

A Comprehensive Study on Mess Meal Planner Application

by

Md.Tamim

ID: CSE1902017004

Md.Mahmudul Hasan

ID: CSE1902017011

Md. Shakib Khan

ID: CSE1902017079

Md.Abu Sadat Rana

ID: CSE1902017078

Supervised by

Sadia Tasnim Barsha

Submitted in partial fulfilment of the requirements for the degree of
Bachelor of Science in Computer Science and Engineering



**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
SONARGAON UNIVERSITY (SU)**

January 2023

A Comprehensive Study on Mess Meal Planner Application

by

Md.Tamim

ID: CSE1902017004

Md.Mahmudul Hasan

ID: CSE1902017011

Md. Shakib Khan

ID: CSE1902017079

Md.Abu Sadat Rana

ID: CSE1902017078

Supervised by

Sadia Tasnim Barsha

Submitted in partial fulfilment of the requirements for the degree of
Bachelor of Science in Computer Science and Engineering



**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
SONARGAON UNIVERSITY (SU)**

January 2023

APPROVAL

The Project titled “**A Comprehensive Study on a Mess Meal Planner Application**” submitted by Md. Tamim(CSE1902017004), Md. Mahmudul Hasan(CSE1902017011), Md. Shakib Khan(CSE1902017079) and Md. Abu Sadat Rana(CSE1902017078) to the Department of Computer Science and Engineering, Sonargaon University (SU), has been accepted as satisfactory for the partial fulfilment of the requirements for the degree of Bachelor of Science in Computer Science and Engineering and approved as to its style and contents.

Board of Examiners

Sadia Tasnim Barsha
Lecturer & Exam Coordinator,
Department of Computer Science and Engineering
Sonargaon University (SU)

Supervisor

(Examiner Name & Signature)
Department of Computer Science and Engineering
Sonargaon University (SU)

Examiner 1

(Examiner Name & Signature)
Department of Computer Science and Engineering
Sonargaon University (SU)

Examiner 2

(Examiner Name & Signature)
Department of Computer Science and Engineering
Sonargaon University (SU)

Examiner 3

DECLARATION

We, hereby, declare that the work presented in this report is the outcome of the investigation performed by us under the supervision of **Sadia Tasnim Barsha**, Lecturer & Exam Coordinator, Department of Computer Science and Engineering, Sonargaon University, Dhaka, Bangladesh. We reaffirm that no part of this project has been or is being submitted elsewhere for the award of any degree or diploma.

Countersigned

Signature

(Sadia Tasnim Barsha)
Supervisor

Md. Tamim
ID: CSE1902017004

Md.Mahmudul Hasan
ID: CSE1902017011

Md. Shakib Khan
ID: CSE1902017079

Md.Abu Sadat Rana
ID: CSE1902017078

ABSTRACT

The Project mess Meal Planner is a mobile application that allows the user to handle all dining activities online. Interactive GUI and real-time database. The ability to manage cash credit and debit makes this system very flexible and convenient. The group of people is very busy and does not have the time to manage and account for the entire activities of dining manually on the paper pen. This application gives them the power and comfort to manage the entire dining from a single system. Mess Meal Planner provides meal management with mess members and every member transaction of the cost of meals. Any person can create a mess. The creator of the mess will be the mess admin. Any member can join a mess by entering the mess's unique name which is approved by the admin. Admin has the power of removing any member of the mess. Every member of the mess can see the number of their meal and the amount they have to pay. The application is hence useful for both members and managers to portably manage mess activities.

ACKNOWLEDGMENT

At the very beginning, we would like to express our deepest gratitude to Almighty Allah for giving us the ability and strength to finish the task successfully within the scheduled time.

We are auspicious that we had the kind association as well as supervision of **Sadia Tasnim Barsha**, Lecturer & Exam Coordinator, Department of Computer Science and Engineering, Sonargaon University whose hearted and valuable support with the best concern and direction acted as necessary recourse to carry out our project.

We would like to convey our special gratitude to **Prof. Dr Md Alamgir Hossain**, Dean, Faculty of Science and Engineering for his kind concern and precious suggestions.

