurship

SYNOPSIS OF THE PROJECT

Title of the project:

SMART RESTAURANT MANAGEMENT SYSTEM

Abstract

Nowadays technology is increasing rapidly in many areas, people are always craving for faster and easier day to day tasks. Improving the business techniques in food industry like restaurants and hotels will help to increase the customer satisfaction and also reduce the time taken by the user for various activities like waiting for table, menu, and also helps for manager and chef to perform their functions easily. Smart restaurant system application will reduce the manual errors which always takes place in manual restaurants like wrong food serving, billing etc. In smart restaurant system the details of restaurant was opened, here the customer can login to book the table, select the menu, and place the order without intervention of waiter. The manager can update the table status, menu status and other updates regarding restaurant manager panel.

Objectives of the study:

The aim of the project is to provide streamline operations by automating certain tasks, such as taking orders, processing payments, slot booking and table reservation which can reduce wait times of the customer. A smart restaurant management system can improve efficiency, reduce costs, and enhance the overall dining experience for customers.

The primary objective of this restaurant management system is to create an efficient system for booking and managing restaurants. Customers can access it with ease. The restaurant management initiative seeks to minimize workload and paperwork. Customer and manager wait times will be cut using this technology, and restaurant manual errors will also be decreased. And it enhances the customer experience with features such as mobile ordering and payment, personalized recommendations, and easy feedback collection, a smart restaurant management system can improve the overall dining experience for customers.

The rationale of the study:

- Existing system analysis
- Study of workflow and identify problems in the existing system.
- Requirements to the proposed system.

Detailed methodology used for carrying out the study:

The methodology used for carrying out a study of a restaurant management system would depend on the specific objectives and scope of the study.

- Agile methodology is a routine used for the development of the project which supports the respond to the volatility of building software through incremental,iterative work place.A mobile application is developed using agile techniques
- The first step is to define the scope of the project, which includes the features and functionalities of the smart restaurant management system. The scope of this proposed smart restaurant system is categorized into three categories that are guest-customer,staff,Restaurants owners.
 1. If it is admin he/she login the account and would involve managing and overseeing various administrative tasks and functions within the app to ensure smooth and efficient operations,Primarily he will give access to manager to update the restaurant activities.
 2. If it is manager he/she login the account the credentials which is given by the admin. And also responsible for analyzing data such as sales data, customer feedback, and staff performance data and managing the staff in the app, and also update restaurant activities.
 3. If it is user he/she login the account and typically involves using the app to interact with the restaurant and access its products and services such as browsing the menu,placing orders,making reservations,paying for order,providing feedback,accessing promotions and discounts

The expected contribution from the study:

- Flutter's widget-based architecture enables developers to create interactive user interfaces quickly and easily. The user interface of a smart restaurant management system should be designed to provide a user-friendly experience to customers, allowing them to easily reserve the tables, and order their food.
- A smart restaurant management system can contribute to the success of a restaurant in several ways, such as increased efficiency,improves customer satisfaction,Enhanced data analysis,Better inventory management,increased revenue and time savings

Examples of Smart Restaurants Management Applications are:

1. Easta
2. Inamo
3. Spyce
4. Big daddy's Dinner
5. Shake Shack

List of activities to be carried out to complete the project (with the help of bar chart showing the time schedule):

- Problem Identification
- Requirement Gathering
- Design
- Implementation
- Result Analysis
- Project Report Writing

Below bar graph represents the time allocated for the completion of project.

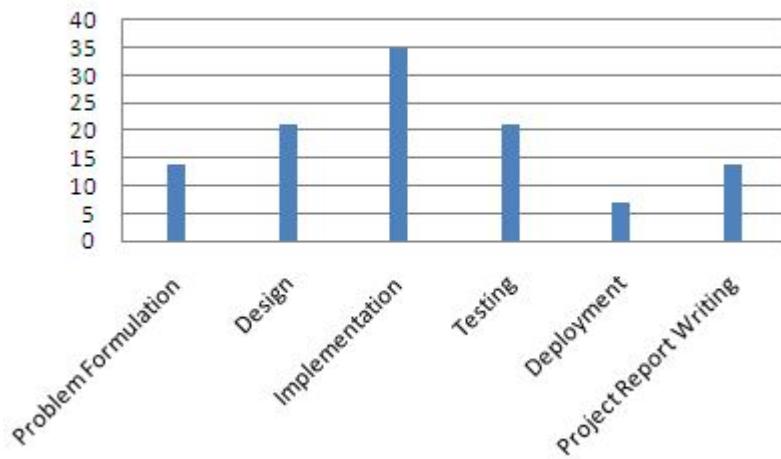
Places/Labs/Equipment and tools required and planning of arrangements:

HARDWARE REQUIREMENTS:

- Processor – Above i3
- Hard Disk – 40 GB
- Ram – 16GB RAM

SOFTWARE REQUIREMENTS:

- OPERATING SYSTEM : Windows 11
- Coding Language : Dart
- Back-end: NodeJs
- Front-end; Flutter
- IDE : Visual studio editor, Android studio

Problems envisaged in carrying out the project, if any :Nill**Time Schedule of the Project****Synopsis Prepared by****N.Pavithra**(19K61A1238)**Synopsis Guided by****Dr.Phani Kumar Solleti****B.Sarika**(19K61A1210)

Associate Professor

P.Hemanth(19K61A1243)

Department of IT

B.Sai Kiran(19K61A1211)

Signature:

Signature:

Signature:

Signature:

Signature:

Abstract

Now a days technology is increasing rapidly in many areas, people are always craving for faster and easier day to day tasks. Improving the business techniques in food industry like restaurants and hotels will help to increase the customer satisfaction and also reduce the time taken to the user for various activities like waiting for table, menu, and also helps for manager and chef to perform their functions easily. Smart restaurant system application will reduce the manual errors which always takes place in manual restaurants like wrong food serving, billing etc. In smart restaurant system the details of restaurant was opened, here the customer can login to book the table, select the menu, and place the order without intervention of waiter. The manger can update the table status, menu status and other updates regarding restaurant manager panel.

Contents

Acknowledgement	i
Vision and Mission	ii
Program Outcomes	iii
Synopsis of the Project	vi
Abstract	x
List of Tables	xiv
List of Figures	xv
1 INTRODUCTION	1
1.1 Description	1
1.2 Problem Formulation	2
1.3 Existing System	3
1.4 Motivation	4
1.5 Proposed System	4
1.6 Scope of the Project	4
1.6.1 Guest - Customer	4
1.6.2 Staff	4
1.6.3 Restaurants/Cafe/Stall owners	5
2 REVIEW OF LITERATURE	6
2.1 Reference Papers	6
2.2 Used Software Installations	14
2.2.1 Flutter Installations	14
2.2.2 VS Code	16
2.2.3 Dart	17
3 SYSTEM DESIGN AND ANALYSIS	20
3.1 System Architecture	20
3.2 UML Diagrams	21
3.2.1 Use Case Diagram	21
3.2.2 Sequence Diagrams	22

3.2.3 Class Diagram	25
3.2.4 Object Diagram	26
3.2.5 Activity Diagram	27
3.2.6 State Chart Diagram	28
3.2.7 Component Diagram	29
3.2.8 Deployment Diagram	30
3.3 Existing System	31
3.3.1 Disadvantages of Existing System:	31
3.4 Proposed System	32
3.4.1 Advantages of Proposed System:	32
3.5 Requirement Specification	32
3.5.1 Hardware Requirements:	32
3.5.2 Software Requirements	33
4 IMPLEMENTATION	34
4.1 Modules	34
4.1.1 Admin module	34
4.1.2 Manager Module	35
4.1.3 User Module	36
4.1.4 How the system works?	36
5 RESULTS	38
6 TESTING	49
6.1 Testing	49
6.2 Manual testing	50
6.2.1 White Box Testing	50
6.2.2 Black Box Testing	50
6.2.3 Gray Box Testing	50
6.3 Black Box Testing	51
6.3.1 Functional Testing	51
6.3.2 Non-Functional Testing	51
6.4 Functional Testing	51
6.4.1 Unit Testing	51
6.4.2 Integration Testing	52
6.4.3 System Testing	53
6.5 Non-Functional Testing	53
6.5.1 Performance Testing	53
6.5.2 Usability Testing	54
6.5.3 Compatibility Testing	54
6.6 Automation	54
6.7 Test Cases	54
7 CONCLUSION & FUTURE ENHANCEMENTS	56
BIBLIOGRAPHY	57

PRO vs PO RELEVANCE	59
APPENDIX 1	65
APPENDIX 2	74
APPENDIX 3	88
APPENDIX4	101

List of Tables

2.1 Literature Survey Table	13
6.1 Testcases	55
7.1 Project Outcomes	59
7.2 Summary of Project Outcomes mapping to Program Outcomes Project Phase - 1 . .	59
7.3 Summary of Project Outcomes mapping to Program Outcomes Project Phase - 2 . .	59
7.4 PRO Relevance to PO	63
7.5 PRO Relevance to PSO	64

List of Figures

2.1 Flutter Installation	15
2.2 Setup Progress for flutter	15
2.3 Setup was Successful for flutter	16
2.4 Setup was Successful for VS code	17
2.5 Download Dart SDK	18
2.6 Extract zip file	18
2.7 Setup was Successful for dart	19
3.1 Architecture Diagram for restaurant management system	20
3.2 Use Case Diagram for restaurant management system	21
3.3 Sequence diagram for Login	22
3.4 Sequence diagram for creating a manager	23
3.5 Sequence diagram for restaurant selection	23
3.6 Sequence diagram for reservation of tables	24
3.7 Sequence diagram for payment	24
3.8 Sequence diagram for ordering food	25
3.9 Sequence diagram for update restaurant status	25
3.10 Class Diagram for restaurant management system	26
3.11 Object Diagram for restaurant management system	27
3.12 Activity Diagram for restaurant management system	28
3.13 State Chart Diagram for restaurant management system	29
3.14 Component Diagram for restaurant management system	30
3.15 Deployment Diagram for restaurant management system	31
5.1 Login admin page	38
5.2 Adding admin account	39
5.3 Adding manager	39
5.4 Adding Manager and available managers	39
5.5 Sharing manager id	40
5.6 Admin details and logout page	40
5.7 Login page of manager	40
5.8 Welcome page of manager	41
5.9 Adding categories	41
5.10 Adding food items	42

5.11 Adding Tables	42
5.12 Managing orders	43
5.13 Manager details Logout from the Application	43
5.14 Restaurants available	43
5.15 Selecting one of the Restaurant available	44
5.16 Available tables in restaurant	44
5.17 Table Booking	44
5.18 Menu page	45
5.19 Added items	45
5.20 Table status of the customer	45
5.21 Items added in the cart	46
5.22 Payment options	46
5.23 Order and payment details	46
5.24 Review option	47
5.25 Online payment	47
5.26 review of the user	47
5.27 Logout from the Application	48

Chapter 1

INTRODUCTION

A Smart Restaurant system is a system proposed to develop the food industry by removing some disadvantages that occur in the traditional systems. The system developed will help the people and world in saving time and energy in finding best restaurants in his desire locations and reduce the time taken in restaurants by making the process automatic.

1.1 Description

The methods that are followed by restaurants nowadays are pen and paper based methods. In this method, firstly they will print a menu of the restaurant using paper that is paper-based menu card. Every time they have to print new menu cards if they want to do any changes to the menu, like based the food available in the restaurants, offering some discounts and special offers or adding some new items and even if the menu card is printed wrongly, there is no chance to change that card, instead of printing a new card. It is also a time waste for waiters to wait until the customers, select the desired food. Taking orders using pen and paper is another process used in the traditional system. Taking orders using paper and sending them to the kitchen is results in a lot of time waste. Sometimes the handwriting of the waiter is not understood by the kitchen department. So in this case preparing wrong food and sending them to customer's results in a lot of customer dissatisfaction. In some seasons when the customer visiting hotels or restaurants will also lead to customer dissatisfaction as the waiters cannot able to provide immediate service to all the customers, giving them the wrong orders and sometimes it takes a lot of delays to deliver food even after customer waited a long time. Sometimes the papers in which the orders are taken are also lost in a lot of order receipts. Hence a lot of complaints are raised about the services provided in the hotel at the management desk. After the food is consumed food by customers, the bill receipt will send is also through paper. Employing more waiters to satisfy customers in crowd situations will also be a drawback as there is no need for more waiters in low times and leads to loss of money in form of salary. Even it results in a lot of paper waste for taking orders, printing

the menu cards,making bill receipts.

Hence the current restaurant system results in a lot of inconvenience to both the customers and management through the many manual errors performed. Hence a lot of researches are done, to improve this existing system and to help the management to increase the service, managing the restaurants and the customers are also need not wait for waiters to take orders and it also reduces the time for both customers and restaurant staff and improves customer satisfaction.

People are rapidly moving towards a smarter world, with the implementation of smart cities, smart classrooms, and smartphones. At present, information and communication technology has been brought to several business models to make the operation more convenient and effective. In almost every single nation and culture in the world today, restaurants hold an important role as a hub for social interactions. Not only do they provide food, drinks, and sustenance to individuals, but they also function as a place where groups of people come together to socialize, connect, and share great experiences. Restaurants are also extremely lucrative businesses, as the global restaurant industry grew by 6.2 percent in 2014 to reach a value of 2,737.1 dollar billion, and it is forecasted to grow to a value of 3,805.8 dollar billion by 2019. However, despite restaurants holding such an important role in our society and despite them earning high revenues, the restaurant industry has had few changes to how it has been operating in the past century. Despite all of the great technological advancements that have been made in recent decades with the rise of the internet, mobile smart devices, and cloud computing, the restaurant industry still has not fully exploited the full capabilities of these technologies and still has a lot of inefficiencies that are unaddressed.

1.2 Problem Formulation

As the restaurant systems around the world are following the traditional restaurant system that contains paper-based menu cards and uses pen and paper to take the orders from customers, and the orders are written are taken to the kitchen for food preparation. Through manual ordering there is a chance for a lot of mistakes, sometimes when there is a huge crowd in the restaurant the waiter will take order firstly, so in that case, the handwriting of waiter will not understand by kitchen department results in the preparation of food other than customer order will result in customer dissatisfaction, results in a lot of complaints raised by the customers about the services,receiving wrong order or even not receiving the order even after waiting for a long time. Sometimes the papers in which the order is also lost. Even it results in a lot of paper waste for taking orders, printing the menu cards, making bill receipts. In case if a menu card is printed wrongly or to change the menu available in the restaurant and even to change the prices of the menu it is not possible to change it easily and we have to print another menu card which results in the paper and money waste. This system is also a waste of time and energy of taking the orders

waiting for the orders and even waiting for tables to sit and also the customers are not aware of the time taken to prepare the food. Hence the existing system results in a lot of inconvenience to both the customers and management. Hence a system is proposed that will help the management to increase the services and management and the customers cannot wait for waiters to take orders and it also reduces the time for both customers and restaurant staff and improves customer satisfaction.

To solve those problems a smart restaurant system was implemented. Using this system customer satisfaction and restaurant productivity were increased. It saves time and energy for both customers and restaurant staff also. It also reduces pen and paper waste. It reduces the manual errors and waiters required in restaurants.

1.3 Existing System

Nowadays, various services become available on smartphones. In this system, the customer will find the restaurants in his desired locations with the rating given by the other customers. After selecting the restaurant he will come know the information about the availability of tables in the desired restaurants prior. After entering the restaurant, customers can able to see the menu available in the restaurant along with their cost. They can browse the variety of dishes available in the list and they should add the required item into the cart with the desired quantity. They can also add or delete the items to or from the cart respectively. Along with the items, the cost of the items is also visible and the total cost of items in the cart also appears at the end. After adding all the required items to the cart he should have to do the bill payment online or in the form of cash and then click on the proceed button to place an order. The manager can mange the restaurant updates and transactiondetails. The data will store in the database about the items ordered and it helps to know which item is ordered more, which helps to manage the budgeting and planning according to it. As the restaurant systems around the world are following the traditional restaurant system that contains paper-based menu cards and uses pen and paper to take the orders from customers, and the orders are written are taken to the kitchen for food preparation. Through manual ordering there is a chance for a lot of mistakes, sometimes when there is a huge crowd in the restaurant the waiter will take order fastly, so in that case, the handwriting of waiter will not understand by kitchen department results in the preparation of food other than customer order will result in customer dissatisfaction, results in a lot of complaints raised by the customers about the services, receiving wrong order or even not receiving the order even after waiting for a long time. Sometimes the papers in which the order is also lost. Even it results in a lot of paper waste for taking orders, printing the menu cards, making bill receipts. In case if a menu card is printed wrongly or to change the menu available in the restaurant and even to change the prices of the menu it is not possible to change it easily and we have to print another menu card

which results in the paper and money waste. This system is also a waste of time and energy of taking the orders waiting for the orders and even waiting for tables to sit and also the customers are not aware of the time taken to prepare the food. Hence the existing system results in a lot of inconvenience to both the customers and management. Hence a system is proposed that will help the management to increase the services and management and the customers cannot wait for waiters to take orders and it also reduces the time for both customers and restaurant staff and improves customer satisfaction.

1.4 Motivation

The present Restaurant systems follows a manual process for every aspect that will done in restaurant which consumes lot of time and tests the customer patience and results in lot of dissatisfaction of the customer, which effects the restaurant revenue. Hence the smart restaurant system application will reduce the time taken in manual restaurants and saves the people time and helps in increasing revenue of restaurants.

1.5 Proposed System

Nowadays, wireless communication has been extended such as 3G, 4G, and 5G for efficient and effective communication between the users. This was developed based on the wireless channel bandwidth for the data transmission between M and E-commerce. The proposed smart restaurant system will help the customers to have the best dining experience and staff.

1.6 Scope of the Project

The scope of this proposed smart restaurant system is categorized into three categories:

1.6.1 Guest - Customer

1. Customer will find the restaurant in his location with ratings.
2. Customers can able to check the availability of food in the restaurant.
3. Customers will use the application to see the menu and to order the menu they want along with the cost by selecting the quantity.

1.6.2 Staff

1. Staff will serve the food to the customers which they order based on the table number.

2. Staff will also be able to know whether the guests order the food or not.

1.6.3 Restaurants/Cafe/Stall owners

1. Admin can able to add a new menu or can delete the menu if not available and updating the food menu with changed prices.
2. Admin can view the report of data

Chapter 2

REVIEW OF LITERATURE

2.1 Reference Papers

Liyanage, Ekanayake, Premasiri, *et al.* 2018 [1] proposed a “Foody” system to address the deficiencies and provide efficient and accurate services to the customer, by providing unique menus to each customer considering their taste. This concept is implemented as a mobile application using latest IT concepts such as Business Intelligence, Data Mining, Predictive Analysis and Artificial Intelligence. This includes graphics and 3D modeling that provide existent physical information related to food such as colors, sizes and further user can view the ingredients of the meal as well as the available tables. In addition, the app shows the real-time map to the restaurant. Current table reservation status is indicated by the color change of the table. Unique food recommendation and it's order for each customer is generated by analyzing their social media information and the system notifies the customer the wait time by calculating it. Preparation of food and allocation is done subjectively.

Koubai and Bouyakoub 2019 [2] proposed a methodology for a smart restaurant with a recommendation system using IOT, which aims to connect different objects in restaurants like oven, refrigerator, and people in the restaurant. In this the customer can have to register to the page which shows him the nearest restaurant using his GPS and he can also able to book a table in advance after entering into restaurant a customer can order through connected tables which displays menu electronically on the table, after entering into page by uploading preference the recommendation will shown to the customer. It also provide games for customers playing when waiting for order and they can also keep track of their food. The refrigerator also connected to it which indicates about the details of the food it contains and the oven is also digitally connected which automatically heats the food based on the chef instruction depend on chef system using “WAMP server”, for web application based on windows APACHE, MYSQL, PHP, android application

is developed by android mobile app, Eclipse, JAVA, XML, PHP for database maintenance.

Shedole [3] proposed a system that covers the whole order process of a restaurant includes the interaction between customer, the waiter, the kitchen and the cashier through android application and desktop based software. The system will also include a database maintaining the record of employees. Digital Smart Menu could be replacement to paper-based menu. We conducted an survey concerning perception of paperbased menus and their expectation to digital menus. The digital menu will provide interactive user interface with which user will easily place his order by itself.

Joshi, Dhawale, Dhekane, *et al.* [4] proposed a automated restaurant service management system, In this once the order is placed and confirmed with their respective table number, then the order will be delivered by the robots to their respective tables. This application was designed using Android Studio, XML, User Interface. The customers can download the app in their devices which consist of menu of the restaurant and after selecting the menu and placing order, Final bill can be displayed on app. UI was made in the XML part and the backend done in the java classes. Databases were used for adding of data automatically. This will ensure more work profit and digitilized environment and would also cut down human error.

Cheng, Chang, and Chen 2021 [5] proposed a smart restaurant: from order to delivery and security system to automate the process in the restaurant. In this system customers can use their smartphones to order food from a digital menu available in the website. Orders are displayed to chef in the kitchen where he can prepare the food according to the queue. After the preparation a waiter robot will deliver the food to the respective table. Using a digital menu to take an order and a robot to serve the food reduces human error. It also helps in reducing the employability costs and increases profit margin. Some security measures are also implemented to prevent theft and an alert system in case of an emergency.

Bolaños, Valdivia, and Radeva 2019 [6] proposed a semantic food detection for smart restaurant, to address the problem of automatic food through analysis in canteens and restaurants environment, which consists in predicting multiple foods placed on a tray image. In this they proposed a new approach for food analysis based on convolutional neural networks, and named it as Semantic Food Detection, which integrates in the same framework food localization, recognition and segmentation. We demonstrate that our method improves the state-of-art food detection by a considerable margin on the public dataset UNIMIB2016, achieving about 90 percentage in terms of F-measure, and thus provides a significant technological advance towards the automatic billing in restaurant environments.

Nikhitha Deshpande 2018 [7] proposed a system using Ardunio mega, Wire Resistive touch screen, TFT Display RFID card, to automate the food ordering and billing process in restaurant as well as to improve the dining experience of customers. In this the order will automatically

displayed to the kitchen department through rfid flags.

Literature Review

Liyanage, Ekanayake, Premasiri, *et al.* 2018 [1] proposed a semantic food detection for smart restaurant, to address the problem of automatic food through analysis in canteens and restaurants environment, which consists in predicting multiple foods placed on a tray image. In this they proposed a new approach for food analysis based on convolutional neural networks, and named it as Semantic Food Detection, which integrates in the same framework food localization, recognition and segmentation. We demonstrate that our method improves the state-of-art food detection by a considerable margin on the public dataset UNIMIB2016, achieving about 90 percentage in terms of F-measure, and thus provides a significant technological advance towards the automatic billing in restaurant environments.

Nikhitha Deshpande 2018 [7] proposed a system using Ardunio mega, Wire Resistive touch screen, TFT Display RFID card, to automate the food ordering and billing process in restaurant as well as to improve the dining experience of customers. In this the order will automatically displayed to the kitchen department through rfid flags.