We are also thankful to all our teachers during our whole education, for exposing us to the beauty of learning.

Finally, our deepest gratitude and love to my parents for their support, encouragement, and endless love.

LIST OF ABBREVIATIONS

API	Application Programming Interface
CDN	Content Delivery Network
CPU	Central Processing Unit
CSS	Cascading Style Sheets
GIT	Global Information Tracker
GUI	Graphical user interface
HTML	Hyper Text Markup Language
IDE	Integrated Development Environment
I/O	Input/Output
iOS	iPhone Operating System
JIT	Just In Time
JSX	JavaScript XML
ORM	Object Relational Mapper
RDBMS	Relational Database Management System
SQL	Structured Query Language
TLS	Transport Layer Security
UI	User Interface
UX	User Experience
XML	Extensible Markup Language

TABLE OF CONTENTS

<u>Title</u>	<u>Page No</u>
DECLARATION	ii
ABSTRACT	iii
ACKNOWLEDGEMENT	iv
LIST OF ABBREVIATIONS	v
CHAPTER 1	1-3
INTRODUCTION TO MESS MEAL PLANNER APPLICATION	
1.1 Introduction	1
1.2 Objective of Mess Meal Planner Application	1
1.3 Needs of Mess Meal Planner Application	1
1.4 Methodology Development Model	2
CHAPTER 2	4-16
TOOLS AND TECHNIQUE	
2.1 Tools and Technique	4
2.2 Specification Requirement	
2.2.1 External Interfaces	9
2.2.2 Software Product Features	14
CHAPTER 3	17-20
SYSTEM DESIGN	
3.1 Table Design	17
3.2 Use Case Diagram	
3.2.1 Members Use Case Diagram	18
3.2.2 Admin Use Case Diagram	19
3.3 Flow Chart	20

CHAPTER 4	21-24
TASK AND ACTIVITIES PERFORMED	
4.1 Advantage	21
4.2 Structure of The Project	21
4.3 Feasibility Study	
4.3.1 Economic Feasibility	22
4.3.2 Technical Feasibility	23
4.4 System Analysis	23
4.5 System Design	23
4.6 Implementation	24
4.7 Test Generation	24
CHAPTER 5	25-26
LIMITATIONS, POST IMPLEMENTATION & CONCLUSION	
5.1 Limitation	25
5.2 Future Work	25
5.3 Conclusion	26
REFERENCES	27-28

LIST OF FIGURES

<u>Figures No.</u>	<u>Title</u>	<u>Page No.</u>
Fig.1.1	Waterfall Model of Methodology Development.....	2
Fig.3.1	Table Design of Mess Meal Application	17
Fig.3.2	Members Use Case Diagrams	18
Fig.3.3	Admin Use Case Diagram.....	19
Fig.3.4	Flowchart of Mess Meal Application	20
Fig.4.2	Cost and Details.....	22

CHAPTER 1

INTRODUCTION TO MESS MEAL PLANNER

1.1 Introduction

For a better understanding of mess meal management, this application is designed simply. It would avoid confusion and help operate the management easily. Also, such an application that is easy to use will reduce the work of mess managers who still maintain all the logs in registers and files. It would be of great benefit as all calculations would be done easily with the click of a button. The overall goal of this project was to first study and understand the existing mess/canteen management application, then identify the limitations and contribute to the same topic with greater benefits. The main advantage of our proposed system over other existing systems is the GUI in the English language. This has enabled complete understanding and convenience for the user. Also, complex calculations are done within seconds and bills are generated with the single click of a button. As the name suggests, it is an application for maintaining any mess/canteen but it also enables handling of the information related to the students/employees who are a part of the organisation. Another attractive feature in our application which is lacking in the existing application is the Backup and recovery option. All data can be stored as a copy, that is, taking a backup is also possible with a single click, plus recovery of lost data in case any failure occurs is also possible with a single click. Further in this thesis, we will discuss meal planning with advanced technology management through an app and its premium features.