Harpanahalli, Bhingradia, Jain, *et al.* 2020 [8] proposed a smart Restaurant system using RFID technology. The proposed system contained two sub systems, one is master another is slave system. The master system is implemented for handling the processing, GUI and transactions. The master contains the Raspberry pi. An Arduino UNO is connected to slave system which connected to RFID readers. Using several peripheral interface, the connection between the reader and the Aurdino was takes placed. Passive RFID tags are used to identify the food items placed on the trey. The master system will start it working by checking it connection with database, arduino and GUI boots up and waited for button t0 pressed. If the scanned data from the database that displayed by GUI and total amount is calculated. By pressing the key button, the money is debited from users wallet. The slave system work was started by scanning the RFID tags, which are send to Raspberry pi. If the arduino is not connected or the money is not debited an error message is displayed on the screened. The items which are placed on tray will also cross verified by the arduino sub system. The advantage of the proposed system helps to reduce the time taken for scanning and payment and removed the bottle necks in high tomes in restaurants.

Qamar, Tariq, Aqeel, *et al.* [9] proposed a SMARTABLE - Digital Ordering System for Restaurant based on Android applications and Web application. There are separate applications for all the stakeholders of the restaurant i.e. diner, waiter, chef and administrator. The system is

developed after observing several limitations with traditional restaurants, by conducting a survey from 100+ participants. The features of SMARTABLE are picked after analyzing the survey responses, hence assuring an efficient solution for restaurant automation. SMARTABLE comprises of four android applications according to the nature of users of restaurants and a fully functional web based admin panel that is capable of not only controlling the management activities but also provide trends via data mining algorithms.

A study on IoT application in the Food Industry using Keywords was proposed by Bigliardi, Bottani, and Filippelli 2022 [10]. They proposed this system to identify and organize the main research topics regarding the development of IoT systems in the food sector, through a keyword analysis. This methodology adopted allows to identify the concepts that persist over time, those that emerge with high frequency and those that, instead, have disappeared over the years. In this the results allow us to make predictions about topics of future interest to researchers, outlining avenues for promising research.

Data Centric Smart Restaurant Management System was developed by Ravekar and Singh 2019 [11]. to create a system that will enable any restaurant to serve as a data-centric service provider for its customers and also automate the process of ordering the food items. Every customer who enters the restaurant needs to have the restaurant's app installed on their phone which can be downloaded by scanning a QR code in the restaurant. The user needs to sign up or sign in to the app and connect to the restaurant Wi-Fi. The entire menu will be available on the app of the user from which the users can select the dishes they'd like to order. The ordered food items will be displayed to the kitchen staff over on a kitchen screen which can also serve as the local server (host). The food ordered by each customer can be tracked through live status along with a live video feed of the kitchen so that the users can get real-time updates about their orders. Also, each order will be recorded per customer which will help the restaurant owner to directly predict the kind of food each user might order and also provide similar recommendations to the customers based on their tastes and choices.

Gregorash 2016 [12] proposed a system called Restaurant revenue management: apply reservation management. The aim of this system will reduce major complaints from customers arise from errors in booking the tables when they are not available even after they booked the table. It will reduce major complaints from customers arise by the errors in booking. The presents data collected from the restaurant, the number of reservation guests, the number of reservation guests and the mean in dollars from each.

Umap, Surode, Kshirsagar, *et al.* 2018 [13] proposed a fully automated menu ordering sys-

tem in which the paper based menu is replaced by a user friendly Matrix keypad based menu card. The system has PIC microcontroller which is interfaced with the input and output modules. The input module is the matrix keypad sensor which is placed on LCD (Graphical Liquid Crystal Display) to have a graphic image display, which takes the input from the user and provides the same information to the microcontroller. The output module is a NRF module which is used for communication between system at the table and system for receiving section. Microcontroller also displays the menu items on the LCD. At the receiving end the selected items will be displayed on the LCD and by using the conveyer belt the received order will send to the particular table.

Qamar, Tariq, Aqeel, *et al.* [14] proposed a SMARTABLE-Digital Ordering System for Restaurant Using Android . Smart restaurant is a concept where a restaurant gets automated, using state of the art technology. Nowadays people are more believing on technologies as they aim to satisfy their needs more comprehensibly. Most of the restaurant industries are interested for any application which improves dining experience and boost profit as well. In traditional way of ordering system the waiter take an order from diner then places it to the kitchen and then the manual billing is performed. These all consume much time and cause the chances of errors. The purpose of this research is to give a solution in terms of digital food ordering where whole order process becomes automated. This paper proposes SMARTABLE; a digital ordering system for restaurants based on Android applications and Web application. There are separate applications for all the stakeholders of the restaurant i.e. diner, waiter, chef and administrator. The system is developed after observing several limitations with traditional restaurants, by conducting a survey from 100+ participants. The features of SMARTABLE are picked after analyzing the survey responses, hence assuring an efficient solution for restaurant automation. SMARTABLE comprises of four android applications according to the nature of users of restaurants and a fully functional web based admin panel that is capable of not only controlling the management activities but also provide trends via data mining algorithms.

Priyanka, Prathyusha, Tejeswini, *et al.* 2020 [15] proposed a Smart Food Quality Testing and Ordering System Using at Mega328 in Restaurants. This system designed using AT Mega328, Passive infrared (PIR) sensor, MQ3 sensor, Liquid Crystal Display(LCD), Bluetooth HC-05 and keypad. The PIR sensor gives customer availability at a particular table.The MQ3 sensor detects food quality. The customer order the required items by the keypad. The LCDdisplaysthe available items and cost of items. The Bluetooth HC-05 module will send messages or information about the ordered items to the chef.

Saeed, Shouman, Elfar, *et al.* 2016 [16] proposed a Near-field communication sensors and cloud-based smart restaurant management system is a web application and an android version to

restaurant management. It will have all the features of the rapid involving science and its different attributes. Through a strategic design and customer orientation, HARMS is integrated and has been created to optimize the work force and streamline restaurant work flow. It will run on an android, and is both scalable and modular to meet the needs of any establishment. HARMS is an effort to bring technology into the dining menu of customers. HARMS offer robust features that not only help your restaurant to update the menu any time but also improve the overall dining experience. The digital menu is to provide a user-friendly interface by offering smooth navigation and browsing through digital menu thus providing a delightful experience. The customers can order the food, through that android interface.

Kapoor and Vij 2018 [17] proposed a system called "Technology at the dinner table: Ordering food online through mobile apps" to investigate online food aggregators (OFA's) by proposing and empirically testing mobile app attribute-conversion model, to examine how mobile app attributes of online food aggregators influence the purchase decision of a consumer and subsequently lead to conversion. A mix method design was adopted for the study and a pilot study comprising of (n=350) respondents was carried out. The study focuses on four key attributes – visual, navigational, information and collaboration design and identifies the most important mobile app attributes while choosing a particular online food aggregator (OFA) in India.

Jakhete and Mankar 2015 [18] proposed a Implementation of Smart Restaurant with e-menu Card. They proposed this system to automate day to day activity of a restaurant. Restaurant is a kind of business that serves people all over world with ready-made food. This system is developed to provide service facility to restaurant and also to the customer. A Restaurant Management System (RMS) is a type of Point-of-Sale (POS) software specifically designed for restaurants, bars, food trucks and others in the food service industry. Unlike a POS system and RMS encompasses all back-end needs, such as inventory to staff management. Also e-Restaurant is a restaurant reservation system designed to replace the pen paper approach to reservation management. The software has been build from the ground-up to be as easy to use as possible as well as extremely powerful and highly customisable to suit any restaurant environment.

Tarjan, Šenk, Tegeltija, *et al.* 2014 [19] proposed a QR CODE BASED SMART DINING SYSTEM to improve the customer dining experience which is developed by using JavaScript, PHP, Android Studio, My SQL. In this QR codes are placed on each table of the restaurant, so the customer can able to scan the QR codes using their smart phone and they will redirect to the restaurant web page, where the menu list are available on the page. So the customer can order the food by selecting the desired type and quantity and clicking send order. That will send order to management desk which will intimates to kitchen department and after food preparation a robot

will send food to the customer, Admin will login to admin page to contact with customers as well as with kitchen department and also to manage the database [33] using PHP scripting.

Bhargave, Jadhav, Joshi, *et al.* 2013 [20] proposed a Digital Restaurant .They developed this system to implement where order will be taken from an application installed in customers' Android device and the order will be displayed in kitchen. This all is to reduce human labor low power consumption, no need of person to take order. As we have seen in many restaurant such as KFC, Dominos or McDonalds but it is partially implemented. The menu will be displayed on table with the help of Android system or device. Customer can place the order through the device which will be of their choice. The chef in the kitchen will place order on the conveyor belt then customer can pick up their order from the conveyor belt, if suppose customer needs anything they can call the waiter to help them. The payment can be made online with the help of card swipers and to complete their transactions. Entire project has been developed keeping in view of the e-service computing technology.

S.No	Author Name and Year	Methodology	Advantages	Disadvantages
1	Liyanage, Ekanayake et al 2019	Current table reservation status is indicated by the color change of the table.	Reduce the time taken to order the food..	Food wastage in case the customer left the restaurant after ordering food..
2	Koubai and Bouyakoub et al. 2019	using the GPS and we can also able to book a table	Improves efficiency and save time and money	Dependence on technology.
3	Shedole et al. 2022	The digital menu will provide interactive user interface with which user will easily place his order by itself	Improves customer experience and satisfaction.	Food wastage in case the customer left the restaurant after ordering food.
4	Joshi, Dhawale et al. 2022	This will ensure more work profit and digitized environment and would also cut down human error.	Reducing the employability costs	Reduced human interaction
5	Cheng, Chang et al. 2022	security measures are also implemented to prevent theft and an alert system in case of an emergency.	Improved customer experience.	Dependence on technology
6	Dependence on technology 2022	It proposed a new approach for food analysis based on convolutional neural networks. .	Able to know quality of food	Stops the entire process in case of failure of the technology
7	Nikhitha Deshpande et al. 2020	order will automatically displayed to the kitchen department through rfid flags..	Reduces paper waste	Stops the entire process in case of failure of the technology.
8	Harpanahalli jain et al. 2019	Radio Frequency identification technology based smart restaurant system using open technologies like python and Raspberry pi	reduce the bottlenecks caused for cashiers.	Reduces customer experience and satisfaction
9	Qamar, Tariq et al. 2020	capable of not only controlling the management activities but also provide trends via data mining algorithms..	Improves customer experience and satisfaction.	Stops the entire process in case of failure of the technology
10	Bottani, and Filippelli et al. 2022	.They proposed this system to identify topics regarding the development of IoT systems in the food sector, through a keyword analysis.	Reducing the employability costs	time is taken for delivering food will not be known by customers.
11	Ravekar and Singh et al. 2020	It also provide similar recommendations to the customers based on their tastes and choices.	This can help to boost sales and profitability.	Decreased data privacy concerns.
12	Gregorash et al. 2020	It will reduce major complaints from customers arise by the errors in booking..	Improved efficient and revenue.	Stops the entire process in case of failure of the technology
13	Umap, Surode et al. 2018	The output module is a NRF module which is used for communication between system at the table and system for receiving section	Better data management.	Decreased human interaction
14	Qamar, Tariq, Aqeel et al. 2022	Smart restaurant is a concept where a restaurant gets automated, using state of the art technology.	Get food recommendations.	Food wastage in case the customer left the restaurant after ordering food..
15	Priyanka, et al. 2022	The Bluetooth HC-05 module will send messages or information about the ordered items to the chef.	Improved efficient and revenue	Decreased data privacy concerns.
16	Saeed, Shouman et al. 2021	The customers can order the food, through that android interface	This can help to boost sales and profitability.	Stops the entire process in case of failure of the technology.
17	Kapoor et al. 2020	The study focuses on four key attributes – visual, navigational, information and collaboration design.	Better data management	Decreased human interaction.
18	Jakhete et al. 2020	Unlike a POS system and RMS encompasses all back-end needs, such as inventory to staff management..	Improved efficient and revenue	Food wastage in case the customer left the restaurant after ordering food..
19	Tarjan, Senk et al. 2020	QR codes are placed on each table of the restaurant, so the customer can able to scan the QR codes using their smart phone	Improved efficient and revenue.	Reduced personal interaction.
20	Bhargaveet al. 2020	Entire project has been developed keeping in view of the e-service computing technology.	Get better data management.	Decreased data privacy concerns.

Table 2.1: Literature Survey Table

2.2 Used Software Installations

2.2.1 Flutter Installations

Flutter is basically Google's portable user interface (UI) toolkit, used to build and develop eye-catching, natively-built applications for mobile, desktop, and web, from a single codebase. Flutter is free, open-sourced, and compatible with existing code. It is utilized by companies and developers around the world, due to its user-friendly interface and fairly simple, yet to-the-point commands.

Installing Flutter on Windows:

Step 1: Flutter Download

Navigate to flutter.dev on your webpage. On the top menu bar, select Docs → Get Started → Install → Windows.

Step 2: Download flutter

Restart the system after installing Git on your windows. Once done, let's get to the installation of Flutter Software development Kit (Flutter SDK). Click on the download link for the latest version (as of today).

Step 3: Install Flutter

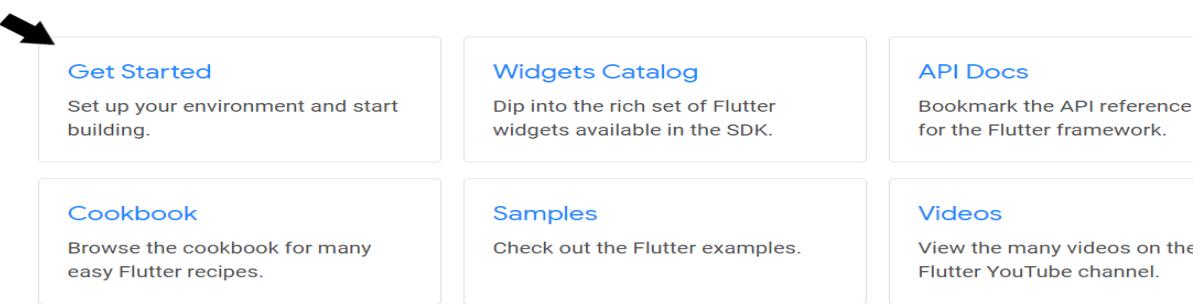
Now, you have to analyze and check whether something is missing/has to be installed further. To do this, under the Command Prompt terminal, type in 'Flutter Doctor' to check for other requirements.

set sdk as environmental variable and give permissions to android licesnce and click on android device.

[Docs](#)
[Showcase](#)
[Comm](#)

Get started with Flutter 2.2. See [What's new in docs](#), including a list of the new instructor-led video

Flutter documentation



What's new on this site

To see changes to the site since our last release, see [What's new](#).

Figure 2.1: Flutter Installation

Run `flutter doctor`

From a console window that has the Flutter directory in the path (see above), run the following command to see if there are any platform dependencies you need to complete the setup:

```
C:\src\flutter>flutter doctor
```



This command checks your environment and displays a report of the status of your Flutter installation. Check the output carefully for other software you might need to install or further tasks to perform (shown in **bold** text).

For example:

```
[+] Android toolchain - develop for Android devices
  • Android SDK at D:\Android\sdk
  X Android SDK is missing command line tools; download from https://goo.gl/XxQghQ
  • Try re-installing or updating your Android SDK,
    visit https://flutter.dev/setup/#android-setup for detailed instructions.
```



The following sections describe how to perform these tasks and finish the setup process. Once you have installed any missing dependencies, you can run the `flutter doctor` command again to verify that you've set everything up correctly.

Figure 2.2: Setup Progress for flutter

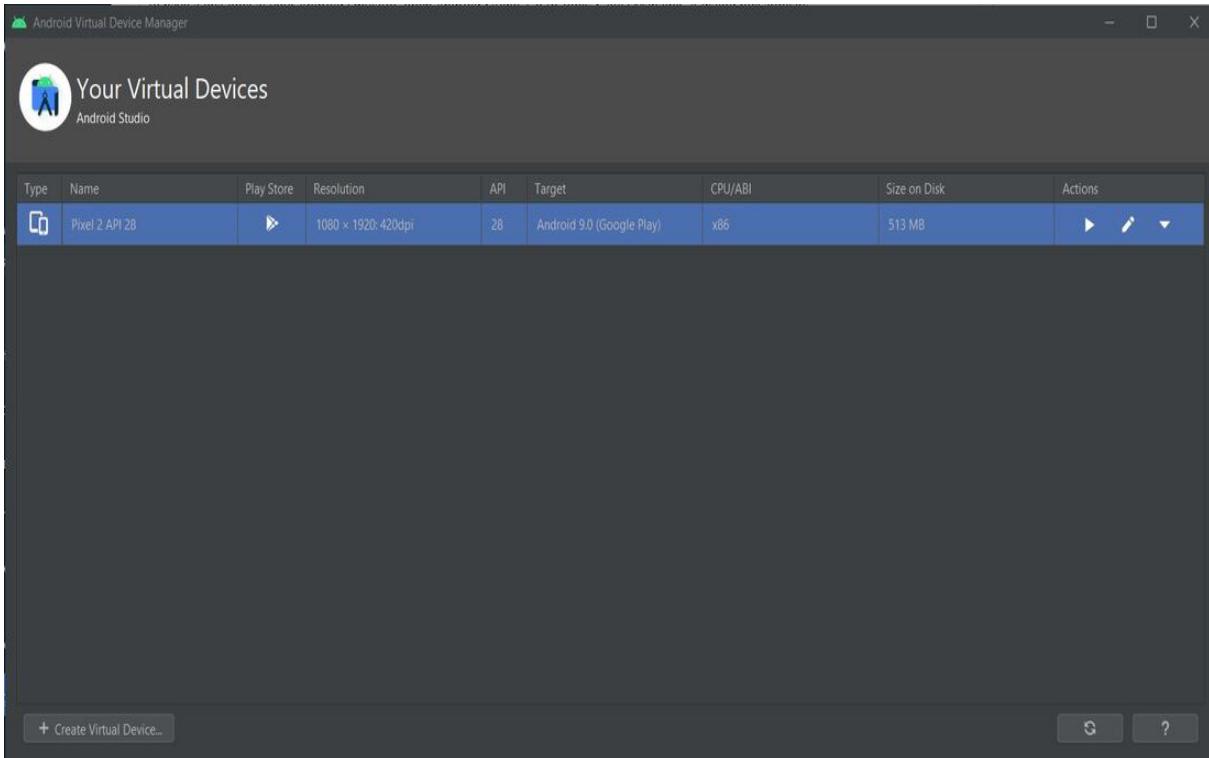


Figure 2.3: Setup was Successful for flutter

Finally, Flutter has successfully installed.

2.2.2 VS Code

Visual Studio Code is a lightweight but powerful source code editor which runs on your desktop and is available for Windows, macOS and Linux. It comes with built-in support for JavaScript, TypeScript and Node.js and has a rich ecosystem of extensions for other languages (such as C++, Java, Python, PHP, Go) and runtimes (such as .NET and Unity).

1. Download the Visual Studio Code installer for Windows.
2. Once it is downloaded, run the installer (VSCodeUserSetup-version.exe). This will only take a minute.
3. By default, VS Code is installed under C:VS Code.

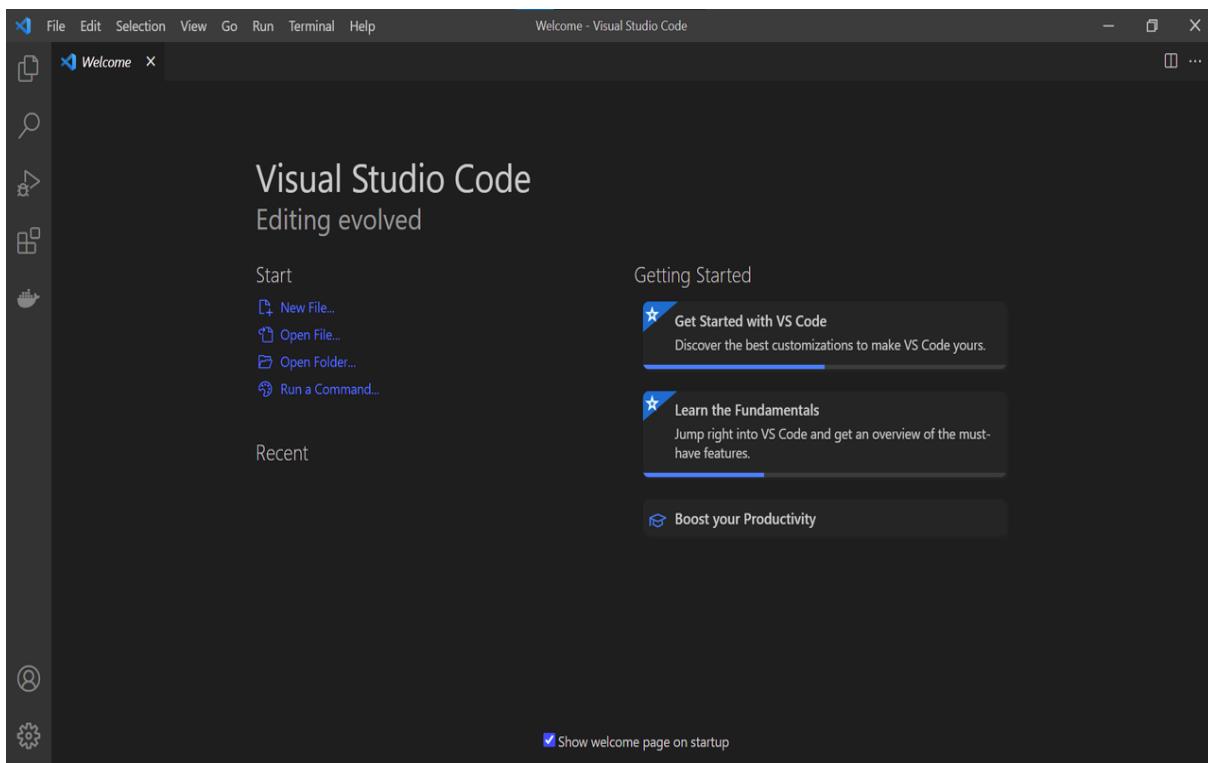


Figure 2.4: Setup was Successful for VS code

2.2.3 Dart

Dart is a client-optimized language for developing fast apps on any platform. Its goal is to offer the most productive programming language for multi-platform development, paired with a flexible execution runtime platform for app frameworks.

Step 1: Download Dart SDK

Go to Dart SDK archive page.

The URL is <https://dart.dev/tools/sdk/archive>.

Click on the Dart SDK link. The download will start immediately and a zip file will be downloaded.

Step 2: Extract zip file

Extract the contents of Dart SDK zip file. The contents of the folder would be as shown below.

Step 3: Run Dart

- You can run Dart command.

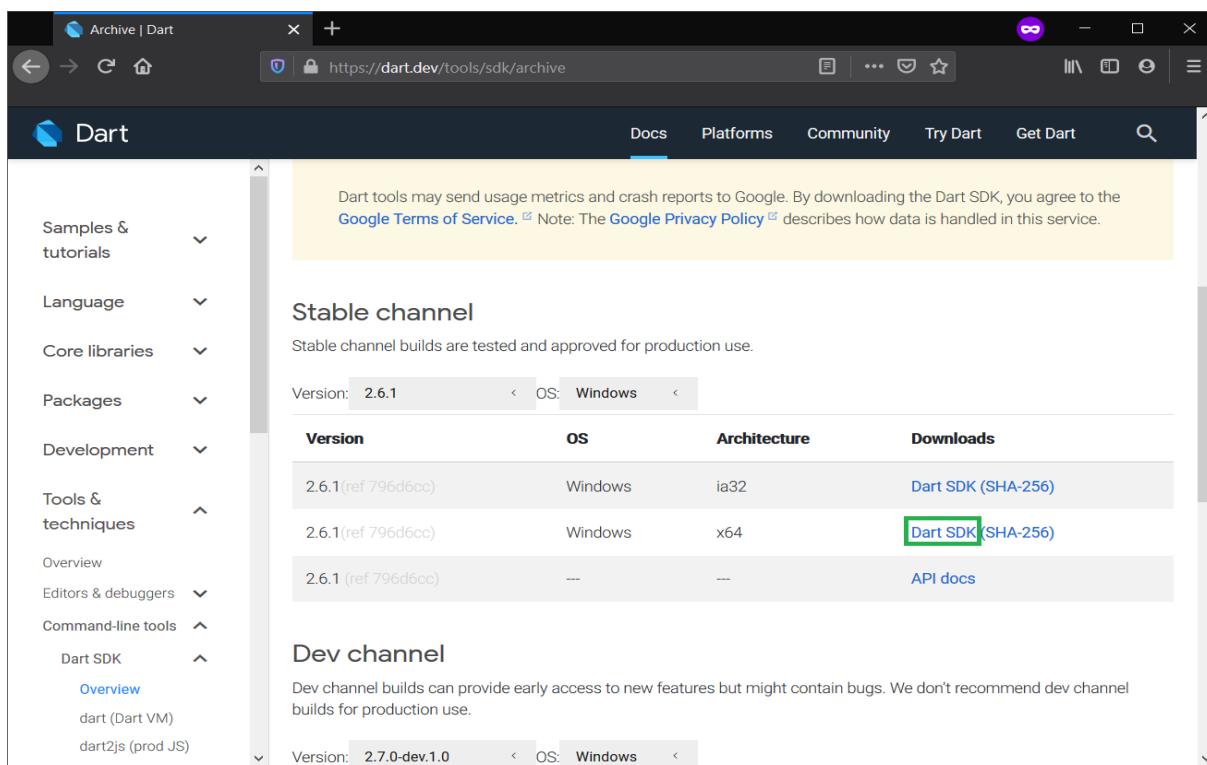


Figure 2.5: Download Dart SDK

The screenshot shows a Windows File Explorer window with the path 'This PC > DATA (D:) > software > dart-sdk'. The table below lists the contents of the 'dart-sdk' folder.

Name	Date modified	Type	Size
bin	11-11-2019 18:01	File folder	
include	11-11-2019 17:55	File folder	
lib	11-11-2019 17:55	File folder	
dartdoc_options.yaml	11-11-2019 17:55	YAML File	1 KB
LICENSE	11-11-2019 17:48	File	2 KB
README	11-11-2019 17:48	File	1 KB
revision	11-11-2019 17:55	File	1 KB
version	11-11-2019 17:55	File	1 KB

Figure 2.6: Extract zip file

- Open command prompt and navigate to the bin folder.
- Run the command dart.

Step 4: Add Dart Path to PATH Environment Variable

As of now, you can run dart command only from bin folder of dart sdk. To run dart command from anywhere in your file system, add dart bin path to PATH environment variable. Open Environment Variables. Under System variables, click on Path and click Edit button. Edit environment variable window appears. Click on New and paste the dart sdk bin path as shown below.

Step 5: Restart Command Prompt Close the existing command prompt window and open a new command prompt. Just run the dart command from any working directory. We shall run from D:

```
D:\>dart
Usage: dart [<vm-flags>] <dart-script-file> [<script-arguments>]
Executes the Dart script <dart-script-file> with the given list of <script-arguments>.

Common VM flags:
--enable-asserts
  Enable assert statements.
--help or -h
  Display this message (add -v or --verbose for information about all VM options).
--package-root=<path> or -p<path>
  Where to find packages, that is, "package:...." imports.
--packages=<path>
  Where to find a package spec file.
--observe[=<port>[/<bind-address>]]
  The observe flag is a convenience flag used to run a program with a set of options which are often useful for debugging under Observatory. These options are currently:
    --enable-vm-service[=<port>/<bind-address>]
    --pause-isolates-on-exit
    --pause-isolates-on-unhandled-exceptions
    --warn-on-pause-with-no-debugger
  This set is subject to change.
  Please see these options (--help --verbose) for further documentation.
--write-service-info=<file_name>
  Outputs information necessary to connect to the VM service to the specified file in JSON format. Useful for clients which are unable to listen to stdout for the Observatory listening message.
--snapshot-kind=<snapshot_kind>
--snapshot=<file_name>
  These snapshot options are used to generate a snapshot of the loaded Dart script:
    <snapshot-kind> controls the kind of snapshot, it could be kernel(default) or app-jit
    <file_name> specifies the file into which the snapshot is written
--version
  Print the VM version.

D:\>-
```

Figure 2.7: Setup was Successful for dart

Chapter 3

SYSTEM DESIGN AND ANALYSIS

3.1 System Architecture

A system architecture is the conceptual model that defines the structure, behavior and more views of a system. An architecture description is a formal description and representation of a system, organized in a way that supports reasoning about the structures and behaviors of the system. A system architecture can consist of system components and the sub-systems developed, that will work together to implement the overall system.

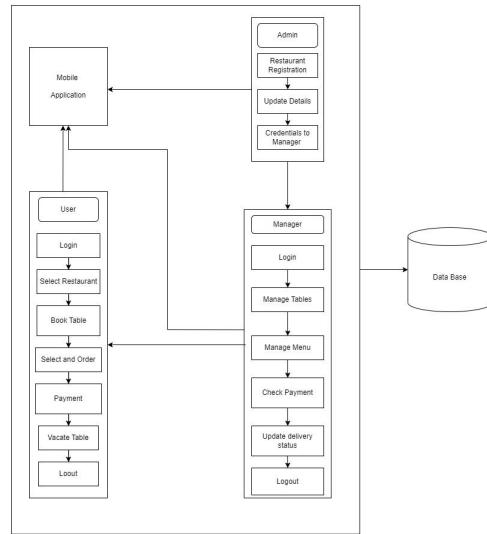


Figure 3.1: Architecture Diagram for restaurant management system

3.2 UML Diagrams

3.2.1 Use Case Diagram

Use case diagram shows a set of use cases and actors and their relationships. These diagrams are important in organizing and modeling the behaviors of a system. To model a system the most important aspect is to capture the dynamic behavior. The dynamic behavior means the behavior of the system when it is running/operating. Use case diagram is dynamic in nature there should be some internal or external factors for making the interaction. These internal and external agents are known as actors. So use case diagrams are consists of actors, use cases and their relationships. The diagram is used to model the system/subsystem of an application. A single use case diagram captures a particular functionality of a system.

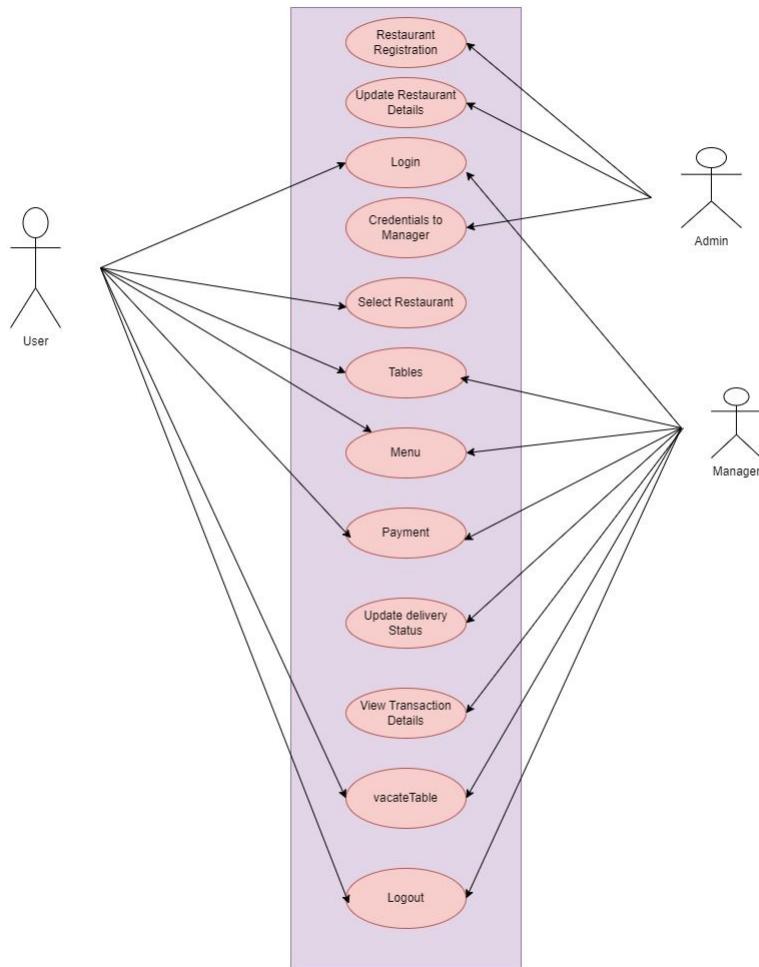


Figure 3.2: Use Case Diagram for restaurant management system

3.2.2 Sequence Diagrams

Sequence Diagrams are interaction diagrams that detail how operations are carried out. They capture the interaction between objects in the context of a collaboration. Sequence Diagrams are time focus and they show the order of the interaction visually by using the vertical axis of the diagram to represent time what messages are sent and when.

Purpose of Sequence Diagram

- Model high-level interaction between active objects in a system
- Model the interaction between object instances within a collaboration that realizes a use case
- Model the interaction between objects within a collaboration that realizes an operation
- Either model generic interactions (showing all possible paths through the interaction) or specific instances of a interaction (showing just one path through the interaction)

This diagram shows the sequential diagram for login page.

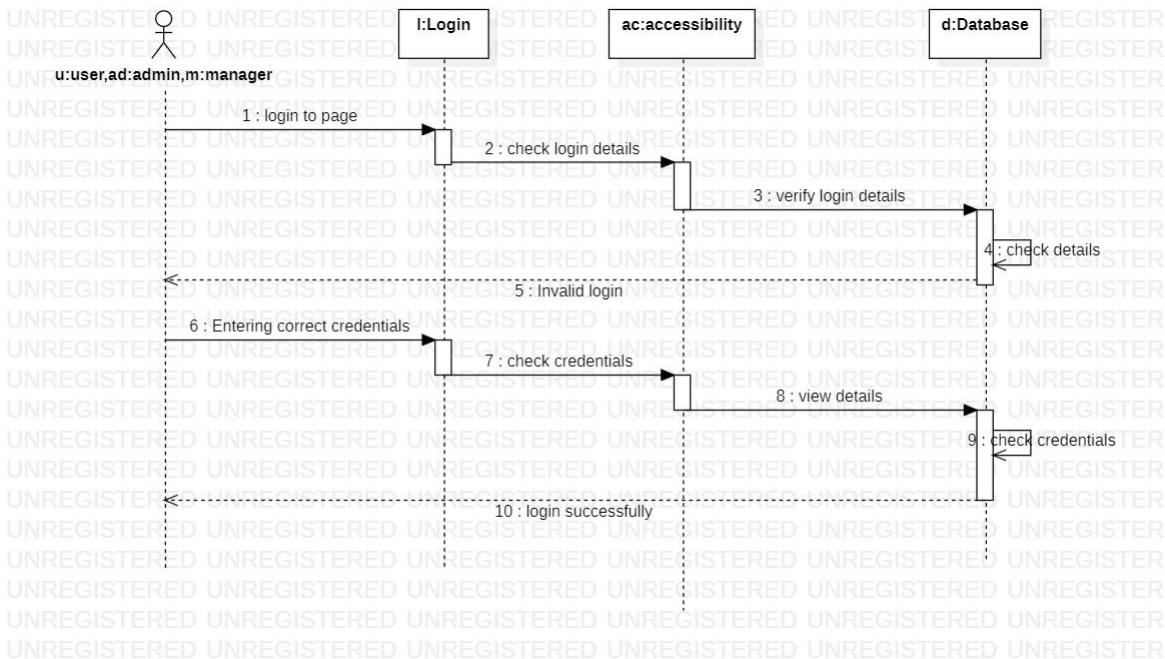


Figure 3.3: Sequence diagram for Login

The below diagram shows the sequence diagram for creating manager.

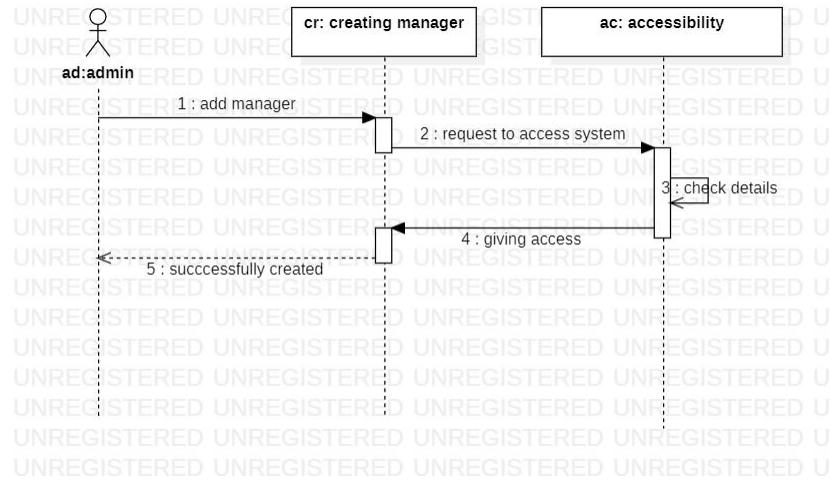


Figure 3.4: Sequence diagram for creating a manager

The below diagram shows the sequence diagram for selection of restuarant.

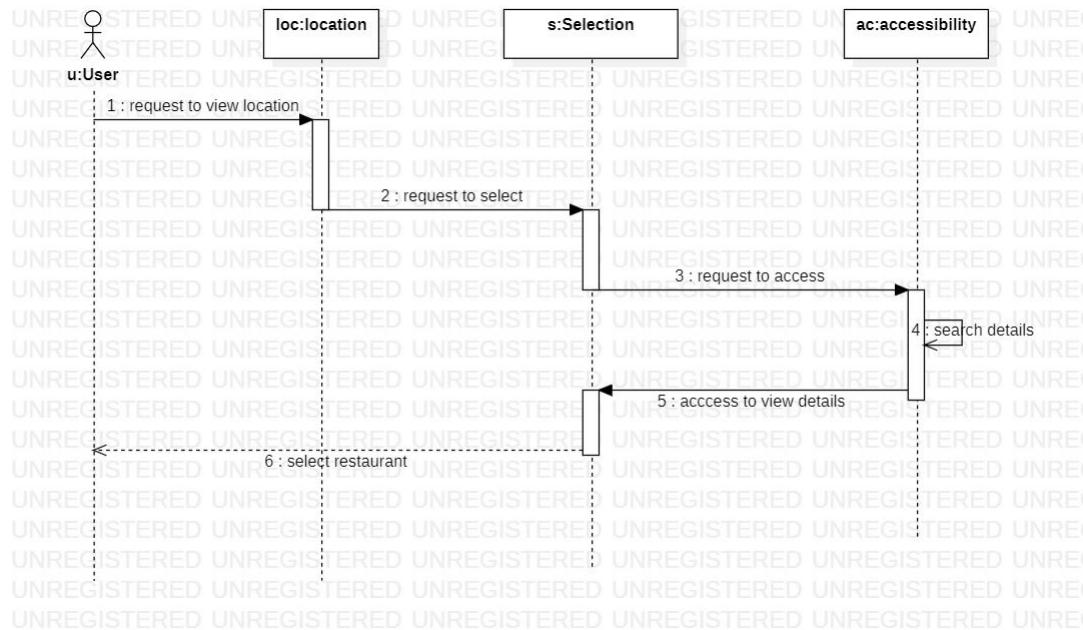


Figure 3.5: Sequence diagram for restaurant selection

The following diagram shows the sequence diagram for reservation of tables.

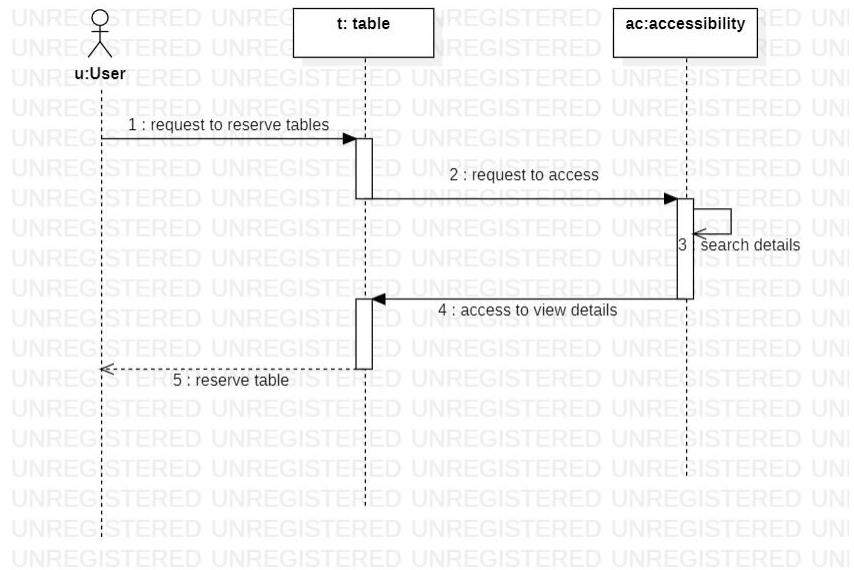


Figure 3.6: Sequence diagram for reservation of tables

The following diagram shows the sequence diagram for payment.

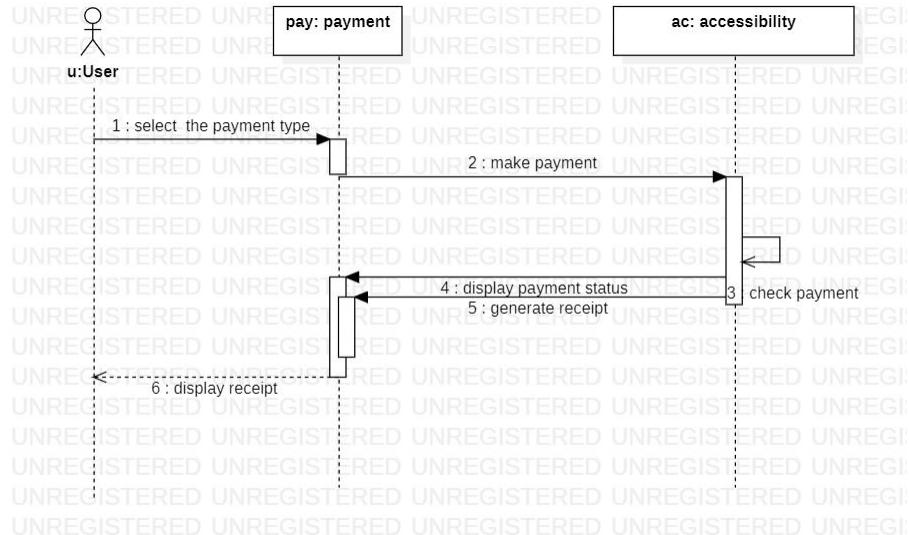


Figure 3.7: Sequence diagram for payment

The following diagram shows the sequence diagram for ordering food.

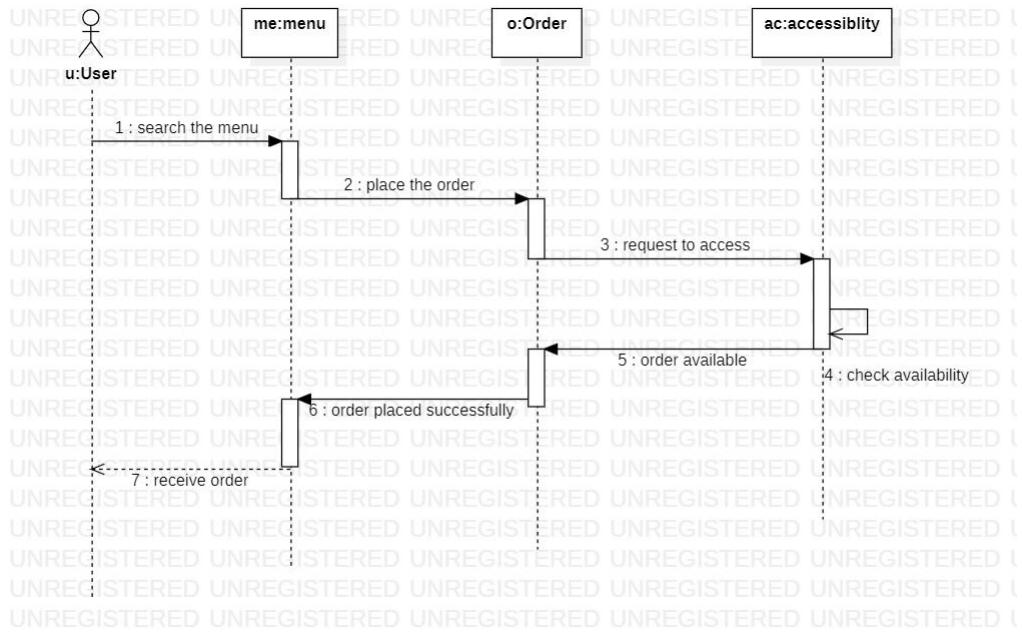


Figure 3.8: Sequence diagram for ordering food

The following diagram shows the sequence diagram for updating restaurant status.

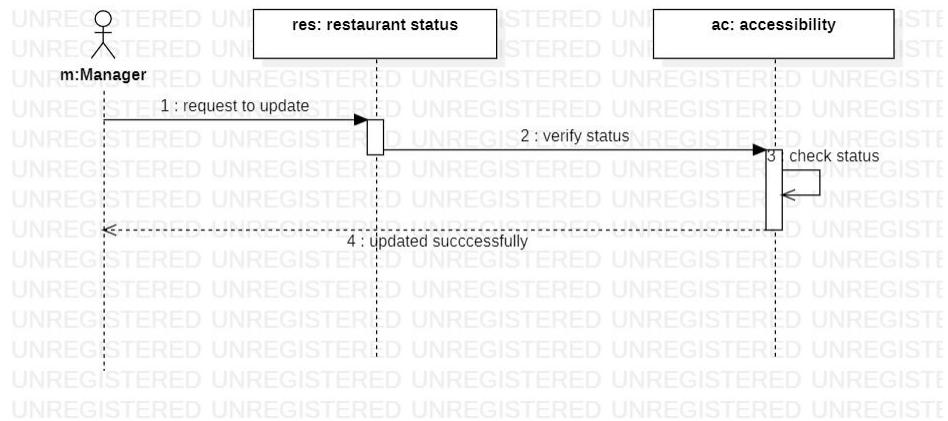


Figure 3.9: Sequence diagram for update restaurant status

It represents the methodology used in system analysis to identify, clarify and organize system requirements.

3.2.3 Class Diagram

A Class Diagram in Software engineering is a static structure that gives an overview of a software system by displaying classes, attributes, operations, and their relationships between each other. This Diagram includes the class name, attributes, and operation in separate designated compartments. Class Diagram helps construct the code for the software application development. It

defines the types of objects in the system and the different types of relationships that exist among them. It gives a high-level view of an application. This modeling method can run with almost all Object-Oriented Methods. A class can refer to another class. A class can have its objects or may inherit from other classes.