1.2 Objective of mess Meal Planner Application

When some or more people live together in a place like a hostel, army base, mess etc. They face problems to maintain their dining management. It's hard to remember what amount of meal they had and what amount of cost they expensed. They try to solve this problem by calculating manually and faced so many difficulties.

Mess Meal Application solved this problem in a moment. Now There is no need to paper pen work or manual calculation to solve this problem. It can calculate it digitally for them, freeing them from difficulty. Save valuable time and make life much easier.

1.3 Needs of Mess Meal Planner Application

This application will help the administration to work easily. Because of its easy access and less time-consuming administration, it can get information about the user's meals, shopping, transactions, etc. They do not have to search in the paper file for a long time. Members can easily handle the application.

1.4 Methodology Development Model

Waterfall approach was the first SDLC Model to be used widely in Software Engineering to ensure the success of the project. In "The Waterfall" approach, the whole process of software development is divided into separate phases. In this Waterfall model, typically, the outcome of one phase acts as the input for the next phase sequentially.

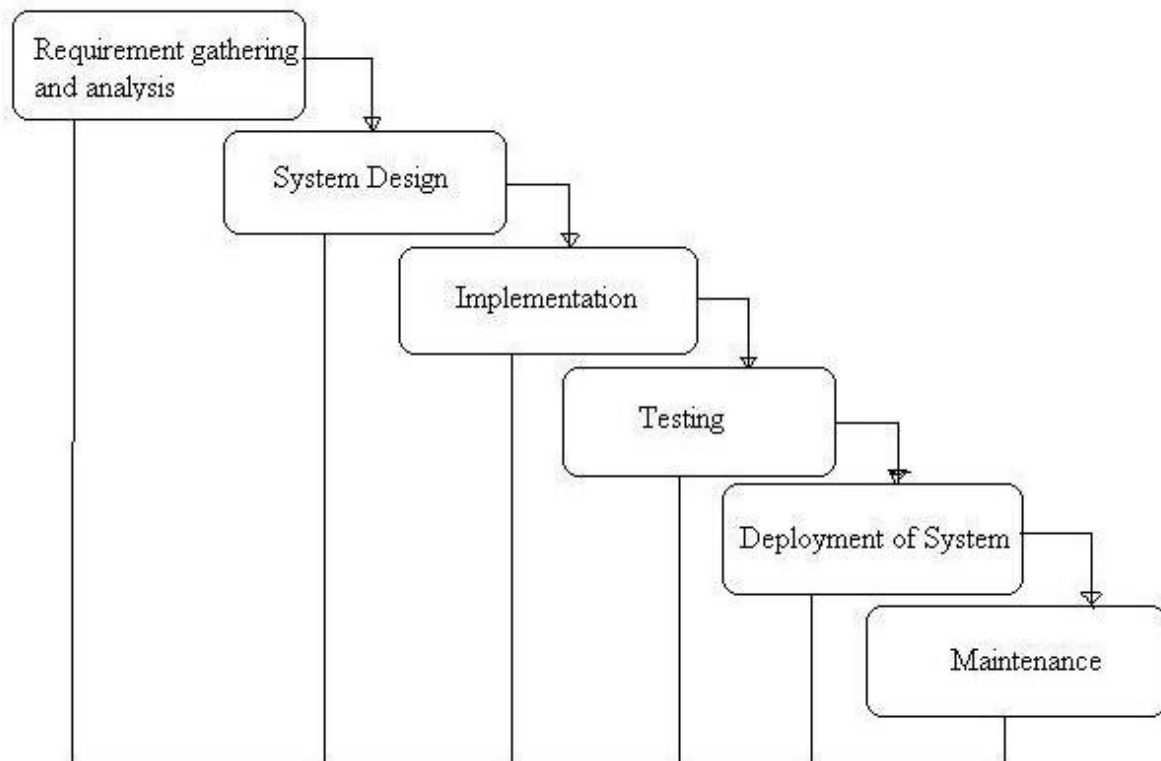


Fig- 1.1 Waterfall Model of Methodology Development

The sequential phases in the Waterfall model are –

- **Requirement Gathering and analysis** – All possible requirements of the system to be developed are captured in this phase and documented in a requirement specification document.