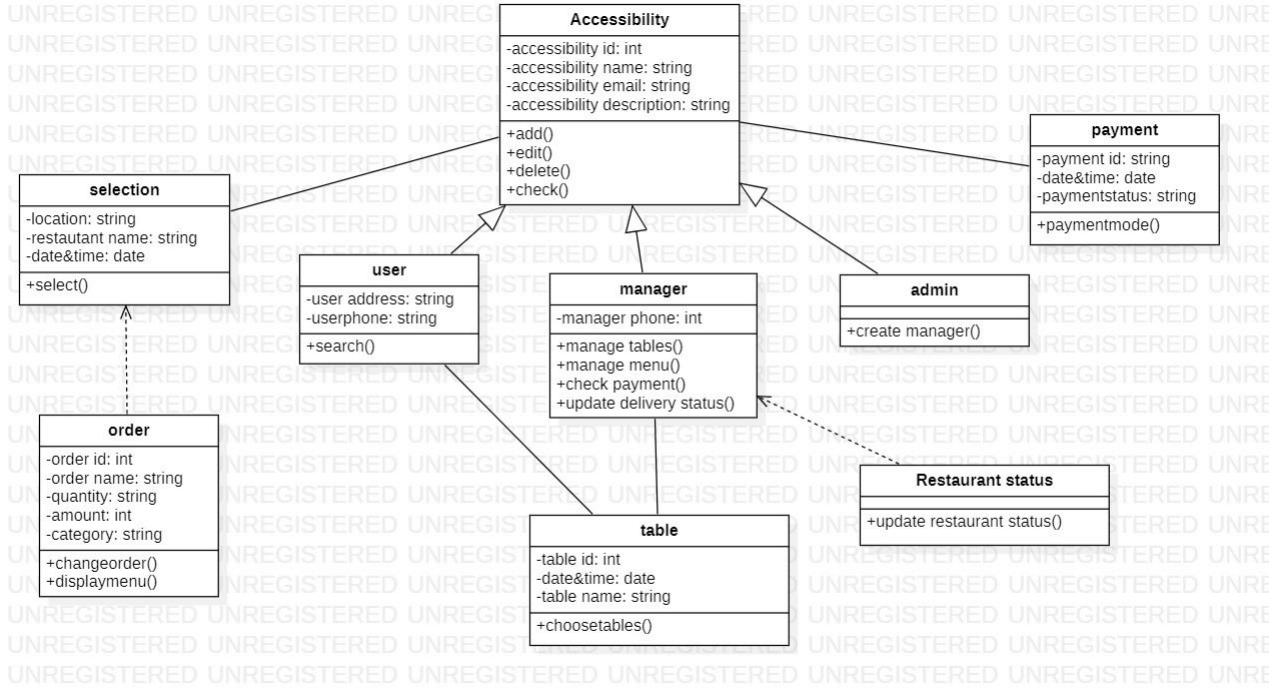


Figure 3.10: Class Diagram for restaurant management system

3.2.4 Object Diagram

Object is an instance of a class in a particular moment in runtime that can have its own state and data values. Likewise a static UML object diagram is an instance of a class diagram; it shows a snapshot of the detailed state of a system at a point in time, thus an object diagram encompasses objects and their relationships which may be considered a special case of a class diagram or a communication diagram.

- It is used to understand object behavior and their relationships practically.
- It is used to get a static view of a system.
- It is used to represent an instance of a system.

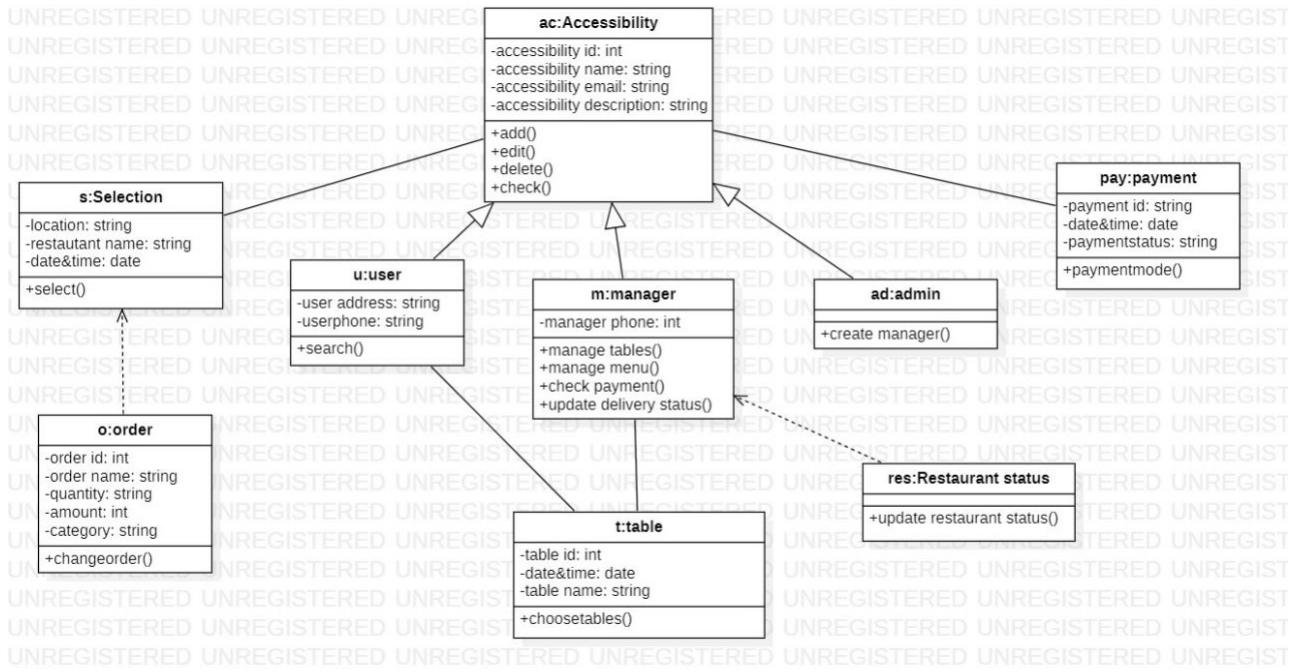


Figure 3.11: Object Diagram for restaurant management system

3.2.5 Activity Diagram

Activity Diagrams describe how activities are coordinated to provide a service which can be at different levels of abstraction. Typically, an event needs to be achieved by some operations, particularly where the operation is intended to achieve a number of different things that require coordination, or how the events in a single use case relate to one another, in particular, use cases where activities may overlap and require coordination. It is also suitable for modeling how a collection of use cases coordinate to represent business workflows. The activity diagram helps in envisioning the workflow from one activity to another. It puts emphasis on the condition of flow and the order in which it occurs.

- Modeling work flow by using activities.
- Modeling business requirements.
- High level understanding of the system's functionalities.
- Investigating business requirements at a later stage.

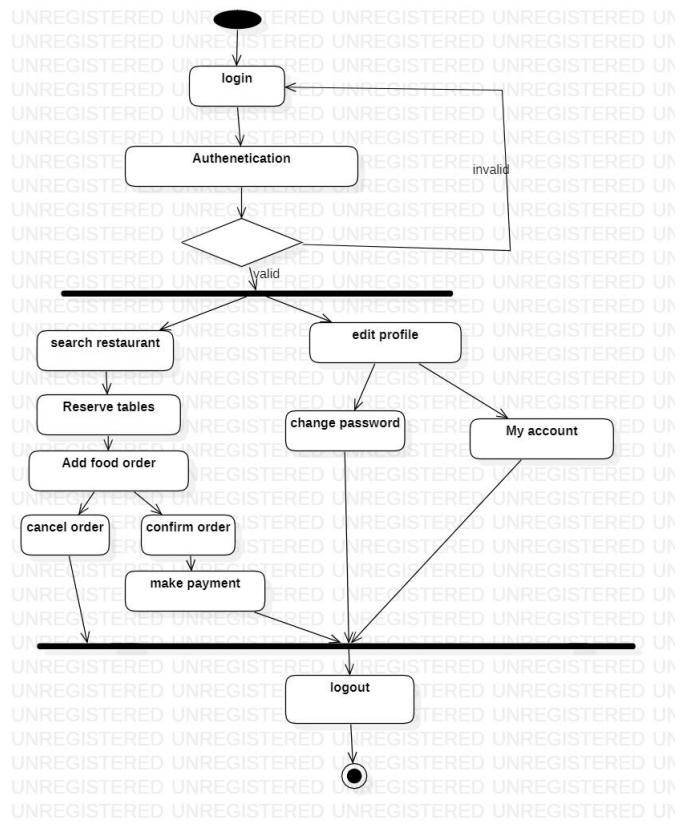


Figure 3.12: Activity Diagram for restaurant management system

3.2.6 State Chart Diagram

State chart diagram is used to describe the states of different objects in its life cycle. Emphasis is placed on the state changes upon some internal or external events. These states of objects are important to analyze and implement them accurately. State chart diagrams are very important for describing the states. States can be identified as the condition of objects when a particular event occurs.

- To model the dynamic aspect of a system.
- To model the life time of a reactive system.
- To describe different states of an object during its life time.
- Define a state machine to model the states of an object.

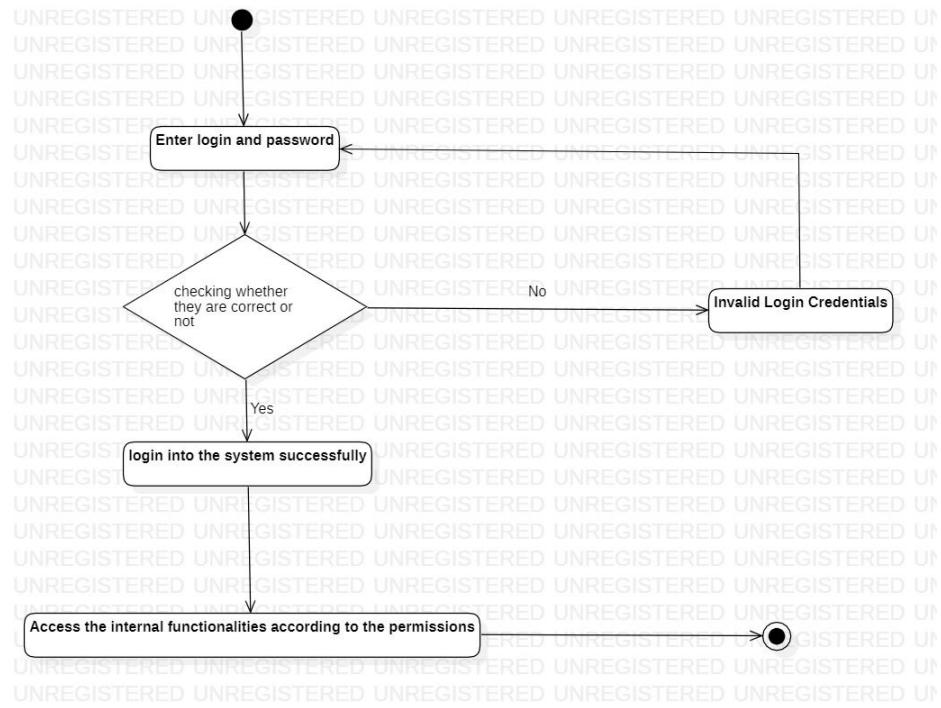


Figure 3.13: State Chart Diagram for restaurant management system

3.2.7 Component Diagram

UML Component diagrams are used in modeling the physical aspects of object-oriented systems that are used for visualizing, specifying, and documenting component-based systems and also for constructing executable systems through forward and reverse engineering. Component diagrams are essentially class diagrams that focus on a system's components that often used to model the static implementation view of a system. The component diagrams have remarkable importance. It is used to depict the functionality and behavior of all the components present in the system, unlike other diagrams that are used to represent the architecture of the system, working of a system, or simply the system itself.

- It envisions each component of a system.
- It constructs the executable by incorporating forward and reverse engineering.
- It depicts the relationships and organization of components.

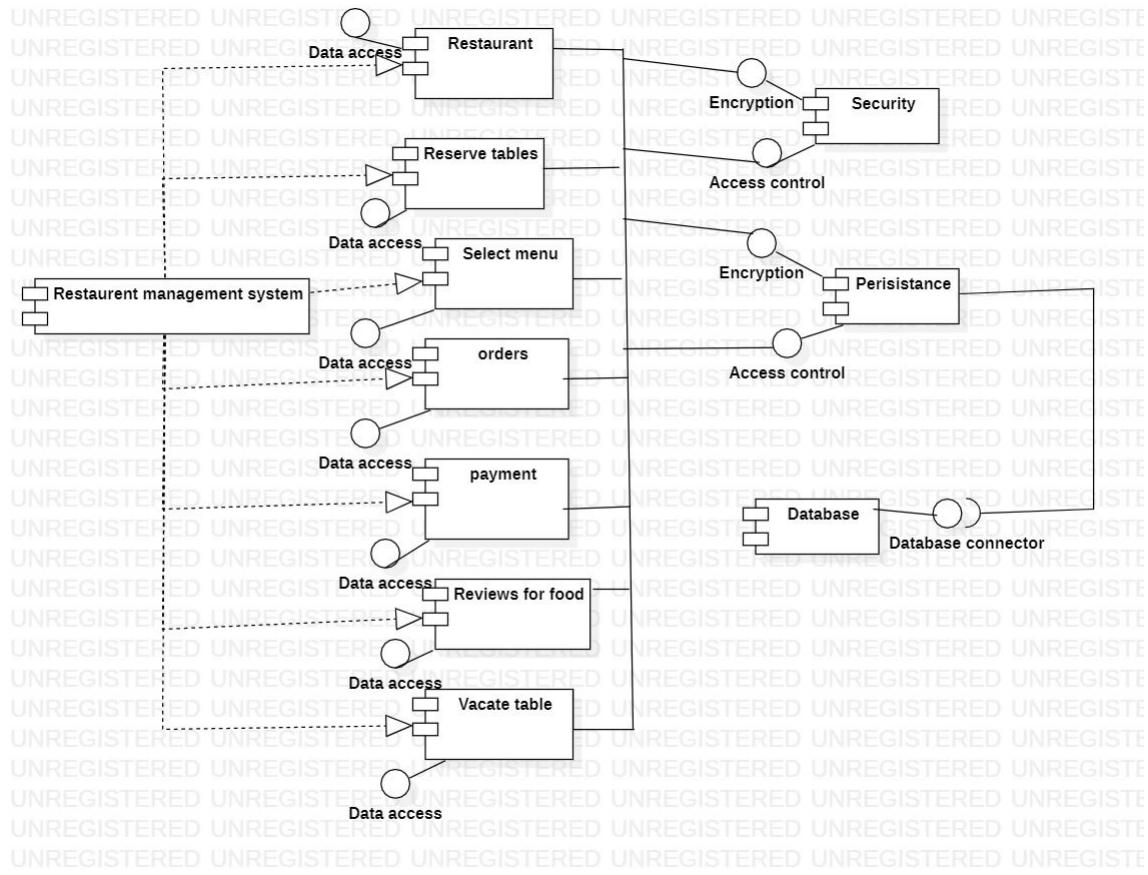


Figure 3.14: Component Diagram for restaurant management system

3.2.8 Deployment Diagram

A deployment diagram is a UML diagram type that shows the execution architecture of a system, including nodes such as hardware or software execution environments, and the middleware connecting them. Deployment diagrams are typically used to visualize the physical hardware and software of a system. Using it you can understand how the system will be physically deployed on the hardware. Deployment diagrams help model the hardware topology of a system compared to other UML diagram types which mostly outline the logical components of a system.

Deployment diagrams are used to visualize the hardware processors/ nodes/ devices of a system, the links of communication between them and the placement of software files on that hardware.

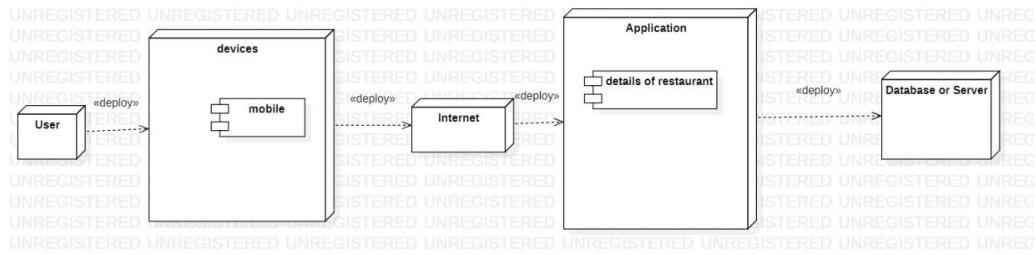


Figure 3.15: Deployment Diagram for restaurant management system

3.3 Existing System

As the restaurant systems around the world are following the traditional restaurant system that contains paper-based menu cards and uses pen and paper to take the orders from customers, and the orders are written are taken to the kitchen for food preparation. Through manual ordering there is a chance for a lot of mistakes, sometimes when there is a huge crowd in the restaurant the waiter will take order fastly, so in that case, the handwriting of waiter will not understand by kitchen department results in the preparation of food other than customer order will result in customer dissatisfaction, results in a lot of complaints raised by the customers about the services, receiving wrong order or even not receiving the order even after waiting for a long time. Sometimes the papers in which the order is also lost. Even it results in a lot of paper waste for taking orders, printing the menu cards, making bill receipts. In case if a menu card is printed wrongly or to change the menu available in the restaurant and even to change the prices of the menu it is not possible to change it easily and we have to print another menu card which results in the paper and money waste. This system is also a waste of time and energy of taking the orders waiting for the orders and even waiting for tables to sit and also the customers are not aware of the time taken to prepare the food. Hence the existing system results in a lot of inconvenience to both the customers and management. Hence a system is proposed that will help the management to increase the services and management and the customers cannot wait for waiters to take orders and it also reduces the time for both customers and restaurant staff and improves customer satisfaction.

3.3.1 Disadvantages of Existing System:

1. Manual work.
2. reduced limited human interaction
3. Increases customer time

3.4 Proposed System

Nowadays, wireless communication has been extended such as 3G, 4G, and 5G for efficient and effective communication between the users. This was developed based on the wireless channel bandwidth for the data transmission between M and E-commerce. The proposed smart restaurant system will help the customers to have the best dining experience and staff.

Nowadays, various services become available on smartphones. In this system, the customer will find the restaurants in his desired locations with the rating given by the other customers. After selecting the restaurant he will come know the information about the availability of tables in the desired restaurants prior. After entering the restaurant, customers can able to see the menu available in the restaurant along with their cost. They can browse the variety of dishes available in the list and they should add the required item into the cart with the desired quantity. They can also add or delete the items to or from the cart respectively. Along with the items, the cost of the items is also visible and the total cost of items in the cart also appears at the end. After adding all the required items to the cart he should have to do the bill payment online or in the form of cash and then click on the proceed button to place an order. The manager can mange the restaurant updates and transaction details. The data will store in the database about the items ordered and it helps to know which item is ordered more, which helps to manage the budgeting and planning according to it.

3.4.1 Advantages of Proposed System:

1. Improved work quality.
2. Provide automatic response.
3. No fee for use of tool.

3.5 Requirement Specification

3.5.1 Hardware Requirements:

The hardware requirement specifies each interface of the software elements and the hardware elements of the system. These hardware requirements include configuration characteristics.

Hardware Requirements

- Processor – above i3

- Hard Disk – 40 GB
- RAM – 16GB

3.5.2 Software Requirements

The software requirements specify the use of all required software products like data management system. The required software product specifies the numbers and version. Each interface specifies the purpose of the interfacing software as related to this software product.

Software Requirements

- Flutter
- Dart
- Firebase

Chapter 4

IMPLEMENTATION

4.1 Modules

The main modules in Smart Restaurant System are:

1. Admin Module
2. Manager Module
3. User Module

The code of admin module is placed at APPENDIX 1

4.1.1 Admin module

The admin module is a crucial component of a smart restaurant management system. It is a backend interface that allows the restaurant manager or owner to manage various aspects of the restaurant operations.

1. The admin creates a new manager account and generates a unique code for that account.
2. The admin provides the code to the manager via email, phone, or in-person

- **Menu Management:** The admin module allows the restaurant manager to create and manage the menu items. They can add or remove items, update prices and descriptions, and set up special deals or discounts.
- **Table Management:** The admin module allows the restaurant manager to manage table reservations and table assignments. They can view the real-time table status, assign tables to customers, and monitor table turnover time.

- **Employee Management:** The admin module allows the restaurant manager to manage the employee profiles, roles, and schedules. They can track attendance, manage shifts, and set up task assignments.
- **Order Management:** The admin module allows the restaurant manager to manage the order processing and order tracking. They can view the orders in real-time, manage order status updates, and monitor order fulfillment times.
- **Billing Management:** The admin module allows the restaurant manager to generate bills and process payments efficiently. They can apply discounts, manage different payment methods, and view payment reports.
- **Security and Access Control:** The admin module allows the restaurant manager to set up access controls and security features. They can set up user accounts, set up access levels, and manage passwords.

4.1.2 Manager Module

The code of Manager module is placed at APPENDIX 2

The manager module is another important component of a smart restaurant management system, designed to help restaurant managers streamline their operations and make informed decisions.

1. The manager opens the restaurant management system and navigates to the login page.
 2. On the login page, the manager is prompted to enter their unique code.
 3. The manager enters the code provided by the admin. If the code is valid, the system prompts the manager to access the application and manage the functions in the restaurant.
- **Dashboard:** The manager module provides an intuitive dashboard that displays real-time data on sales, customer feedback, inventory levels, and employee performance.
 - **Menu Management:** The manager module allows the manager to create and manage the menu items. They can add or remove items, update prices and descriptions, and set up special deals or discounts.
 - **Table Management:** The manager module allows the manager to manage table reservations and table assignments. They can view the real-time table status, assign tables to customers, and monitor table turnover time.
 - **Employee Management:** The manager module allows the manager to manage the employee profiles, roles, and schedules. They can track attendance, manage shifts, and set up task assignments.

- **Order Management:** The manager module allows the manager to manage the order processing and order tracking. They can view the orders in real-time, manage order status updates, and monitor order fulfillment times.
- **Billing Management:** The manager module allows the manager to generate bills and process payments efficiently. They can apply discounts, manage different payment methods, and view payment reports.
- **Analytics and Reporting:** The manager module allows the manager to access various reports and analytics. They can view sales reports, order reports, employee performance reports, and inventory reports.

4.1.3 User Module

The code of User module is placed at APPENDIX 3

The user module in a smart restaurant management system is designed to provide a seamless experience to customers who visit the restaurant. It includes the following features:

- **Table Reservation:** The user module allows customers to make table reservations online. They can select the date, time, and number of people in their party, and the system will show them available tables.
- **Ordering:** The user module allows customers to place orders directly from their table. They can browse the menu, select items, and customize them if needed. The orders are sent directly to the kitchen or bar for preparation.
- **Payment:** The user module allows customers to pay their bills through various payment methods such as credit cards, debit cards, or mobile wallets. They can split the bills or add tips if needed.
- **Feedback:** The user module allows customers to provide feedback on their dining experience. They can rate the food quality, service, ambiance, and overall experience. This feedback is useful for the restaurant to improve its operations and enhance customer satisfaction.
- **Order History:** The user module allows customers to view their order history and previous bills. They can use this information to track their spending, reorder their favorite items, or provide feedback on past orders.

4.1.4 How the system works?

Smart restaurant management system was developed to automate the process that are done in the restaurant and to make the process of the people of finding restaurants in the

near locations. In this system the super admin is the person who register the restaurant into the application and adds the managers who are working in the restaurants and they will give the credentials to the restaurant. The admin is the manager who login to the application and add the available dishes, table status, and he is the one who manages the payment details etc.

Finally the user can login to the application and by selecting the nearby or required location, they will get the restaurant details, which are available and registered in the selected locations. Therefore after entering into the application he gets the table details. From there they can select the table available and they like and by entering time they can book the table. Afterwards they can see the menu and they can add the items they want to cart, and they can place order by selecting the payment methods i.e online or cash on delivery. After completion of taking food they can able to give feedback to the restaurants and vacate the table.

Chapter 5

RESULTS

Project results are the changes or effects expected to take place after implementing the project. The results are generally positive improvements to the lives of the beneficiaries.

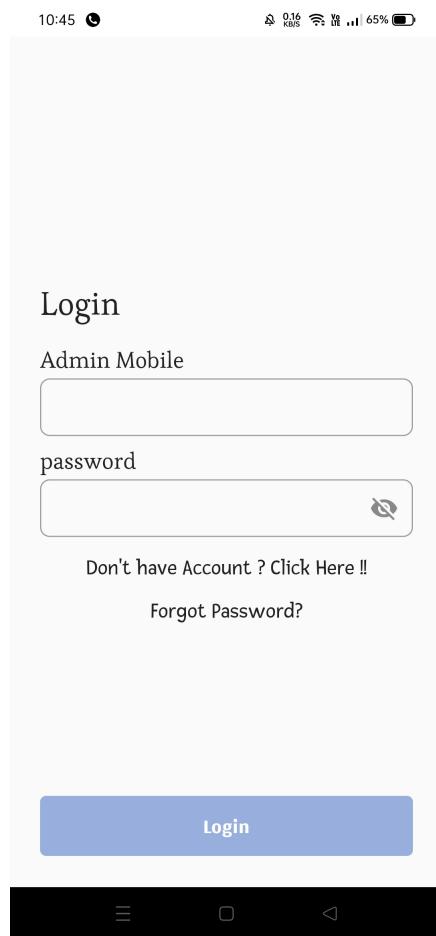


Figure 5.1: Login admin page

Add Account
Pavithra
Mobile
9032271280
Email
pavithra555@gmail.com
password

Restaurant Name
Usha Grand Spicy hot Restaurant
Restaurant Address
Near Housing Board colony
Restaurant Location
Mumbai
Create Account

Figure 5.2: Adding admin account

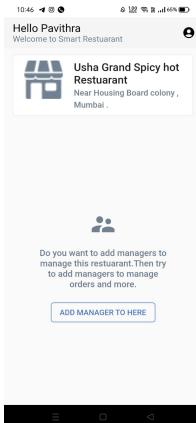


Figure 5.3: Adding manager

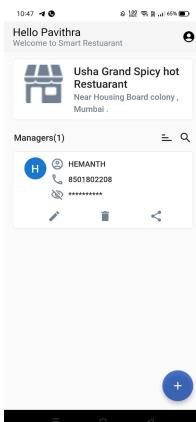


Figure 5.4: Adding Manager and available managers

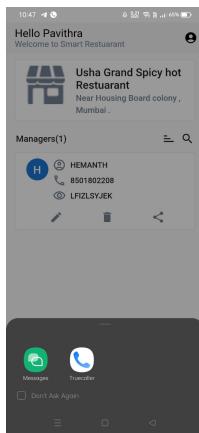


Figure 5.5: Sharing manager id

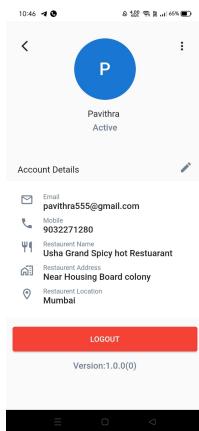


Figure 5.6: Admin details and logout page

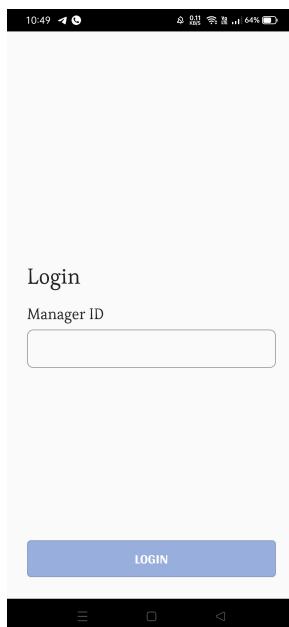


Figure 5.7: Login page of manager

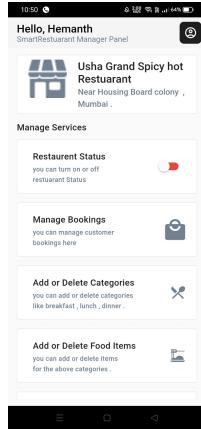


Figure 5.8: Welcome page of manager

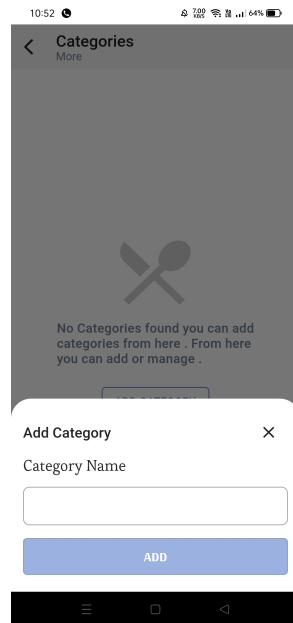


Figure 5.9: Adding categories

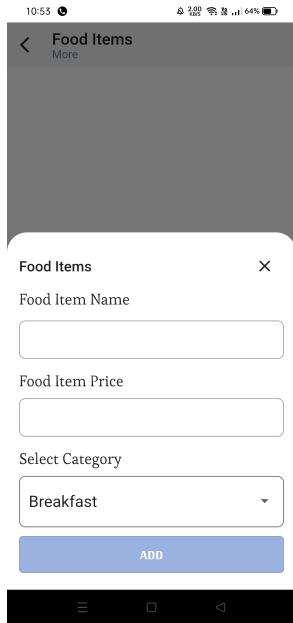


Figure 5.10: Adding food items

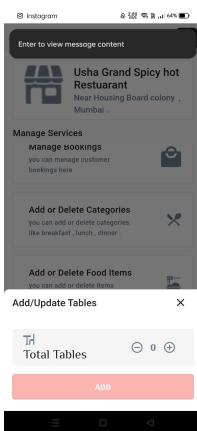


Figure 5.11: Adding Tables

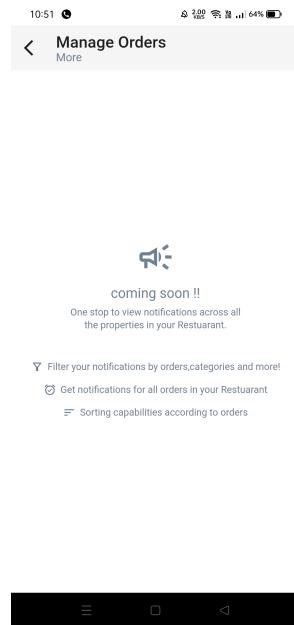


Figure 5.12: Managing orders

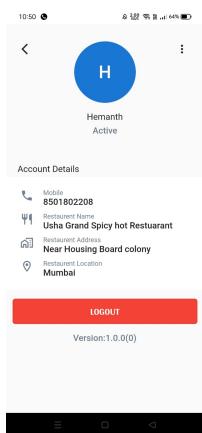


Figure 5.13: Manager details Logout from the Application

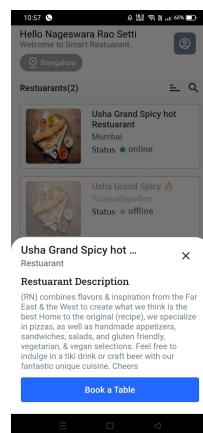


Figure 5.14: Restaurants available

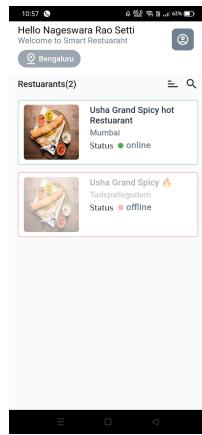


Figure 5.15: Selecting one of the Restaurant available

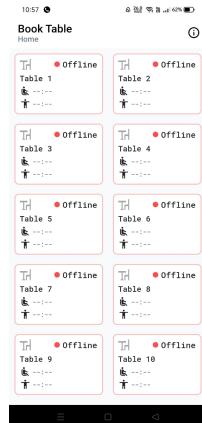


Figure 5.16: Available tables in restaurant

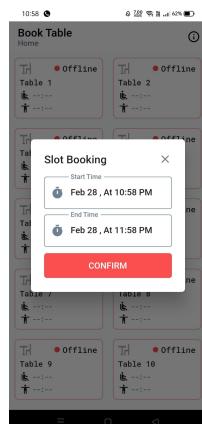


Figure 5.17: Table Booking



Figure 5.18: Menu page

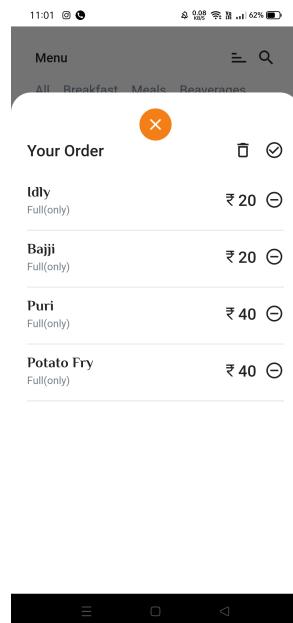


Figure 5.19: Added items

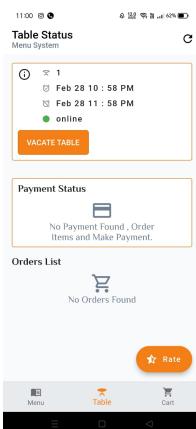


Figure 5.20: Table status of the customer

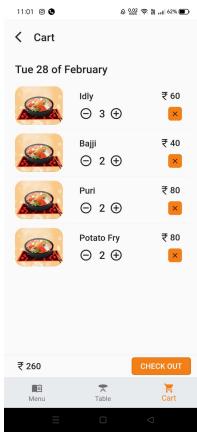


Figure 5.21: Items added in the cart.

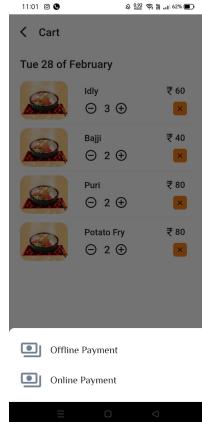


Figure 5.22: Payment options

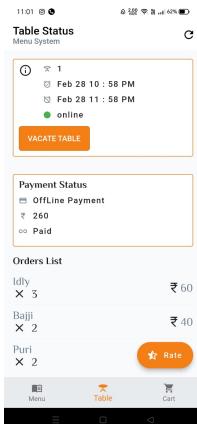


Figure 5.23: Order and payment details

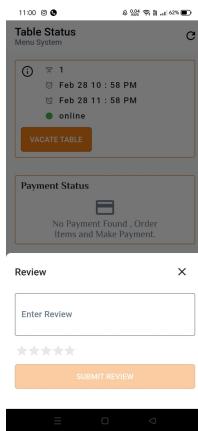


Figure 5.24: Review option

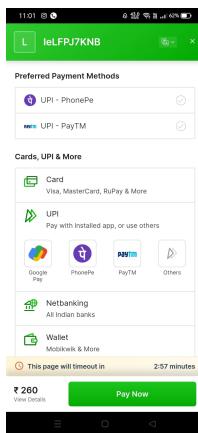


Figure 5.25: Online payment

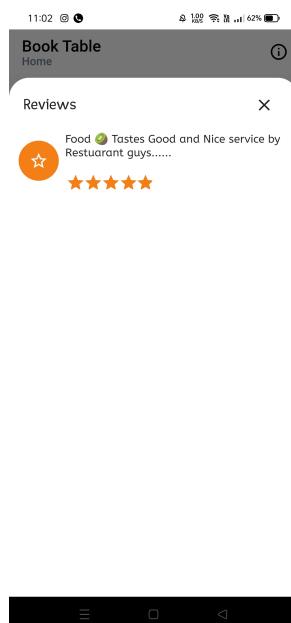


Figure 5.26: review of the user

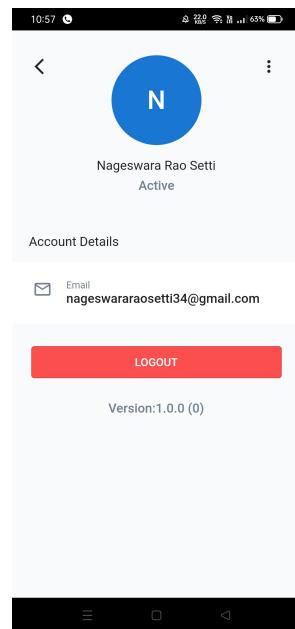


Figure 5.27: Logout from the Application

Chapter 6

TESTING

6.1 Testing

Testing is the process of detecting errors. Testing performs a very critical role for quality assurance and for ensuring the reliability of software. The results of testing are used later on during maintenance also. Software testing is the process used to access the quality of computer software. Software testing the empirical technical investigation conducted to provide stakeholder with information about the quality of the product or service under test, with respect to the context in which it is intended to operate

There are many approaches to software testing. Reviews, walk throughs or inspections are considered as static testing, and where actually running the program with a given set of test cases in a given development stage is referred to as dynamic testing. Software testing is used in association with verification and validation.

Verification:

Have we built the software right (i.e., does it match the specification).

Validation:

Have we built the right software (i.e., is this what the user wants). Software testing methods are traditionally divided into black box testing and white box testing; these two approaches are used to describe the point of view that a test engineer takes when designing test cases.

There are two types of testing:

1. Manual testing
2. Automation testing

6.2 Manual testing

Manual Testing is a kind of software testing in which a software tester develops and executes the test cases without using any automated testing tools. The main objective of manual testing is to detect the issues, bugs, and defects of a software application.

Manual testing is divided into 3 types:

1. White Box testing
2. Black Box testing
3. Gray Box testing

6.2.1 White Box Testing

White box testing is done when the tester has access to the internal data structure, code, and algorithms white box testing methods including creating tests to cause all statements in the program to be executed at least once. Other examples white box testing are mutations testing and fault injection methods White box testing methods can also be used to evaluate the completeness of a test suite that was created with black box testing methods.

6.2.2 Black Box Testing

Black box testing treats the software as a black-box without any understanding as to how the internals behave. It aims to test the functionality according to the requirements. Thus, the tester inputs the data and only sees the output from the test object. This level of testing usually requires through test cases to be provided to the tester who then can simply verify that for a given input, the output value (or behavior), is the same as the expected value specified in the test cases, Black box testing methods include equivalence partitioning boundary.

Black Box testing is further divided into 2 types:

1. Functional Testing
2. Non-Functional Testing

6.2.3 Gray Box Testing

Gray box testing is a method you can use to debug software and evaluate vulnerabilities. In this method, the tester has limited knowledge of the workings of the component being tested.

6.3 Black Box Testing

6.3.1 Functional Testing

Functional Testing is a type of software testing that validates the software system against the functional requirements/specifications. The purpose of Functional tests is to test each function of the software application, by providing appropriate input, verifying the output against the Functional requirements.

Functional testing mainly involves black box testing and it is not concerned about the source code of the application. This testing checks User Interface, APIs, Database, Security, Client/Server communication and other functionality of the Application Under Test. The testing can be done either manually or using automation.

Functional Testing is Sub divided into 3 types

1. Unit Testing
2. Integration Testing
3. System Testing

6.3.2 Non-Functional Testing

Non functional testing is a type of software testing that verifies non functional aspects of the product, such as performance, stability, and usability. Whereas functional testing verifies whether or not the product does what it is supposed to, non functional testing verifies how well the product performs.

Non-Functional testing is divided into 3 types:

1. Performance Testing
2. Usability Testing
3. Compatibility Testing

6.4 Functional Testing

6.4.1 Unit Testing

Unit Testing concentrates on each unit of the software as implemented in source code and is a white box oriented. Using the component level design description as a guide, important control paths are tested to uncover errors within the boundary of the module in the unit testing, the steps can be conducted in parallel for multiple components in my project. I tested all the modules individually related to main function codes.

6.4.2 Integration Testing

Integration Testing focus is on design and construction of the software architecture. Integration Testing is a systematic technique for constructing the program structure while at the same time conducting tests to uncover errors associated with interfacing. The objective is to take unit tested components and build a program structure that has been dictated by design. The goal here is to see if modules can be integrated properly, the emphasis being unit testing interfaces between modules.

Again Integration testing divides into 2 phases:

1. Incremental Testing
2. Non-Incremental Testing

Incremental Testing

Incremental testing is an approach used during the testing phase of integration tests. It uses a number of stubs and drivers to test each module independently. This helps identify any errors or defects in the modules.

1. Top-down
2. Bottom-up

Top-down

The top-down approach is a testing methodology used in incremental testing, where the testing of the higher-level components or modules is carried out first, followed by the integration of lower-level modules that support the higher-level components. mainly it allows for the testing of the most critical or high-risk components first, reducing the risk of critical issues being discovered late in the development cycle.

Bottom-up

The bottom-up approach is a testing methodology used in incremental testing, where the testing of individual components or modules is carried out first, followed by the integration of these components to form larger modules or subsystems, and finally, the integration of these subsystems into the complete system.

Non-Incremental Testing

Non-incremental Integration Testing. We will use this method when the data flow is complex, and it is difficult to find out who the parent and the child is. In such a case, we will create data in

any module and check whether the data exists in all existing modules.

6.4.3 System Testing

System testing of software or hardware is testing conducted on a complete, integrated system to evaluate the system compliance with its specified requirements. System testing falls thin the scope of black box testing, and as such, should require no knowledge of the inner design of the code or logic. Here the entire software system is tested. The reference document for this process is the requirements document, and the goal is to see if software meets its requirements.

6.5 Non-Functional Testing

6.5.1 Performance Testing

Performance testing checks how well software components work. These tests can find issues in software design and architecture performance. This is typically done by: Measuring response times, Identifying bottlenecks, Locating failure points. Performance tests ensure several elements of software quality. They validate that it's fast, scalable, stable, and reliable.

Performance testing is again divided into three types:

1. Load Testing
2. Stress Testing
3. Stability

Load Testing

Load testing checks how the software behaves under both normal and peak conditions. This is done to determine how much work load the software can handle before performance is negatively affected. You can perform load tests by running multiple applications simultaneously, subjecting a server to a high amount of traffic, or downloading a large quantity of files. Load tests are used to ensure fast and scalable software.

Stress Testing

Stress testing checks how the software behaves under abnormal conditions. This determines the limit at which the software will break. Stress tests help testers analyze what happens when a system fails. This ensures that software is recoverable, stable, and reliable.

Stability Testing

Stability Testing is the process for determining, through storage at defined conditions and testing at specific intervals, how long a drug substance or product remains safe and effective at particular storage conditions.

Scalability Testing

Scalability testing is important for ensuring that a system can grow and adapt to changing business requirements and user demands. It can help identify potential bottlenecks and performance issues before they become critical, allowing organizations to proactively address these issues and optimize their systems for scalability.

6.5.2 Usability Testing

Usability testing refers to evaluating a product or service by testing it with representative users. Typically, during a test, participants will try to complete typical tasks while observers watch, listen and takes notes.

6.5.3 Compatibility Testing

A compatibility test is an assessment used to ensure a software application is properly working across different browsers, databases, operating systems (OS), mobile devices, networks and hardware.

6.6 Automation

Automation testing is the process of testing software and other tech products to ensure it meets strict requirements. Essentially, it's a test to double-check that the equipment or software does exactly what it was designed to do. It tests for bugs, defects, and any other issues that can arise with product development.

6.7 Test Cases

Test case in software engineering in a set of conditions or variables under which tester will determine whether an application or software system meets specifications. The mechanism for determining whether a software program or system has passed or failed such a s known as a test oracle. It may take many test cases to determine that a software program system is functioning correctly. The written test cases are usually collected into test suites.

S.no	Testcase ID	Test Priority	Testcase Name	Inputs	Pre-Condition	Expected Results	Actual Results	Status Pass-P Fail-F
1	@Res01	high	Register	Registration	The user must have administrative privileges to access the register page	To open registration page	Opened registration page	P
2	@Res02	high	Register	Filling all fields	the user must have completed all the required fields on the registration form accurately	Registration successful: Data need to be stored in a database	Registration successful, Admin details are stored in a database	P
3	@Res03	low	Register	Missing fields	The registration form should clearly indicate which fields are mandatory and cannot be left blank, and provide clear instructions on how to complete each field.	Prompt an error if any search missing fields	Prompted Error for missing fields	P
4	@Res04	medium	Login	Enter username and password	The user must have a valid account in the restaurant application	Login successful; if login details are correct	Login successful	P
5	@Res05	medium	Login	Enter username and password	The user enters an incorrect username or password into the login form	Login is unsuccessful, if any of the fields are incorrect	Login unsuccessful	P
6	@Res06	high	Open home screen	Home screen Adding manager	Ensure that the user has successfully logged in to the system before attempting to access the home screen.	Manager should be added	Manager added	P
7	@Res07	high	Manager login page	Enter Manager Credentials	Manager have appropriate access from the admin to log in the manager page.	Manager page should be opened	Open manager page	P
8	@Res08	high	Adding categories and Items	Entering categories, items and tables	Ensure that the manager has successfully logged in to the system and has the appropriate permissions to add categories and items.	Categories, Food items and table list should be added	Categories, Food items and table list was added	P
9	@Res09	medium	Select Restaurant	Available restaurants will be displayed	Ensure that the user interface for selecting a restaurant is working properly and that it is easy to use.	Selected restaurant should be opened	Opened selected restaurant	P
10	@Res10	high	Table Booking	Table booking	the restaurant must open for booking a table	The table should be booked by clicking on particular table.	The table was booked by clicking on particular table.	P
11	@Res11	high	Ordering food	Ordering Food	for ordering the menu is available in the restaurant	The food should be ordered by selected and checkout button	The food was ordered by selected and checkout button	P
12	@Res12	high	Payment	Online and offline modes available	Ensure that the system supports the payment options that the user wants to use, such as credit/debit card, cash, or mobile payments.	Payment should be done by selecting particular payment	Payment was done by selecting particular payment	P
13	@Res13	low	Logout	logout	we should log out of the application	The user has to log out by clicking on logout button	The user will be logged out by clicking logout button	P

Table 6.1: Testcases

Chapter 7

CONCLUSION & FUTURE ENHANCEMENTS

Conclusion

The goal of our project Smart Restaurant System is to enhance the eating experience for customers while automating typical restaurant procedures including table reservations, menu selection, ordering, and paying. The manager may also oversee the entire process through a control panel. It will lessen the workload in restaurants, as well as the requirement for servers and paper waste. The user can locate the restaurant using this system, reserve a table, view the menu items, choose the menu of interest, and place an order using this system solely. He has the option of paying the bill offline or online. Using the manager panel, the manager may oversee the entire restaurant. This technology will shorten wait times for both customers and the manager while also reducing manual errors in the restaurant. Customer satisfaction is improved by this system.

Future Enhancements

In future a common application for all the restaurants will be developed. So using this customer can select the restaurant of his interest and go through the process that designed in current system. Online food delivery will also be declined in future. Automatic serving will also be implemented.