- **System Design** – The requirement specifications from the first phase are studied in this phase and The system design is prepared. This system design helps specify hardware and system requirements and helps define the overall system architecture.
- **Implementation** – With inputs from the system design, the system is first developed in small programs called units, which are integrated with the next phase. Each unit is developed and tested for its functionality, referred to as Unit Testing.
- **Integration and Testing** – All the units developed in the implementation phase are integrated into a system after testing each unit. Post integration the entire system is tested for any faults and failures.
- **Deployment of system** – Once the functional and non-functional testing is done; the product is deployed in the customer environment or released into the market.
- **Maintenance** – There are some issues which come up in the client environment. To fix those issues, patches are released. Also to enhance the product some better versions are released. Maintenance is done to deliver these changes in the customer environment.

CHAPTER 2

TOOLS AND TECHNIQUE

2.1 Tools and technique

- **Frontend**
 - React Native
 - Typescript
 - Redux
 - Tailwind CSS
- **Backend**
 - Express & Node Js
 - Prisma
- **Database**
 - Postgres
- **Version Control**
 - Github
- **Database Hosting**
 - Render

React Native:

React Native is a JavaScript framework for writing real, natively rendering mobile applications for iOS and Android. It's based on React, Facebook's JavaScript library for building user interfaces, but instead of targeting the browser, it targets mobile platforms. In other words: web developers can now write mobile applications that look and feel truly "native," all from the comfort of a JavaScript library that we already know and love. Plus, because most of the code you write can be shared between platforms, React Native makes it easy to simultaneously develop for both Android and iOS.

Similar to React for the Web, React Native applications are written using a mixture of JavaScript and XML-Esque markup, known as JSX. Then, under the hood, the React Native "bridge" invokes the native rendering APIs in Objective-C (for iOS) or Java (for Android).

Thus, your application will render using real mobile UI components, not web views, and will look and feel like any other mobile application. React Native also exposes JavaScript interfaces for platform APIs, so your React Native apps can access platform features like the phone camera, or the user's location.

In contrast, React Native translates your markup to real, native UI elements, leveraging existing means of rendering views on whatever platform you are working with. Additionally, React works separately from the main UI thread, so your application can maintain high performance without sacrificing capability. The update cycle in React Native is the same as in React, when props or state change, React Native re-renders the views. The major difference between React Native and React in the browser is that React Native does this by leveraging the UI libraries of its host platform, rather than using HTML and CSS markup.

Typescript:

TypeScript extends JavaScript and improves the developer experience. It enables developers to add a type of safety to their projects. Moreover, TypeScript provides various other features like interfaces, type aliases, abstract classes, function overloading, tuple, generics, etc.

Some benefits of using TypeScript over JavaScript are:

- Static typing – TypeScript comes with optional static typing and a type inference system, which means that a variable, declared with no type may be inferred by TypeScript based on its value.
- Object-oriented programming – TypeScript supports object-oriented programming concepts like classes, inheritance, etc. Compile time checks – JavaScript is an interpreted programming language. There is no compilation involved. Hence, the errors get caught during the runtime. Since TypeScript compiles into JavaScript, errors get reported during the compile time rather than the runtime.
- Code editor support – IDEs or code editors like VS Code support autocomplete for a TypeScript codebase. They also provide inline documentation and highlight the errors.

Redux:

React Redux is the official Redux UI binding library for React. It is a predictable state container for JavaScript apps.

While Redux can be used with any UI layer, it was originally designed and intended for use with React. There are UI binding layers for many other frameworks, but React Redux is maintained directly by the Redux team.

As the official Redux binding for React, React Redux is kept up-to-date with any API changes from either library, to ensure that our React components behave as expected. Its intended usage adopts the design principles of React - writing declarative components. It implements performance optimizations for us.

React is generally fast, but by default, any updates to a component will cause React to re-render all of the components inside that part of the component tree. This does require work, and if the data for a given component hasn't changed, then re-rendering is likely some wasted effort because the requested UI output would be the same.

It helps us write applications that behave consistently, run in different environments (client, server, and native), and are easy to test. On top of that, it provides a great developer experience, such as live code editing combined with a time-travelling debugger.

Tailwind CSS:

Tailwind CSS is an open-source CSS framework. The main feature of this library is that, unlike other CSS frameworks like Bootstrap, it does not provide a series of predefined classes for elements such as buttons or tables.

Tailwind CSS makes it quicker to write and maintain the code of our application. By using this utility-first framework, you don't have to write custom CSS to style our application. Instead, we can use utility classes to control the padding, margin, colour, font, shadow, and more of our application.

Before Tailwind v3, it used to purge through all the styles to remove any unused styles, so that the production build remained as small as possible. According to Tailwind, the production build used to be between 5-10 kB. But that's the story in production. In a development environment, CSS might get big especially if we use a lot of personalized configuration.

With v3 and above, Tailwind released a new feature called the Just-in-Time compiler. The JIT compiler avoids compiling all the CSS upfront and compiles only the CSS as and when we need it. This results in lightning-fast build times in all environments. And as the styles are generated as and when we need them, there is no need to purge unused styles. This means that the CSS in all the environments will be the same. This helps us get rid of the fear of any important CSS getting purged in production.

Express & Node Js:

Node.js is a very powerful JavaScript-based platform built on Google Chrome's JavaScript V8 Engine. It is easily employed as a server-side proxy, where it can handle a large number of simultaneous connections in a nonblocking manner. A Node.js app runs in a single process, without creating a new thread for every request. Node.js provides a set of asynchronous I/O primitives in its standard library that prevent JavaScript code from blocking and generally, libraries in Node.js are written using non-blocking paradigms, making blocking behaviour the exception rather than the norm. When Node.js performs an I/O operation, like reading from the network, or accessing a database or the filesystem, instead of blocking the thread and wasting CPU cycles waiting, Node.js will resume the operations when the response comes back.

This allows Node.js to handle thousands of concurrent connections with a single server without introducing the burden of managing thread concurrency, which could be a significant source of bugs.

Express is a node js web application framework that provides broad features for building web and mobile applications. It is used to build a single page, multipage, and hybrid web application. It's a layer built on top of the Node js that helps manage servers and routes. Express derives various features from Node. js, and one of them is the non-blocking servers that can handle user requests better. As a result, it is easier for developers to create easily scalable web apps.

Prisma:

Prisma is an open-source database toolkit that makes it easy for developers to reason about their data and how they access it, by providing a clean and type-safe API for submitting database queries. It mainly consists of three tools:

- Prisma Client: An auto-generated and type-safe query builder for Node.js & TypeScript.
- Prisma Migrate (experimental): A declarative data modelling & migration system.
- Prisma Studio (experimental): A GUI to view and edit data in a database.

Prisma is a server-side library that helps developers read and write data to the database in an intuitive, efficient and safe way. It is a next-generation object-relational mapper (ORM) that claims to help developers build faster and make fewer errors. Also, it is an evidence-based minimum set of items for reporting in systematic reviews and meta-analyses.

Benefits of using PRISMA:

- Demonstrate the quality of the review
- Allow readers to assess strengths and weaknesses
- Permits replication of review methods
- Structure and format the review using PRISMA headings

Postgres:

PostgreSQL also known as Postgres. PostgreSQL is an advanced, enterprise-class, and open-source relational database system. PostgreSQL supports both SQL (relational) and JSON (non-relational) querying.

PostgreSQL comes with many features aimed to help developers build applications, administrators protect data integrity and build fault-tolerant environments and help you manage your data no matter how big or small the dataset.

It's less prone to data corruption. It's still the most advanced open-source RDBMS in the world. And it's an excellent choice for those who are new to SQL.

GitHub:

GitHub is a for-profit company that offers a cloud-based Git repository hosting service. Essentially, it makes it a lot easier for individuals and teams to use Git for version control and collaboration.