Bibliography

- [1] V. Liyanage, A. Ekanayake, H. Premasiri, P. Munasinghe, and S. Thelijjagoda, “Foody-smart restaurant management and ordering system,” in *2018 IEEE Region 10 Humanitarian Technology Conference (R10-HTC)*, IEEE, 2018, pp. 1–6.
- [2] N. Koubai and F. Bouyakoub, “Myrestaurant: A smart restaurant with a recommendation system,” *International Journal of Computing and Digital Systems*, vol. 8, no. 02, pp. 143–156, 2019.
- [3] S. Shedole, “Smart restaurant system using android,”
- [4] K. V. Joshi, A. P. Dhawale, C. Y. Dhekane, *et al.*, “Automated restaurant service management system,” *International Journal of Innovations in Engineering Research and Technology*, pp. 1–3,
- [5] C.-C. Cheng, Y.-Y. Chang, and C.-T. Chen, *Construction of a service quality scale for the online food delivery industry*, 2021.
- [6] M. Bolaños, M. Valdivia, and P. Radeva, “Where and what am i eating? image-based food menu recognition,” in *Computer Vision–ECCV 2018 Workshops: Munich, Germany, September 8–14, 2018, Proceedings, Part VI 15*, Springer, 2019, pp. 590–605.
- [7] S. T. Nikhitha Deshpande Shivani, “A review paper on smart restaurant ordering system,” Springer, 2018, pp. 590–605.
- [8] J. Harpanahalli, K. Bhingradia, P. Jain, and J. Koti, “Smart restaurant system using rfid technology,” in *2020 Fourth International Conference on Computing Methodologies and Communication (ICCMC)*, IEEE, 2020, pp. 876–880.
- [9] T. Qamar, S. B. Tariq, A. Aqeel, F. Jaffri, and T. Hussain, “Smartable-digital ordering system for restaurant using android,”
- [10] B. Bigliardi, E. Bottani, and S. Filippelli, “A study on iot application in the food industry using keywords analysis,” *Procedia Computer Science*, vol. 200, pp. 1826–1835, 2022.
- [11] U. Ravekar and S. Singh, “Data centric smart restaurant management system,” *Int. Res. J. Eng. Tech.(Irjet)*, vol. 6, p. 1425, 2019.

- [12] B. J. Gregorash, "Restaurant revenue management: Apply reservation management?" *Information Technology & Tourism*, vol. 16, no. 4, pp. 331–346, 2016.
- [13] S. Umap, S. Surode, P. Kshirsagar, M. Binekar, and N. Nagpal, "Smart menu ordering system in restaurant," *IJSRST*, vol. 4, no. 7, pp. 207–212, 2018.
- [14] T. Qamar, S. B. Tariq, A. Aqeel, F. Jaffri, and T. Hussain, "Smartable-digital ordering system for restaurant using android,"
- [15] D. Priyanka, P. Prathyusha, C. Tejeswini, D. V. Vardhan, C. R. Prasad, and W. Urban, "Smart food quality testing and ordering system using at mega328 in restaurants," *International Journal of Scientific Research and Engineering Development*, vol. 3, no. 1, pp. 645–50, 2020.
- [16] H. Saeed, A. Shouman, M. Elfar, M. Shabka, S. Majumdar, and C. Horng-Lung, "Near-field communication sensors and cloud-based smart restaurant management system," in *2016 IEEE 3rd World Forum on Internet of Things (WF-IoT)*, IEEE, 2016, pp. 686–691.
- [17] A. P. Kapoor and M. Vij, "Technology at the dinner table: Ordering food online through mobile apps," *Journal of Retailing and Consumer Services*, vol. 43, pp. 342–351, 2018.
- [18] M. D. Jakhete and P. C. Mankar, "Implementation of smart restaurant with e-menu card," *International Journal of computer applications*, vol. 119, no. 21, 2015.
- [19] L. Tarjan, I. Šenk, S. Tegeltija, S. Stankovski, and G. Ostojic, "A readability analysis for qr code application in a traceability system," *Computers and Electronics in Agriculture*, vol. 109, pp. 1–11, 2014.
- [20] A. Bhargave, N. Jadhav, A. Joshi, P. Oke, and S. Lahane, "Digital ordering system for restaurant using android," *International journal of scientific and research publications*, vol. 3, no. 4, pp. 1–7, 2013.

PRO vs PO RELEVANCE

Project Outcomes:

OutCome	Description
PRO 1	Identifying the problems by doing a through literature survey of the existing research related to restaurant management system and generate solutions to the same with innovative ideas.
PRO 2	Analyze, design and develop a solution for the problems that arises through publications and reporting works.
PRO 3	Develop employability and the ability to work in a team following the best ethical practices with a spirit for life-long learning and sharpening communication and presentation skills for developing a system to manage restaurant activities.
PRO 4	Make use of appropriate tools or techniques for sustainable development of a solution.
PRO 5	Create interest in researching different sources to perform restaurant activities online.
PRO 6	Construct a platform that makes users to interact with proposed system.

Table 7.1: Project Outcomes

PRO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
PRO1	3	3	3	2	3	0	2	0	2	3	0	1	2	2
PRO2	3	2	3	2	3	1	3	0	2	3	1	2	3	2
PRO3	2	2	2	1	3	2	1	3	0	2	1	1	3	2
Overall Course	3	2	1	2	3	1	2	1	2	3	1	1	3	2

Table 7.2: Summary of Project Outcomes mapping to Program Outcomes Project Phase - 1

PRO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
PRO4	2	3	2	1	3	0	1	1	1	3	2	0	3	1
PRO5	2	3	2	2	2	1	1	1	3	2	1	1	3	1
PRO6	2	3	1	2	2	2	2	0	3	3	1	2	3	2
Overall Course	2	3	2	2	2	1	2	1	2	3	1	1	3	2

Table 7.3: Summary of Project Outcomes mapping to Program Outcomes Project Phase - 2

PROs Relevance TO POs

PRO	PO	PI	Relevance
PRO1	PO1	1.6.1	Apply process mapping to assess restaurant operations
	PO2	2.6.3	Determine the problem's current solutions, especially those that involve making justifiable approximations and assumptions like security risks and revenue projections
	PO3	3.6.1	able to investigate design options(slot booking, reservation,payment)for a frameworkto improve the restaurant management system
	PO4	4.4.3	Able to choose appropriate hardware/software tools to conduct the experiment.
	PO5	5.6.2	Having the ability to review the state of literature by surveying.
	PO7	7.4.2	Apply sustainable development and preventative engineering principles to a product or engineering activity that is important to the operations of a restaurant.
	PO9	9.4.2	To achieve a goal, put into practise the best practises such as clear and consistent communication,reservation managementfor effective teamwork
	PO10	10.4.2	checking engineering papers for smart restaurant management system that are easy to understand,well supported, and well constructed.
	PO12	12.6.2	Analyze reasearch papers and technical conferences for future process.
	PO1	1.7.1	Apply database-design and user interface design to solve an existing problems
	PO2	2.8.3	The successfull implementation and integration of the vendor's software into the restaurant operations
	PO3	3.5.6	To be able to create introduction and system overview for an managing of restaurant activities
PRO2	PO4	4.6.4	Use the data stored in database to combine facts and knowledge about the issue in order to draw the proper improved reporting.
	PO5	5.4.2	Develop analytics and reporting tools address manual tasks in restaurant and adapt,modify, and expand them
	PO6	6.3.1	Efficiency ,cost-effectiveness,automation,safety and security must be connected to the application.

PRO	PO	PI	Relevance
PRO3	PO7	7.3.1	Determine the risks and effects throughout the module checking in applications lifecycle.
	PO9	9.4.1	Recognize a variety of working and learning preferences of team members.
	PO10	10.4.3	create a logical flow of ideas in a document or presentation to make the main point evident
	PO11	11.6.1	Identify the management resources required to complete the tasks in the project.
	PO12	12.4.2	The technology is been updating every time so management system may improved.
PRO4	PO1	1.5.1	Verify data efficiently by doing so..
	PO2	2.8.2	Analyze and interpret the results using mobile applications.
	PO3	3.8.2	Able to implement and integrate the modules(user,admin,manager) for restaurant.
	PO4	4.6.1	Use appropriate procedures, tools, and techniques to collect information in restaurant
	PO5	5.6.1	Project may be limited by budget constraints,making it difficult to implement all desired improvements or intiatives
	PO6	6.4.1	Interpret applicable laws,rules and standards for mobile application and describe how they contribute to public safely
	PO7	7.4.1	Describe quality ,time and risk manageements for sustainable development
	PO8	8.4.2	Examine and apply responsibility,honesty and integrity related to project
	PO10	10.5.1	Listen to and comprehend information, instructions, and viewpoints of others in the project.
	PO11	11.5.1	Analyze and select the most appropriate proposal based on economic and financial considerations related to restaurant project
	PO12	12.5.1	Identify historical engineering technological advancements that forced practitioners to pursue education in order to keep current with trends
	PO1	1.6.1	To create a smart restaurant management system, apply engineering principles.
	PO2	2.6.4	Compare and contrast alternative solution/methods to select the best methods

PRO	PO	PI	Relevance
PRO5	PO3	3.8.2	Able to implement activities and integrate the modules.
	PO4	4.6.2	Analyze data critically for patterns and relationships mentioning any faults or constraints.
	PO5	5.5.2	Use the mobile application frameworks appropriate to your subject with competence
	PO7	7.3.2	Understand the relationship between the Infrastructure and energy consumption of sustainability.
	PO8	8.3.1	Determine instances of misuse of user information and offer morally sound alternatives.
	PO9	9.5.4	Maintain composure in difficult situations like in giving inputs as notepad files only.
	PO10	10.5.1	Listen to and comprehend information, instructions, and viewpoints of others.
	PO11	11.4.1	Give an overview of the various financial and economic expenses and benefits of the smart food service management programme.
	PO1	1.7.1	Developing a capacity plan that outlines the number of tables that will be available in restaurant based on management actions in restaurant.
	PO2	2.6.1	Reframe the computer-based system into interconnected modules (user,admin,manager)
	PO3	3.8.1	Able to refine restaurant management system architecture design into a detailed design within the existing constraints
	PO4	4.4.1	Define a problem for purposes of investigation, its scope and importance of the smart restaurant activities.
	PO5	5.6.2	Verify the credibility of results from flutter tool use with reference .
	PO6	6.4.1	Interpret legislation, regulations, codes, and standards relevant to your project modules.
	PO7	7.3.1	Determine the risks and effects that may arise during the system testing
	PO8	8.3.1	Identify situations of unethical professional conduct and propose ethical alternatives.
	PO9	9.5.4	Maintain composure in difficult situations.

PRO	PO	PI	Relevance
PRO6	PO10	10.6.1	Create UML diagrams reports and drawings to complement writing and presentations
	PO11	11.6.2	Use agile project management tools to schedule an engineering project, so it is completed on time and on budget
	PO12	12.6.1	Source and comprehend technical literature and other credible sources of information.
	PO1	1.6.1	Apply engineering principles for smart system for managing restaurants. .
	PO2	2.8.4	Arrive at conclusions with respect to the objectives.
	PO3	3.5.5	Explore and synthesize system requirements from larger social and professional concerns.
	PO4	4.6.3	Represent grading and test cases data in tabular forms so as to facilitate analysis and explanation of the data, and drawing of conclusions.
	PO5	5.4.2	Create/adapt/modify/extend tools and techniques to solve issues in restaurant activities.
	PO6	6.3.1	Modules(user,admin,manager) in the application should be identified and explained.
	PO7	7.4.2	Apply principles of preventive engineering and sustainable development to an engineering activity or product relevant to the restaurant
	PO9	9.5.4	Maintain composure in difficult situations to complete the project
	PO10	10.6.2	use a presentation or paper that uses a mix of media to effectively communicate a message
	PO11	11.4.1	Describe various economic and financial benefits of the application
	PO12	12.4.1	Describe the rationale for the requirement for continuing professional development for the restaurant project.

Table 7.4: PRO Relevance to PO

PROs Relevance to PSO

PRO	PSO	Relevance
PRO1	PSO1	Ability to develop objectives of system
	PSO2	To deploy stoppage of manual activities in restaurant
PRO2	PSO1	To deploy solutions for the existing problem for a smart restaurant system
	PSO2	To analyze existing real-world problems related to restaurant activities
PRO3	PSO1	To develop appropriate methods like implementing new services to solve existing the problem.
	PSO2	Ability to apply ethical principles to proposed system.
PRO4	PSO1	To deploy services based on the copied sources in the mobile.
	PSO2	Ability to develop mobile application using new technologies
PRO5	PSO1	To implement code for developing restaurant services through online.
	PSO2	Ability to apply services to the proposed system.
PRO6	PSO1	Choosing alternative techniques to work with the user interface through framework.
	PSO2	Studying about user component interface.

Table 7.5: PRO Relevance to PSO

APPENDIX 1

```
1 import 'package:flutter/material.dart';
2 import 'package:flutter/services.dart';
3 import 'package:go_router/go_router.dart';
4 import 'package:shared_preferences/shared_preferences.dart';
5
6 import 'package:smartrestuarantadmin/bloc/bloc_exports.dart';
7 import 'package:smartrestuarantadmin/model/model_exports.dart';
8 import 'package:smartrestuarantadmin/screens/screen_exports.dart';
9 import 'package:smartrestuarantadmin/utilis/utilis_export.dart';
10 import 'package:smartrestuarantadmin/widgets/widgets_export.dart';
11
12 import 'package:google_fonts/google_fonts.dart';
13
14 class AdminLogin extends StatefulWidget {
15   const AdminLogin({super.key});
16
17   @override
18   State<AdminLogin> createState() => _AdminLoginState();
19 }
20
21 class _AdminLoginState extends State<AdminLogin> {
22   final TextEditingController mobileController = TextEditingController();
```

```
23   final TextEditingController passwordController =
24     TextEditingController();
25
26   void _createAccountPressHandler() {
27     Navigator.push(
28       context,
29       MaterialPageRoute(
30         builder: (context) => const AddRestuarant(),
31       ),
32     );
33   }
34
35   bool isObscure = true;
36
37   bool isEnabled = false;
38
39   final FocusNode _mobileFocus = FocusNode(); //added globally
40
41   final FocusNode _passwordFocus = FocusNode();
42
43   @override
44
45   void initState() {
46     super.initState();
47
48     createRestaurantBloc.isObscureActiveStream.listen((event) {
49
50       if (mounted) {
51
52         setState(() => isObscure = event);
53
54       }
55     });
56
57     createRestaurantBloc.enableButtonStream.listen((event) {
58
59       if (mounted) {
60
61         setState(() => isEnabled = event);
62
63       }
64     });
65
66     createRestaurantBloc.showToastMessageStream.listen((event) {
67
68       if (mounted) {
```

```
52     ToastMessage.showToast(
53         message: event
54             ? accountCreatedSuccessfullyText
55             : passwordUpdatedSuccessfullyText ,
56     );
57     Navigator.pop(context);
58 }
59 });
60 }
61
62 void _inputChangeListner() {
63     createRestuarantBloc.enableLoginButtonWithCredinals(
64         checkCredinals: CreateAccount(
65             mobile: mobileController.text ,
66             password: passwordController.text ,
67         ),
68     );
69 }
70
71 void _redirectDashBoard() async {
72     SharedPreferences pref = await SharedPreferences.getInstance();
73     pref.setString("mobilenumber", mobileController.text);
74     Navigator.pushReplacement(
75         context,
76         MaterialPageRoute(
77             builder: (context) => DashBoard(
78                 phoneNumber: mobileController.text ,
79             ),
80             ),
81     );
}
```

```
82     }
83
84     void _changePasswordPressHandler() {
85         Navigator.push(
86             context,
87             MaterialPageRoute(
88                 builder: (context) => const ForgetPassword(),
89             ),
90         );
91     }
92
93     @override
94     void dispose() {
95         mobileController.dispose();
96         passwordController.dispose();
97         super.dispose();
98     }
99
100    @override
101    Widget build(BuildContext context) {
102        return WillPopScope(
103            onWillPop: () async {
104                SystemNavigator.pop();
105                return false;
106            },
107            child: SafeArea(
108                child: Scaffold(
109                    floatingActionButtonLocation:
110                        FloatingActionButtonLocation.centerFloat,
111                    body: _loginForm(),
112                ),
113            ),
114        );
115    }
116}
```

```
112         floatingActionButton: _loginButton(),
113
114     ),
115 );
116 }
117
118 Widget _loginButton() {
119     return SizedBox(
120
121         height: 70.0,
122
123         width: double.infinity,
124
125         child: Padding(
126
127             padding: const EdgeInsets.symmetric(horizontal: 25.0, vertical:
128
129                 10.0),
130
131             child: StreamBuilder<bool>(
132
133                 stream: createRestuarantBloc.enableButtonLoadingStream,
134
135                 builder: (context, snapshot) {
136
137                     bool loading = snapshot.data ?? false;
138
139                     return loading
140
141                         ? const Center(child: CircularProgressIndicator())
142
143                         : CustomButton(
144
145                             function: isButtonEnabled ? _redirectDashBoard : null,
146
147                             shape: RoundedRectangleBorder(
148
149                                 borderRadius: BorderRadius.circular(5.0),
150
151                             ),
152
153                             child: Text(
154
155                                 loginText,
156
157                                 style: GoogleFonts.gidugu(
158
159                                     fontWeight: FontWeight.w400,
160
161                                     fontSize: 25.0,
162
163                                     color: whiteColor),
164
165                             ),
166
167                         ),
168
169                     );
170
171                 )
172             );
173         );
174     );
175 }
```

```
141
142
143
144
145
146
147
148
149 Widget _loginForm() {
150     return Padding(
151         padding: const EdgeInsets.symmetric(horizontal: 25.0),
152         child: Column(
153             mainAxisAlignment: MainAxisAlignment.start,
154             mainAxisSize: MainAxisSize.center,
155             children: [
156                 const TextWidget(
157                     title: loginText,
158                     fontSize: 27.0,
159                 ),
160                 const SizedBox(
161                     height: 18.0,
162                 ),
163                 const TextWidget(
164                     title: userMobileText,
165                     fontSize: 19.0,
166                 ),
167                 const SizedBox(
168                     height: 4.0,
169                 ),
170                 TextFormField(
```

```
171     textEditingController: mobileController ,
172
173     keyboardType: TextInputType.phone ,
174
175     focusNode: _mobileFocus ,
176
177     onChanged: (_) => _inputChangeListner() ,
178
179   ),
180
181   const SizedBox(
182     height: 8.0 ,
183
184   ),
185
186   const TextWidget(
187
188     title: passwordText ,
189
190     fontSize: 19.0 ,
191
192   ),
193
194   const SizedBox(
195     height: 4.0 ,
196
197   ),
198
199   TextFormField(
200
201     textEditingController: passwordController ,
202
203     focusNode: _passwordFocus ,
204
205     onChanged: (_) => _inputChangeListner() ,
206
207     isObsecure: isObsecure ,
208
209     suffixIcon: InkWell(
210
211       onTap: () => createRestuarantBloc.obsecureStatus(status:
212
213         false) ,
214
215       child: isObsecure
216
217         ? const Icon(Icons.visibility_off)
218
219         : InkWell(
220
221           onTap: () =>
222
223             createRestuarantBloc.obsecureStatus(status:
224
225               true) ,
226
227             child: const Icon(Icons.visibility)) ,
228
229
230   ),
```

```
199      ) ,  
200      ),  
201      const SizedBox(  
202          height: 15.0,  
203      ),  
204      Center(  
205          child: InkWell(  
206              onTap: _createAccountPressHandler,  
207              child: Text(  
208                  "Don't have Account ? Click Here !!",  
209                  textAlign: TextAlign.center,  
210                  style: GoogleFonts.fresca(  
211                      fontWeight: FontWeight.w500, fontSize: 18.0),  
212                  ),  
213                  ),  
214                  ),  
215                  Padding(  
216                      padding: const EdgeInsets.symmetric(vertical: 15.0),  
217                      child: Center(  
218                          child: InkWell(  
219                              onTap: _changePasswordPressHandler,  
220                              child: Text(  
221                                  "Forgot Password?",  
222                                  textAlign: TextAlign.center,  
223                                  style: GoogleFonts.fresca(  
224                                      fontWeight: FontWeight.w500, fontSize: 18.0),  
225                                      ),  
226                                      ),  
227                                      ),  
228                  )
```

```
229     ] ,  
230     ) ,  
231     );  
232 }  
233 }
```

Listing 7.1: Admin Login

APPENDIX 2

```
1 import 'package:flutter/material.dart';
2
3 import 'package:smartrestuarantmanager/bloc/bloc_exports.dart';
4 import 'package:smartrestuarantmanager/screens/screen_exports.dart';
5 import 'package:smartrestuarantmanager/utilis/utilis_export.dart';
6 import 'package:smartrestuarantmanager/widgets/widgets_export.dart';
7
8 class ManagerLogin extends StatefulWidget {
9   const ManagerLogin({super.key});
10
11   @override
12   State<ManagerLogin> createState() => _ManagerLoginState();
13 }
14
15 class _ManagerLoginState extends State<ManagerLogin> {
16   final TextEditingController _managerIDController =
17     TextEditingController();
18   final FocusNode _mangerFocus = FocusNode();
19   bool isEnabledButton = false;
20   String managerDetails = "";
21
22   @override
23   void initState() {
```

```
23     super.initState();  
24  
25     managerAuthBloc.managerDetailsStream.listen((event) {  
26  
27         if (mounted) {  
28             setState(() => managerDetails = event);  
29         }  
30     });  
31  
32     managerAuthBloc.toastMessageStream.listen((event) {  
33         if (mounted) {  
34             event  
35                 ? Navigator.pushReplacement(  
36                     context,  
37                     MaterialPageRoute(  
38                         builder: (context) =>  
39                         ManagerDashBoard(managerId: managerDetails),  
40                     ))  
41                 : ToastMessage.showToast(message: invalidManagerIdText);  
42         }  
43     });  
44  
45     managerAuthBloc.enableButtonStream.listen((event) {  
46         if (mounted) {  
47             setState(() => isEnabledButton = event);  
48         }  
49     });  
50  
51     void _loginPressHandler() {  
52         managerAuthBloc.checkManagerIdCredential(  
53             managerId: _managerIDController.text,  
54         );  
55     }
```

```
53
54     void _inputChangeListener() {
55
56         managerAuthBloc.inputChangeListener(managerId:
57
58             _managerIDController.text);
59
60     }
61
62
63     @override
64     Widget build(BuildContext context) {
65
66         return SafeArea(
67
68             child: Scaffold(
69
70                 floatingActionButtonLocation:
71
72                     FloatingActionButtonLocation.centerFloat,
73
74                 body: _loginForm(),
75
76                 floatingActionButton: _loginButton(),
77
78             ),
79
80         );
81
82
83     }
84
85
86     Widget _loginButton() {
87
88         return Padding(
89
90             padding: const EdgeInsets.symmetric(horizontal: 25.0, vertical:
91
92                 10.0),
93
94             child: SizedBox(
95
96                 child: CustomButton(
97
98                     title: loginText,
99
100                    function: isEnabledButton ? _loginPressHandler : null,
101
102                    loadingStream: managerAuthBloc.enableButtonLoadingStream,
103
104                    isCenterAvailable: false,
105
106                ),
107
108            ),
109
110        );
111
112    }
```

```
80      );
81  }
82
83  Widget _loginForm() {
84    return Padding(
85      padding: const EdgeInsets.symmetric(horizontal: 25.0),
86      child: Column(
87        mainAxisAlignment: MainAxisAlignment.start,
88        mainAxisSize: MainAxisSize.center,
89        children: [
90          const TextWidget(title: loginText, fontSize: 27.0),
91          const SizedBox(height: 18.0),
92          const TextWidget(title: managerIDText, fontSize: 19.0),
93          const SizedBox(height: 8.0),
94          TextFormField(
95            textEditingController: _managerIDController,
96            focusNode: _mangerFocus,
97            onChanged: (_) => _inputChangeListener(),
98          ),
99        ],
100      ),
101    );
102  }
103 }
```