GitHub's interface is user-friendly enough so even novice coders can take advantage of Git. Without GitHub, using Git generally requires a bit more technical savvy and the use of the command line.

Git is so user-friendly, though, that some people even use GitHub to manage other types of projects. Version control helps developers track and manage changes to a software project's code. As a software project grows, version control becomes essential.

Render:

Render is a unified cloud to build and run all your apps and websites with free TLS certificates, global CDN, private networks and auto-deploy from Git.

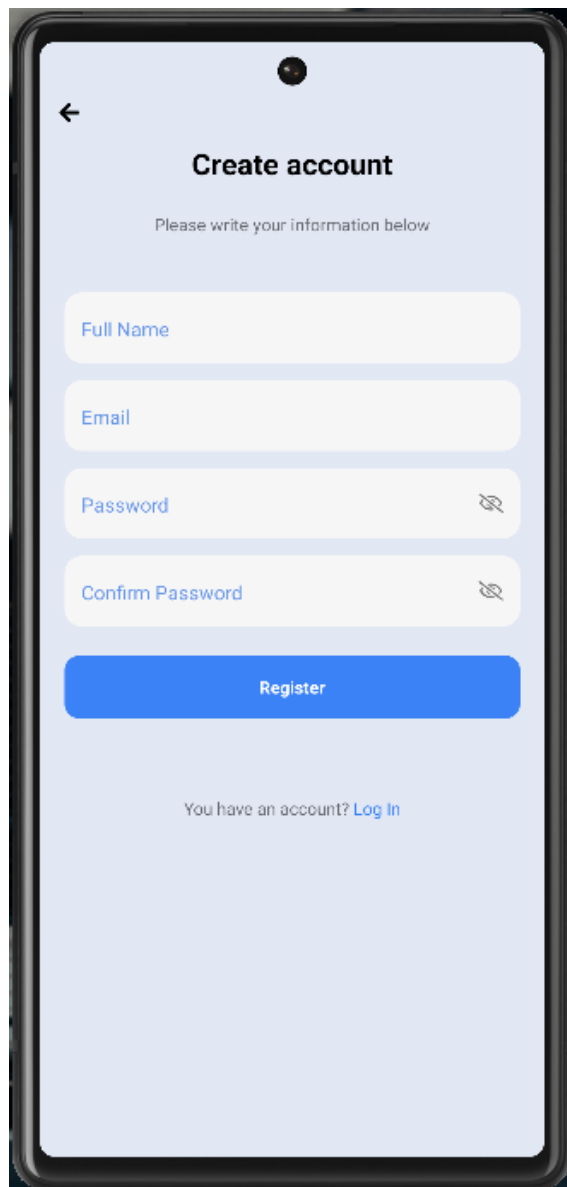
Render is a fully-managed cloud platform where we can host static sites, backend APIs, databases, cron jobs, and all our other apps in one place.

2.2 Specification Requirement

2.2.1 External Interfaces:

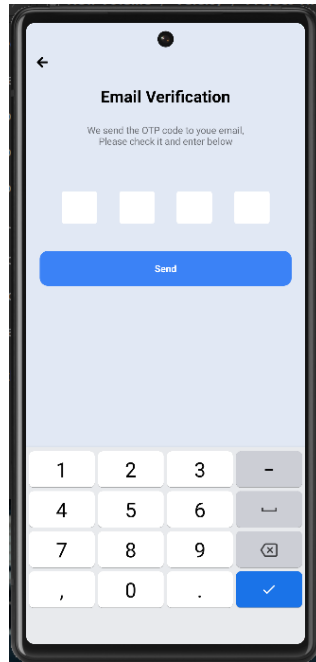
This interface will be the actual interface through which the user will communicate with the application and perform the desired tasks.