Listing 7.2: Manager Login

```
1 import 'package:flutter/material.dart';
2
3 import 'package:smartrestuarantmanager/utilis/utilis_export.dart';
4 import 'package:smartrestuarantmanager/widgets/widgets_export.dart';
5
```

```
6 import 'package:flutter_svg/flutter_svg.dart';
7
8 class ManageOrders extends StatefulWidget {
9   const ManageOrders({super.key});
10
11   @override
12   State<ManageOrders> createState() => _ManageOrdersState();
13 }
14
15 class _ManageOrdersState extends State<ManageOrders> {
16   @override
17   Widget build(BuildContext context) {
18     return Scaffold(
19       backgroundColor: whiteColor,
20       appBar: const PreferredSize(
21         preferredSize: Size.fromHeight(56),
22         child: AppBarWidget(
23           appBarSubTitleText: moreText,
24           appBarTitleText: ordersText,
25         ),
26       ),
27       body: _buildEmptyStateUI(),
28     );
29   }
30
31   Widget _buildEmptyStateUI() {
32     return Column(
33       mainAxisAlignment: MainAxisAlignment.center,
34       crossAxisAlignment: CrossAxisAlignment.center,
35       children: [

```

```
36     Semantics(
37         container: true,
38         label: "speaker",
39         child: SvgPicture.asset("assets/images/svg/comming_soon.svg"),
40     ),
41     const SizedBox(height: 18),
42     _buildTextUI(text: '$comingSoonText!', fontSize: 17),
43     const SizedBox(height: 5),
44     _buildTextUI(
45         text: organizationInboxDescriptionText,
46         fontSize: 12,
47     ),
48     const SizedBox(height: 36),
49     _buildIconTextUI(
50         title: filterNotificationsText,
51         icon: Icons.filter_alt_outlined,
52     ),
53     const SizedBox(height: 12.5),
54     _buildIconTextUI(
55         title: getNotificationOrganizationText,
56         icon: Icons.alarm_on_outlined,
57     ),
58     const SizedBox(height: 12.5),
59     _buildIconTextUI(
60         title: sortingCapabilitiesText,
61         icon: Icons.sort_outlined,
62     )
63 ],
64 );
65 }
```

```
66
67     Widget _buildTextUI({required String text, required double fontSize}) {
68
69         return Text(
70
71             text,
72             textAlign: TextAlign.center,
73             style: TextStyle(
74
75                 fontWeight: FontWeight.w400,
76
77                 fontSize: fontSize,
78
79                 height: 1.3,
80
81                 color: descriptionColor,
82
83             ),
84
85         );
86
87     }
88
89
90     Widget _buildIconTextUI({required String title, required IconData
91
92         icon}) {
93
94
95         return Row(
96
97             mainAxisAlignment: MainAxisAlignment.center,
98
99             children: [
100
101                 Icon(
102
103                     icon,
104
105                     color: descriptionColor,
106
107                     size: 16.0,
108
109                 ),
110
111                 const SizedBox(width: 5),
112
113                 _buildTextUI(text: title, fontSize: 12)
114
115             ],
116
117         );
118
119     }
120 }
```

94 }

Listing 7.3: Managers Order Page

```
1 import 'package:flutter/material.dart';
2 import 'package:smartrestuarantmanager/bloc/bloc_exports.dart';
3
4 import 'package:smartrestuarantmanager/utilis/utilis_export.dart';
5 import 'package:smartrestuarantmanager/widgets/widgets_export.dart';
6
7 import 'package:flutter_svg/flutter_svg.dart';
8 import 'package:google_fonts/google_fonts.dart';
9
10 class AddDelTable extends StatefulWidget {
11   final String restuarantId;
12
13   const AddDelTable({required this.restuarantId, super.key});
14
15   @override
16   State<AddDelTable> createState() => _AddDelTableState();
17 }
18
19 class _AddDelTableState extends State<AddDelTable> {
20   int count = 0;
21   bool enableButton = false;
22   bool disableUpdationTables = true;
23
24   @override
25   void initState() {
26     super.initState();
27     tablesBloc.getTableValue(restuarantId: widget.restuarantId);
28     tablesBloc.checkRestuarantTablesStatus(restuarantId:
29       widget.restuarantId);
```

```
28     tablesBloc.counterStream.listen((event) {
29
30         if (mounted) {
31
32             setState(() => count = event);
33
34         }
35
36         setState(() => enableButton = event);
37
38     });
39
40     tablesBloc.checkTablesStatus.listen((event) {
41
42         if (mounted) {
43
44             setState(() => disableUpdationTables = event);
45
46         }
47
48     });
49
50 }
51
52 @override
53 Widget build(BuildContext context) {
54     return Wrap(
55         children: [
56
57         Padding(
58             padding: EdgeInsets.only(
59                 top: 10,
60                 left: 10,
61                 right: 10,
62                 bottom: 10),
63
64             child: Column(
65                 mainAxisAlignment: MainAxisAlignment.start,
66                 crossAxisAlignment: CrossAxisAlignment.start,
67                 children: [
68
69                 Container(
70                     width: 100,
71                     height: 100,
72                     margin: EdgeInsets.all(10),
73                     decoration: BoxDecoration(
74                         color: Colors.white,
75                         border: Border.all(
76                             color: Colors.black,
77                             width: 1),
78                         shape: BoxShape.rectangle),
79
80                     child: Center(
81                         child: Text(
82                             "Table Count: " + count.toString(),
83                             style: TextStyle(
84                                 color: Colors.black,
85                                 fontSize: 16,
86                                 fontWeight: FontWeight.w500),
87                         ),
88                     ),
89                 ),
90
91                 Container(
92                     width: 100,
93                     height: 100,
94                     margin: EdgeInsets.all(10),
95                     decoration: BoxDecoration(
96                         color: Colors.white,
97                         border: Border.all(
98                             color: Colors.black,
99                             width: 1),
100                        shape: BoxShape.rectangle),
101
102                     child: Center(
103                         child: Text(
104                             "Enable Button: " + enableButton.toString(),
105                             style: TextStyle(
106                                 color: Colors.black,
107                                 fontSize: 16,
108                                 fontWeight: FontWeight.w500),
109                         ),
110                     ),
111                 ),
112             ],
113         ),
114     ),
115 
```

```
58             left: 15.0,
59             right: 15.0,
60             bottom: MediaQuery.of(context).viewInsets.bottom + 20,
61         ),
62         child: Column(
63             children: [
64                 const BottomSheetHeader(headerTitle: tablesText),
65                 StreamBuilder<int>(
66                     stream: tablesBloc.counterStream,
67                     builder: (context, snapshot) {
68                         if (snapshot.hasData) {
69                             count = snapshot.data!;
70                         }
71                         return Column(
72                             children: [
73                                 const SizedBox(height: 24.0),
74                                 _buildAddTablesUI(),
75                                 const SizedBox(height: 14.0),
76                                 disableUpdationTables ? const SizedBox()
77                                     : CustomButton(
78                                         title: addText,
79                                         function: enableButton ? _updateTableCount :
80                                             null,
81                                         loadingStream:
82                                         tablesBloc.enableButtonLoadingStream,
83                                         color: redColor,
84                                         )
85                         ],
86                     );
87                 } else if (snapshot.hasError) {}
88                 return const ShimmerEffect(
89             
```

```
85             slideLineHeight: 4.0,
86             containerHeight: 90.0,
87             numberOfWorkers: 1,
88           );
89         })
90       ],
91     ),
92   ],
93 ],
94 );
95 }
96
97 Widget _buildBoxUI({required Widget child, required Color color}) {
98   return Container(
99     width: MediaQuery.of(context).size.width,
100    constraints: const BoxConstraints(maxHeight: double.infinity),
101    padding: const EdgeInsets.symmetric(
102      vertical: 8.0,
103      horizontal: 20.0,
104    ),
105    decoration: BoxDecoration(
106      color: color,
107      borderRadius: BorderRadius.circular(5.0),
108    ),
109    child: child,
110  );
111 }
112
113 Widget _buildAddTablesUI() {
114   return _buildBoxUI(
```

```
115     child: Row(
116       mainAxisAlignment: MainAxisAlignment.spaceBetween,
117       children: [
118         Column(
119           crossAxisAlignment: CrossAxisAlignment.start,
120           children: [
121             SvgPicture.asset(
122               "assets/images/svg/table.svg",
123               color: descriptionColor,
124             ),
125             const SizedBox(height: 4.0),
126             Text(
127               totalTablesText,
128               style: GoogleFonts.philosopher(
129                 fontWeight: FontWeight.w400,
130                 fontSize: 23.0,
131                 color: blackColor,
132               ),
133             ),
134           ],
135         ),
136         _buildCounterButtonUI()
137       ],
138     ),
139     color: selectedRadioTileColor,
140   );
141 }
142
143 Widget _buildCounterButtonUI() {
144   return Row(
```

```
145     children: [
146       disableUpdationTables ? const SizedBox() : IconButton(
147         onPressed: () => tablesBloc.changeCounterState(
148           counterStateName: CounterEnum.decrement,
149           actualCount: count,
150         ),
151         icon: const Icon(
152           Icons.remove_circle_outline_outlined,
153           color: descriptionColor,
154         ),
155       ),
156       Text(
157         count.toString(),
158         style: GoogleFonts.gidugu(
159           fontWeight: FontWeight.w400,
160           fontSize: 28.0,
161           color: descriptionColor,
162         ),
163       ),
164       disableUpdationTables ? const SizedBox() : IconButton(
165         onPressed: () => tablesBloc.changeCounterState(
166           counterStateName: CounterEnum.increment,
167           actualCount: count,
168         ),
169         icon: const Icon(
170           Icons.add_circle_outline,
171           color: descriptionColor,
172         ),
173       ),
174     ],
```

```
175      );  
176  }  
177 }
```

Listing 7.4: Add or del tables

APPENDIX 3

```
1 import 'package:flutter/material.dart';
2 import 'package:go_router/go_router.dart';
3 import 'package:google_fonts/google_fonts.dart';
4 import 'package:smartrestuarant/model/model_exports.dart';
5 import 'package:smartrestuarant/screens/screen_exports.dart';
6 import 'package:smartrestuarant/utilis/utilis_export.dart';
7 import 'package:smartrestuarant/widgets/widget_exports.dart';
8
9 class RestuarantDetails extends StatefulWidget {
10   final CreateAccount restuarantDetails;
11   final String userEmail;
12
13   const RestuarantDetails({
14     required this.restuarantDetails,
15     required this.userEmail,
16     super.key,
17   });
18
19   @override
20   State<RestuarantDetails> createState() => _RestuarantDetailsState();
21 }
22
23 class _RestuarantDetailsState extends State<RestuarantDetails> {
```

```
24  @override
25  void initState() {
26      super.initState();
27  }
28
29  @override
30  Widget build(BuildContext context) {
31      return Padding(
32          padding: const EdgeInsets.symmetric(vertical: 10.0, horizontal:
33              18.0),
34          child: Wrap(
35              children: [
36                  Row(
37                      mainAxisAlignment: MainAxisAlignment.spaceBetween,
38                      children: [
39                          Expanded(
40                              child: Column(
41                                  crossAxisAlignment: CrossAxisAlignment.start,
42                                  children: [
43                                      Text(
44                                          widget.restaurantDetails.restaurantName,
45                                          style: GoogleFonts.roboto(
46                                              fontWeight: FontWeight.w500,
47                                              fontSize: 20.0,
48                                          ),
49                                      overflow: TextOverflow.ellipsis,
50                                  ),
51                                      Text(
52                                          AppStrings.restaurantText,
```

```
53         style: GoogleFonts.roboto(
54             fontWeight: FontWeight.w500,
55             fontSize: 16.0,
56             color: AppColors.descriptionColor,
57         ),
58     )
59 ],
60 ),
61 ),
62 IconButton(
63     onPressed: () => context.pop(),
64     icon: const Icon(Icons.close_sharp),
65 )
66 ],
67 ),
68 _buildRestuarantDescription(),
69 ],
70 ),
71 );
72 }
73
74 Widget _buildRestuarantDescription() {
75     return Padding(
76         padding: const EdgeInsets.symmetric(vertical: 10.0),
77         child: Column(crossAxisAlignment: CrossAxisAlignment.start,
78             children: [
79                 Text(
80                     AppStrings.restuarantDescription,
81                     style: GoogleFonts.robotoSlab(
82                         fontWeight: FontWeight.w500, fontSize: 18.0),
```

```
82      ),
83      const SizedBox(
84        height: 8.0,
85      ),
86      Text(
87        '(RN) combines flavors & inspiration from the Far East & the
88        West to create what we think is the best Home to the
89        original (recipe), we specialize in pizzas, as well as
90        handmade appetizers, sandwiches, salads, and gluten
91        friendly, vegetarian, & vegan selections. Feel free to
92        indulge in a tiki drink or craft beer with our fantastic
93        unique cuisine. Cheers',
94        style: GoogleFonts.roboto(
95          fontWeight: FontWeight.w400,
96          fontSize: 15.0,
97          color: AppColors.descriptionColor,
98        ),
99      ),
100    ),
101  ],
102  Widget _bookTableButton() {
103    return Widgets.button(
104      callback: () {
105        context.pop();
```

```
106     Navigator.push(
107         context,
108         MaterialPageRoute(
109             builder: (context) => TablesScreen(
110                 restuarantId: widget.restuarantDetails.restuarantId,
111                 userEmail: widget.userEmail,
112             ),
113         ),
114     );
115 },
116     shape: RoundedRectangleBorder(borderRadius:
117         BorderRadius.circular(4)),
118     child: Text(
119         AppStrings.bookTableText,
120         style: GoogleFonts.roboto(
121             fontWeight: FontWeight.w500,
122             fontSize: 16.0,
123             color: Colors.white,
124         ),
125     ),
126 },
127 }
```

Listing 7.5: Restaurants details

```
1 import 'package:date_time_picker_widget/date_time_picker_widget.dart';
2 import 'package:flutter/material.dart';
3 import 'package:go_router/go_router.dart';
4 import 'package:google_fonts/google_fonts.dart';
5 // ignore: depend_on_referenced_packages
6 import 'package:intl/intl.dart';
```

```
7 import 'package:smartrestuarant/utilis/utilis_export.dart';
8 import 'package:smartrestuarant/widgets/widget_exports.dart';
9
10 class DateTimeScreen extends StatefulWidget {
11   const DateTimeScreen({super.key});
12
13   @override
14   State<DateTimeScreen> createState() => _DateTimeScreenState();
15 }
16
17 class _DateTimeScreenState extends State<DateTimeScreen> {
18   String? _d1;
19   String? _t1;
20
21   @override
22   Widget build(BuildContext context) {
23     return Theme(
24       data: Theme.of(context).copyWith(
25         primaryColor: AppColors.introButtonColor,
26         colorScheme: ColorScheme.fromSwatch()
27           .copyWith(secondary: AppColors.introButtonColor),
28     ),
29     child: Wrap(children: [_dateTimePicker()]);
30   }
31
32   Widget _dateTimePicker() {
33     final dt = DateTime.now();
34     return Padding(
35       padding: const EdgeInsets.symmetric(vertical: 10.0),
36       child: Column(
```

```
37     mainAxisAlignment: MainAxisAlignment.start,
38
39     children: [
40
41         CircleAvatar(
42             child: IconButton(
43                 onPressed: () => context.pop(), icon: const
44                     Icon(Icons.close)),
45
46         ),
47
48         DateTimePicker(
49             initialSelectedDate: dt,
50
51             startDate: dt.subtract(const Duration(days: 1)),
52
53             endDate: dt.add(const Duration(days: 360)),
54
55             startTime: DateTime(dt.year, dt.month, dt.day, 6),
56
57             endTime: DateTime(dt.year, dt.month, dt.day, 18),
58
59             timeInterval: const Duration(minutes: 15),
60
61             datePickerTitle: 'Select Date',
62
63             timePickerTitle: 'Select Time',
64
65             timeOutOfRangeError: 'Sorry shop is closed now',
66
67             is24h: false,
68
69             numberOfWeeksToDisplay: 4,
70
71             onDateChanged: (date) {
72
73                 setState(() {
74
75                     _d1 = DateFormat('dd MMM, yyyy').format(date);
76
77                 });
78
79             },
80
81             onTimeChanged: (time) {
82
83                 setState(() {
84
85                     _t1 = DateFormat('hh:mm:ss aa').format(time);
86
87                 });
88
89             },
90
91         ),
92
93     ),
```

```
66         const SizedBox(
67             height: 8.0,
68         ),
69         _bookTableButton()
70     ],
71 ),
72 );
73 }
74
75 Widget _bookTableButton() {
76     return Widgets.button(
77         callback: () {
78             context.pop();
79             context.pushNamed("menuscreen");
80         },
81         shape: RoundedRectangleBorder(borderRadius:
82             BorderRadius.circular(4)),
83         child: Text(
84             "Proceed",
85             style: GoogleFonts.rockSalt(
86                 fontWeight: FontWeight.w500, fontSize: 16.0, color:
87                 Colors.white),
88        ));
89 }
```

Listing 7.6: User Slot booking

```
1
2 import 'package:flutter/material.dart';
3 import 'package:flutter_svg/flutter_svg.dart';
4 import 'package:google_fonts/google_fonts.dart';
```

```
5 import 'package:smartrestuarant/utilis/app_colors.dart';
6 import 'package:smartrestuarant/widgets/widget_exports.dart';
7
8 class PaymentScreen extends StatefulWidget {
9     const PaymentScreen({super.key});
10
11     @override
12     State<PaymentScreen> createState() => _PaymentScreenState();
13 }
14
15 class _PaymentScreenState extends State<PaymentScreen> {
16     @override
17     Widget build(BuildContext context) {
18         return Scaffold(
19             body: SafeArea(
20                 child: Column(
21                     mainAxisAlignment: MainAxisAlignment.center,
22                     crossAxisAlignment: CrossAxisAlignment.center,
23                     children: [
24                         Image.asset("assets/images/png/payment.png"),
25                         Text(
26                             "payment completed",
27                             style: GoogleFonts.gidugu(
28                                 fontSize: 40.0, fontWeight: FontWeight.w400),
29                         ),
30                         Row(
31                             mainAxisAlignment: MainAxisAlignment.center,
32                             children: [
33                             Column(
34                                 crossAxisAlignment: CrossAxisAlignment.start,
```

```
35         children: [
36             Row(
37                 children: [
38                     Image.asset(
39                         "assets/images/png/store.png",
40                         color: AppColors.greyColor,
41                         width: 16.0,
42                         height: 18.0,
43                     ),
44                     const SizedBox(width: 8.0),
45                     _text('Truffles Koramangala'),
46                 ],
47             ),
48             const SizedBox(
49                 height: 10.0,
50             ),
51             Row(
52                 children: [
53                     SvgPicture.asset(
54                         "assets/images/svg/table.svg",
55                         color: AppColors.greyColor,
56                         width: 16.0,
57                         height: 18.0,
58                     ),
59                     const SizedBox(width: 8.0),
60                     _text('Table 5'),
61                 ],
62             ),
63             const SizedBox(
64                 height: 10.0,
```

```
65      ) ,
66      Row(
67        children: [
68          Image.asset(
69            "assets/images/png/calendar.png",
70            color: AppColors.greyColor,
71            width: 16.0,
72            height: 18.0,
73          ),
74          const SizedBox(width: 8.0,),
75          _text('09 NOV 2022'),
76        ],
77      ),
78      const SizedBox(
79        height: 10.0,
80      ),
81      Row(
82        children: [
83          Image.asset(
84            "assets/images/png/time.png",
85            color: AppColors.greyColor,
86            width: 16.0,
87            height: 18.0,
88          ),
89          const SizedBox(width: 8.0,),
90          _text('09:00 AM'),
91        ],
92      ),
93    ],
94  )
```

```
95        ] ,  
96  
97        const SizedBox(height: 40.0,) ,  
98        Widgets.button(  
99            width: 201.0 ,  
100           callback: (){},  
101           child: Text("continue shopping",style: GoogleFonts.gidugu(  
102               fontWeight: FontWeight.w400 ,  
103               fontSize: 25.0  
104           ),),  
105           color: AppColors.whiteColor ,  
106           shape: RoundedRectangleBorder(borderRadius:  
107               BorderRadius.circular(8.0)),  
108       )  
109     ],  
110   );  
111 }  
112  
113 Text _text(String text) {  
114     return Text(  
115         text ,  
116         style: GoogleFonts.fresca(  
117             fontWeight: FontWeight.w400 ,  
118             fontSize: 22.0 ,  
119             color: AppColors.greyColor),  
120     );  
121 }  
122  
123
```

124 }

Listing 7.7: Payment

APPENDIX4

(Literature Survey Publication and Certificates)

SMART RESTAURANT MANAGEMENT SYSTEM

S.Phani Kumar¹, N.Pavithra², B.Sarika³, P.Hemanth⁴, B.Sai Kiran⁵

Associate Professor ¹, Student^(2,3,4,5)

Information Technology,

Sasi Institute Of Technology And Engineering

ABSTRACT

Now a days technology is increasing rapidly in many areas, people are always craving for faster and easier day to day tasks. Enhancing business practices in the restaurant and hotel sectors of the food industry will help to boost client satisfaction, cut down on user wait times for things like menus and tables, and make it easier for managers and chefs to carry out their duties. Applications for smart restaurant systems will lessen the manual errors that are always made in manual restaurants, such as incorrect food serving and billing, etc. The specifics of the restaurant are displayed in the smart restaurant system. Here, customers may login to reserve a table, choose a menu, and place an order without the help of a waiter. The manager has the ability to update the menu status, table status, and other information on the restaurant manager panel.

KEYWORDS: Mobile application, Restaurant management, Flutter, Nodejs, Menu, Order, Booking, Payment.

1. INTRODUCTION

A smart restaurant system is one that has been put up to advance the food sector by doing away with some of the drawbacks of conventional systems. By automating the process, the developed approach will save people and the world time and energy while looking for the best restaurants in their selected locations and will reduce the amount of time spent in restaurants.

The methods that are followed by restaurants nowadays are pen and paper- based methods. In this method, firstly they will print a menu of the restaurant using paper that is paper-based menu card. Every time they have to print new menu cards if they want to do any changes to the menu, like based the food available in the restaurants, offering some discounts and special offers or adding some new items and even if the menu card is printed wrongly, there is no chance to change that card, instead of

printing a new card. It is also a time waste for waiters to wait until the customers, select the desired food. Taking orders using pen and paper is another process used in the traditional system. Time is wasted greatly when orders are taken on paper and then sent to the kitchen. Sometimes the handwriting of the waiter is not understood by the kitchen department. Therefore, in this instance, making the incorrect meal and delivering it to customers leads to a great deal of customer unhappiness. In some seasons when the customer visiting hotels or restaurants will also lead to customer dissatisfaction as the waiters cannot able to provide immediate service to all the customers, giving them the wrong orders and sometimes it takes a lot of delays to deliver food even after customer waited a long time. In many order receipts, the papers on which the orders are taken can also be lost. As a result, numerous complaints regarding the hotel's services are made at the management desk. After the food is

consumed food by customers, the bill receipt will send is also through paper. Employing more waiters to satisfy customers in crowd situations will also be a drawback as there is no need for more waiters in low times and leads to loss of money in form of salary. Even taking orders, printing menu cards, and creating bill receipts use a lot of paper. Due to the numerous manual errors made by the current restaurant system, both customers and management are greatly inconvenienced. In an effort to improve the current system and help management deliver better service, many research have been carried out. This has the dual benefits of reducing wait times for both customers and restaurant personnel while also increasing customer satisfaction. People are rapidly moving towards a smarter world, when used in conjunction with smart cities, smart classrooms, and smartphones. At present, information and communication technology has been brought to several business models to make the operation more convenient and effective. In almost every single nation and culture in the world today, restaurants hold an important role as a hub for social interactions. Not only do they provide food, drinks, and sustenance to individuals, but they also function as a place where groups of people come together to socialize, connect, and share great experiences. Restaurants are also extremely lucrative businesses, as the global restaurant industry grew by 6.2percent in 2014 to reach a value of 2,737.1 dollar billion, and it is forecasts to grow to a value of 3,805.8 dollar billion by 2019. However, despite restaurants holding such an important role in our society and despite them earning high revenues, the restaurant industry has had few changes to how it has been operating in the past century. The restaurant industry still has many inefficiencies that are unaddressed despite all of the great technological developments that made in recent decades with the rise of the internet, mobile smart devices, and cloud computing. This is because the restaurant industry has not fully

exploited the full capabilities of these technologies.

2. SCOPE OF THE PROJECT

The scope of this proposed smart restaurant system is categorized into three categories:

2.1 Guest - Customer

1. Customer will find the restaurant in his location with ratings.
2. Customers can able to check the availability of food in the restaurant.
3. Customers can view the menu on the app and order the menu items they desire, along with the cost, by choosing the quantity.

2.2 Staff

1. Staff will serve the food to the customers which they order based on the table number.
2. Staff will also be able to know whether the guests order the food or not.

2.3 Restaurants/Cafe/Stall owners

1. Admin can able to add a new menu or can delete the menu if not available and updating the food menu with changed prices.
2. Admin can view the report of data.

3. LITERATURE REVIEW

Vindya Liyanage, Achini Ekanayake et al[1] By displaying a real-time map of the restaurant, the status of the reservations, and recommendations for unusual cuisine, this app enables the consumer to quickly select the menu.

Nersine koubai et al[2] To keep track of previous orders so that we can suggest and provide him with similar foods for his upcoming orders.

Akash patil, Rinkesh kalani et al [3] The consumer will find it easy to place an order on their own thanks to the dynamic user interface on the digital menu.

Kalpesh V. Joshi, Akshay P. Dhawale et al[4] "The order was placed after being confirmed at their respective tables. The robots will deliver the order to the appropriate tables.

Pyla Sai kiran,Yadla Abhiram et al[5] "The order was placed after being confirmed at their respective tables. The robots will deliver the order to the appropriate tables.

Marc Bolanos,Petia radeva et al[6] By anticipating the arrangement of various meals on a tray picture, it solves the issue of automatic food analysis in the setting of canteens and restaurants.

Pranav Jain et al[7] Scan and eat the food they desire. Food goods are identified and paid for utilising RFID tags, which are attached to each food item.

Tehreem Hussain et al[8] "The research's objective is to fully automate food ordering utilising digital technology. The idea in this post is to build a "SMART TABLE" using both an Android application and a web application."

Serena Filipelli et al[9] The article uses keyword research to identify areas for system enhancement in the food industry.

Utkarsh Ravekar et al [10] The kitchen workers will be shown the ordered food products on a screen in the kitchen, which can also double as the neighbourhood server.

Biil J.Gregorash et al[11] Customers who frequently complain about booking tables that aren't available while they're even booked will experience less of this.

Shreya Umap et al[12] In this study, a user-friendly matrix keypad-based menu card with an entirely automatic system for placing menu orders is suggested to replace the paper-based menu.

Fatim Jaffri et al [13] In addition to providing fully functional management functions, Smartable, which is composed of four Android applications designed for restaurants, also employs data mining to identify trends.

W. Gawande et al [14] "It aims to improve the restaurant industry while also infusing basic scientific concepts into the menu. As a result, a future-proof restaurant management system keeps track of everything that occurs there, and everything is permission-based to deter theft."

Madhu Vij et al[15] By developing and testing a model of attribute conversion for mobile apps, this study aims to analyse online food aggregators (OFAs) and determine how these features influence a consumer's purchase decision and ultimately lead to conversion.

Shivani Bhavsar et al[16] "This technique was developed to make running a restaurant more efficient. In a restaurant, clients from all over the world can purchase prepared meals. To provide customers and restaurants with service options, this technique was developed."

Abitha Abbas et al[17]"The method recommended in this study entails scanning a QR code to order food from a restaurant. Customers can scan the QR code on the table using their mobile devices to open the current menu".

Yogesh Gite et al[18] With the use of this system, customers will be able to place orders via an Android app, and the orders will subsequently be presented in the kitchen.

Madhuri Agrawat et al[19] The system is put in place to lessen manual labour and improve accuracy in a restaurant. Each order corresponds to a certain seat at the table. Additionally, it is simple to exchange items across the entire table, move or modify them, make notes, and figure out the cost in real time.

Mr. Karthick Panneer selvam[20] To effectively manage all food production and complete customer orders, this system has a

integrated POS system includes kitchen management and order management features.

4.OBJECTIVE

The primary objective of this restaurant management system is to create an efficient system for booking and managing restaurants. Customers can access it with ease. The restaurant management initiative seeks to minimize workload and paperwork. Customer and manager wait times will be cut using this technology, and restaurant manual errors will also be decreased.

5.METHODOLOGY

Architecture Diagram

The conceptual model that outlines a system's structure, behaviour, and other aspects is called a system architecture. A formal description and representation of a system that is set up to facilitate analysis of its structures and behaviours is called an architecture description. A system architecture can be made up of designed subsystems and system components that will cooperate to implement the whole system. The suggested system's architecture is shown in the diagram below. It demonstrates step-by-step how the process was carried out in the system. How the user accesses the

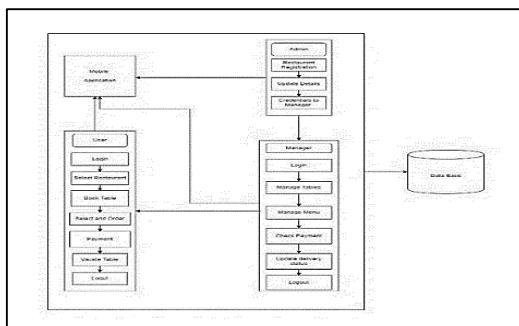


fig: Architecture Diagram

6. RELATED WORK

S.NO	TITLE NAME	AUTHOR	OVERVIEW	TECHNOLOGIES	LIMITATIONS
1.	Foody-Smart Restaurant	Vindya Liyanage et al	This app shows the real time map of the	IOT sensor, cloud database, server , NLP	Furthermore, it is expected that for any person who plans to

application by registering into the application and performing all required restaurant activities.

USER

- Go to the website
- Login to the application
- Select location and restaurant
- Find the availability of tables for reservation
 - Choose table
 - Select menu
 - Select categories
 - Payment
 - Vacate table

ADMIN

- Go to the website
- Login to the application
- Update the user details

MANAGER

- Register the restaurant in the website
- Sign in to the application
- Update the restaurant status and details

REQUIREMENTS

Hardware requirements

- Processor – i3/i5
- Hard Disk – 4 GB
- Memory – 1/2/4/8/16GB RAM

Software requirements

- Back-end : Firebase
- Front-end : Flutter

	Management and Ordering System		restaurant. Current reservation status and unique food recommendation and allow the customer to choose the menu according to their taste in a minimum time.		build a similar system or any other real-time system, results of this research will be an aid and will provide insight on the performance, accuracy and reliability level that can be expected with the combination of tools, technologies and business requirements
2.	My restaurant:A smart restaurant with recommendation System	Nersine koubai et al	To keep track of the client order history in order to suggest and recommend to him similar dishes for his next orders.	PPDL, IOT, SENSOR, DS	we intend to design a communication system to enable the communication between the incompatible objects of the network
3.	Smart Restaurant System Using Android	Akash patil et al	The digital menu will provide interactive user interface with which user will easily place his order by itself.	Java,UI	In future, the ordering system can also be made to be speech recognize ordering system.
4.	Automated Restaurant Service Management System	Kalpesh V. Joshi et al	The order once placed and confirmed at their respective tables. The order will be delivered by the robots to their respective tables	Android Studio, XML, UserInterface	We are planning to introduce online billing feature into the app so as to make payment easy.
5.	Smart Restaurant:From Order to Delivery, and Security	Pyla Sai kiran et al	Orders are displayed to chef in the kitchen .After the preparation a waiter robot will deliver the food of the respective table & serve the food reduces human error.	UI, server, sensors	This robot can also be involved in room service also so that it can carry all the basic items which are required by the customer.
6.	Semantic Food Detection for Smart restaurant	Marc Bolanos et al	It addresses the issue of automatic food	NLP, CNN Bayesian model	Our ongoing study is to fully automate self-service

			analysis in the context of canteens and restaurants by foreseeing various foods to be placed on a tray picture.		invoicing by integrating restaurant menus based on geolocation, as well as semantic ingredient detection for food.
7.	A review paper on Smart Restaurant Ordering System	Nikhitha Deshpand et al	The main aim of project is to automate food ordering and automate billing done	Arduino Mega2560, Resistive touch screen, TFT Display, RFID.	This project takes the order from the customer & lets him pay the bill without any human intervention.
8.	Smart Restaurant System Using RFID Technology Pranav Jain	Pranav Jain et al	Scan and consume the food they want. Each food item is marked with an RFID tag, and food products are identified and paid for using RFID tags.	Raspberry PI, Arduino, Database, RFID reader, RFID tag.	It improves the system by endorsing cashless transactions and streamlining the otherwise chaotic restaurant environment
9.	Digital Ordering System Or Restaurant Using Android	Tehreem Hussain et al	The research's focus is on digital meal ordering, where the entire process is automated. This paper suggests a "SMART TABLE" that is built on an Android app and a web app.	Database, Admin panel, Dinner app, Reservation app, Waiter app.	The restaurant automation is new concept which is bringing revolutionary impact. SMART TABLE is an online system which starts with order taking and ends up with receipt generation through automation system.
10.	A Study on IOT applications in the food Industry Using Keyword Analysis	Serena Filipelli et al	Through keyword analysis, the article seeks to uncover research areas related to system development in the food industry.	IOT sensors, CPS technologies.	Futher research using less stringent inclusion criteria could lead to different results.
11.	Human assisted restaurant management system	Prof. Pranita et al	It strives to enhance restaurant business while also incorporating	Android Studio, XAMPP server, Android SDK.	Therefore, everything that happens inside the restaurant is tracked by a future-proof restaurant

			the fundamentals of science into the menu.		management system, and everything is permission-based to prevent theft.
12.	QR code based Smart dining system	Syed Ramees et al.	The approach suggested in this research involves reading the QR code to place an order for food at a restaurant. Customers can use their cellphones to open the current menu by scanning the QR code that is placed on the table.	JavaScript, PHP, MySQL, and Android studio	To simplify the ordering and bill payment systems to minimize the workload of the restaurant and hotel owners.
13.	Digital Restaurant	Vishal Gupta et al	This system will allow orders to be pulled from a programme clients have installed on their Android devices, and the order will then be displayed in the kitchen.	Conveyor belt, UNO Arduino R3, Adapter, Motors, Android Visual Studio, Google Firebase,	Food ordering application is presented with features of Wireless ordering system.
14.	Smart Food Quality Testing and Ordering System Using at Mega328 in Restaurants	D.Priyanka et al.	The main goal of this paper is to draw in customers and also adds to the efficiency of maintaining the restaurant's ordering and billing sections.	Arduino UNO, Different Sensors,	This technology will increase income by gradually replacing the usual paper menu for the restaurant with a keypad menu.
15	Data Centric Smart Restaurant Management System	Utkarsh Ravekar et al	On a kitchen screen, which can also act as the local server, the ordered food products will be shown to the kitchen workers.	Recommender app, Food ordering app, NLP, Cloud based server, Zigbee.	Our proposed framework thus aims to increases the efficiency of a restaurant by reducing the manual labour needed in taking the orders and at the same time providing personalized

					services to each customer.
16.	Implementation of Smart Restaurant Using Smart Phone Application	Kajal Choudhary et al	The meals that were ordered will be shown to the cooking staff on a kitchen screen, it also functions as neighbourhood server. The customer.	RFD , Zigbee, FFD, RFID, Android, Customer tablet.	The concept can be used to keep track of menu orders placed at tables and to request that they be transmitted straight to a predetermined website for invoicing purposes.
17.	Technology at the dinner table: Ordering food online through mobile apps	Anuj Pal Kapoor et al	This system was created to streamline a restaurant's daily operations. Restaurants are type of business that provides ready-made food to customers all over the world. This method was created to give customers and restaurants service options.	Zomato app, Paytm app, Mobile apps, Digital technology, OFA	Future studies should do a field replication of this study and use complete and actual online purchase transactions, including all phases of online food ordering, to measure conversion.
18.	Smart Menu Ordering System In Restaurant	Shreya U et al	The paper-based menu is replaced with a user-friendly menu card with a matrix keypad in this essay's proposal of a completely automated menu ordering system.	Paper based menu card, Self service food ordering technology, Arduino UNO, NRF, Matrix keypad, LCD.	This technology lowers operating costs by simulating the absence of a waiter, as well as by reducing manual service provided by waiters and serving employees and obviating human error.

19.	SMARTABLE - Digital Ordering System for Restaurant Using Android	Tehreefn Qamar et al	Restaurant-specific, Smart Table comprises of four applications for Android devices that not only provide fully functional management functions but also present trends using data mining techniques.	Firebase Database, Mobile base frontend, Web base back end.	It is designed to enhance the performance of restaurant workers because it is very time-saving, convenient, efficient, and simple to use.
20.	Food Board: A Restaurant Management System	Mr. Karthick Panneerselvam et al	To effectively manage all food production and complete customer orders, This system includes integrated POS mechanism for managing orders tools and kitchen management.	Html, Css, JavaScript, Bootstrap, React JS, NodeJS	For correct handling of the orders that gives the client sufficient satisfaction and proper communication between the personnel, this system offers seamless integration to all the eateries.

7. CONCLUSION

The goal of our project Smart Restaurant System is to enhance the eating experience for customers while automating typical restaurant procedures including table reservations, menu selection, ordering, and paying. The manager may also oversee the entire process through a control panel. It will lessen the workload in restaurants, as well as the requirement for servers and paper waste. The user can locate the restaurant using this system, reserve a table, view the menu items, choose the menu of interest, and place an order using this system solely. He has the option of paying the bill offline or online. Using the manager panel, the manager may oversee the entire restaurant. This technology will shorten wait times for both customers and the

manager while also reducing manual errors in the restaurant. Customer satisfaction is improved by this system.

8. REFERENCES

- [1] Samantha Thelijjagoda, Vindya Liyanage, Achini Ekanayake, Hiranthi Premasiri Prabhushi & Munasinghe, Foody – Smart Restaurant Management and Ordering System”. 978-1-5386-5051-6/18/\$31.00 ©2018 IEEE
- [2] Fayçal M'hamed Bouyakoub and Nesrine Koubai.” MyRestaurant: A Smart Restaurant with a Recommendation System”. DOI:
<http://dx.doi.org/10.12785/ijcds/080206>.
- [3] Rinkesh Kalani, Sachin Shinde, Bhavesh Patil, Akash Patil, and Prof. S. M. Shedole Smart restaurant system using

android ,International Journal of Technical Research and Applications e-ISSN: 2320-8163, www.ijtra.com, Volume 5, Issue 3 (May-June, 2017), PP. 78-80.

[4] Aman Jain, Snehal Chauhan, Suraj Sarange, and Anish Hirlekar, Automated Restaurant Management System, International journal of innovative research in electrical, electronics instrumentation and control engineering, Vol. 4, Issue 5, May 2016.

[5] Gunturi Sai Krishna, Pyla Sai Kiran, Yadla Abhiram, Chigurupalli Akhil, and Bolli Navaneeth. Smart Restaurant: From Order to Delivery, and Security, International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056 Volume: 08 Issue: 05 | May 2021 www.irjet.net p-ISSN: 2395-0072

[6] Petia Radeva, Beatriz Remeseiro, Marc Bolanos, and Eduardo Aguilar Semantic Food Detection for Smart Restaurant, IEEE TRANSACTIONS ON MULTIMEDIA, VOL. 1, NO. 1, JANUARY 2020.

[7] Jaya Sudha, Pranav Jain,Smart restaurant system using RFID Technology, IEEE Xplore Part Number:CFP20K25-ART; ISBN:978-1-7281-4889-2.

[8] Tehreem Hussain et al , Smart table Digital Ordering System for Restaurant Using Android, 1st International Conference on Computational Sciences and Technologies, 10-12 April 2019 (INCCST'19), MUET Jamshoro.

[9] Serena Filippelli, Eleonora Bottania, and Barbara Bigliardi A study on IoT application in the Food Industry using Keywords Analysis.

[10] Shashank Singh, Utkarsh Ravekar, Data Centric Smart Restaurant Management System, International Research Journal of Engineering and

Technology (IRJET) e-ISSN: 2395-0056 Volume: 06 Issue: 07 | July 2019 www.irjet.net p-ISSN: 2395-0072.

[11] Bill J. Gregorash , Restaurant revenue management: apply reservation management, nf Technol Tourism (2016) 16:331–346 DOI 10.1007/s40558-016-0065-0.

[12] Manjusha Binekar, Prajakta Kshirsagar, Shiwani Surode, Shreya Umap Prof. Nakul Nagpal, Smart Menu Ordering System in Restaurant, © 2018 IJSRST | Volume 4 | Issue 7 | Print ISSN: 2395-6011 | Online ISSN: 2395-602X Themed Section: Science and Technology.

[13] Fatim Jaffri et al , Smartable Digital Ordering System for Restaurant Using Android, 1st International Conference on Computational Sciences and Technologies, 10-12 April 2019 (INCCST'19), MUET Jamshoro

[14] Kalyani W. Gawande et al , Human assisted restaurant management system, © 2020 IJCRT | Volume 8, Issue 2 February 2020 | ISSN: 2320-2882.

[15] Anuj Pal Kapoor Madhu Vij,Technology at the dinner table:Ordering food through mobile apps <https://doi.org/10.1016/j.jretconser.2018.04.001>.

[16] Kajal Choudhary, Shivani Bhavsar, Sharmin Mansuri, Ankita Saner, and Professor Shezad Shaikh Implementation of Smart Restaurant Using Smart Phone Application, International Advanced Research Journal in Science, Engineering and Technology Vol. 7, Issue 4, April 2020.

[17] Syed Ramees C. R., Sreerag M., Abitha Abbas, Urmila Pillai L. V., Anjaly A. J. QR based smart dining system, Vol 11, Issue 5,May/2020

ISSN NO:0377-9254.

[18] Yogesh Gite, Lovefaith Narang, Neha Gaddam, and Vishal Gupta, Digital Restaurant, International Research Journal of Engineering and Technology (IRJET), Volume: 07 Issue: 04 | Apr 2020,

[19] Siddharth Purohit, Nisarg Patel, Mukund Patel, Het Patel, Madhuri Agravat Next Generation Restaurant: A Restaurant Management System, © 2019 JETIR May 2019, Volume 6, Issue 5 www.jetir.org (ISSN-2349-5162)

[20] Anmol Pradhan, Nigam Bhatrai, Dhruba Hari Ranabhat, Ashish Gautam, and Mr. Karthick Panneerselvam, as well as Mr. Dilip Bastola, Food Board: A Restaurant Management System, International Journal for Research in Applied Science & Engineering Technology (IJRASET) ISSN: 2321-9653; IC Value: 45.98; SJ Volume 10 Issue V May 2022

GIS SCIENCE JOURNAL



An UGC-CARE Approved Group II Journal

ISSN NO : 18669-9391 / Website : www.gisscience.net/
Email : editorgsjournal@gmail.com

Certificate of Publication

Paper ID : GSJ/10050

This is to certify that the paper titled

SMART RESTAURANT MANAGEMENT SYSTEM

Authored by

S.Phani Kumar

From

Sasi Institute Of Technology And Engineering.

Has been published in

GIS SCIENCE JOURNAL Volume 10, Issue 3, March 2023.




M Palaniswami
Editor-in-chief
GISSCIENCE



GIS SCIENCE JOURNAL



An UGC-CARE Approved Group II Journal

ISSN NO : 18669-9391 / Website : www.gisscience.net/
Email : editorgsjournal@gmail.com

Certificate of Publication

Paper ID : GSJ/10050

This is to certify that the paper titled

SMART RESTAURANT MANAGEMENT SYSTEM

Authored by

N.Pavithra

From

Sasi Institute Of Technology And Engineering.

Has been published in

GIS SCIENCE JOURNAL Volume 10, Issue 3, March 2023.




M Palaniswami
Editor-in-chief
GISSCIENCE



GIS SCIENCE JOURNAL



An UGC-CARE Approved Group II Journal

ISSN NO : 18669-9391 / Website : www.gisscience.net/
Email : editorgsjournal@gmail.com

Certificate of Publication

Paper ID : GSJ/10050

This is to certify that the paper titled

SMART RESTAURANT MANAGEMENT SYSTEM

Authored by

B.Sarika

From

Sasi Institute Of Technology And Engineering.

Has been published in

GIS SCIENCE JOURNAL Volume 10, Issue 3, March 2023.




M Palaniswami
Editor-in-chief
GISSCIENCE



GIS SCIENCE JOURNAL



An UGC-CARE Approved Group II Journal

ISSN NO : 18669-9391 / Website : www.gisscience.net/
Email : editorgsjournal@gmail.com

Certificate of Publication

Paper ID : GSJ/10050

This is to certify that the paper titled

SMART RESTAURANT MANAGEMENT SYSTEM

Authored by

P.Hemanth

From

Sasi Institute Of Technology And Engineering.

Has been published in

GIS SCIENCE JOURNAL Volume 10, Issue 3, March 2023.




M Palaniswami
Editor-in-chief
GISSCIENCE



GIS SCIENCE JOURNAL



An UGC-CARE Approved Group II Journal

ISSN NO : 18669-9391 / Website : www.gisscience.net/
Email : editorgsjournal@gmail.com

Certificate of Publication

Paper ID : GSJ/10050

This is to certify that the paper titled

SMART RESTAURANT MANAGEMENT SYSTEM

Authored by

B.Sai Kiran

From

Sasi Institute Of Technology And Engineering.

Has been published in

GIS SCIENCE JOURNAL Volume 10, Issue 3, March 2023.




M Palaniswami
Editor-in-chief
GISSCIENCE





CERTIFICATE OF COMPLETION

Presented to

Pinnamani Hemanth

For successfully completing a free online course
Introduction to Natural Language Processing

Provided by

Great Learning Academy

(On February 2023)