Create account

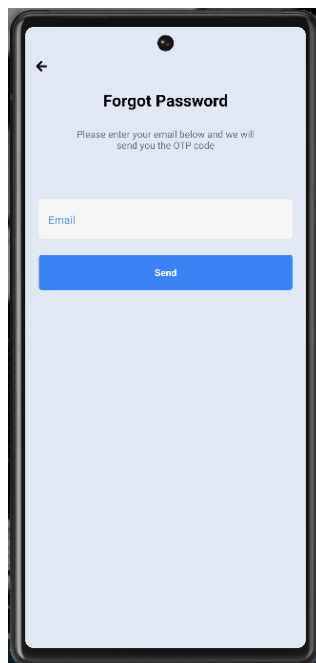


The image shows a mobile application interface for creating an account. The screen has a light blue background. At the top left, there is a back arrow icon. The title 'Create account' is centered at the top. Below the title, a subtitle reads 'Please write your information below'. There are four input fields: 'Full Name', 'Email', 'Password', and 'Confirm Password'. The 'Password' and 'Confirm Password' fields have a small eye icon to the right, indicating a toggle for visibility. Below the input fields is a prominent blue button labeled 'Register'. At the bottom of the screen, there is a link that says 'You have an account? Log In'.

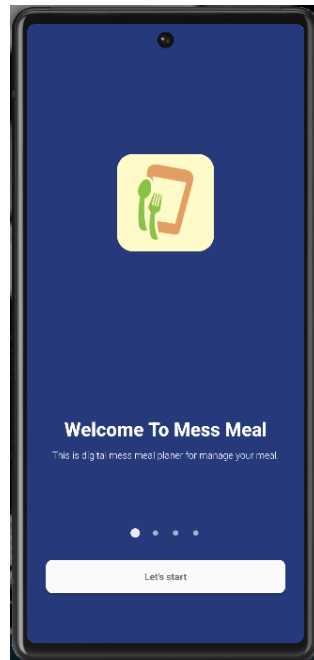
Email verification



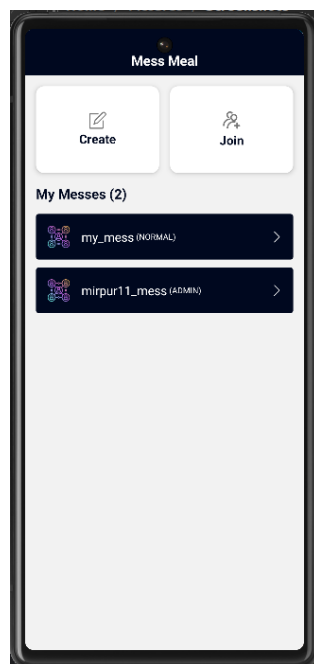
Forgot password



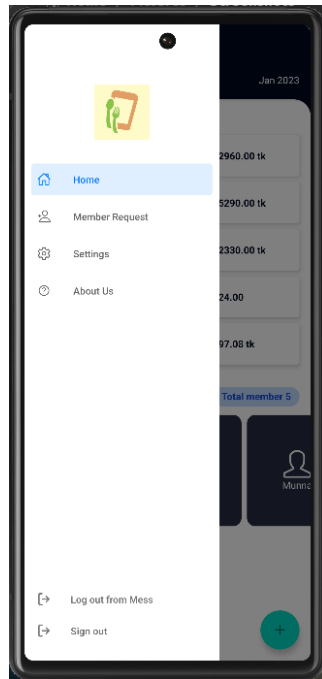
Log in



Home page



☐ Drawer



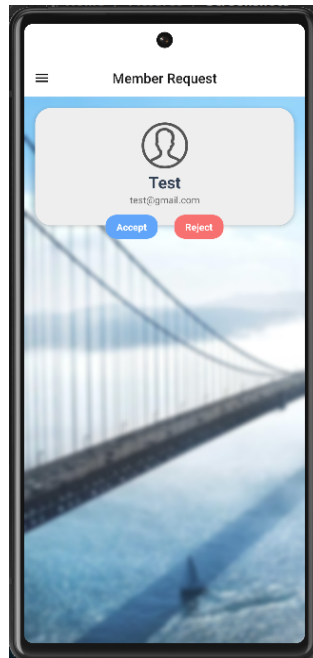
☐ Create mess



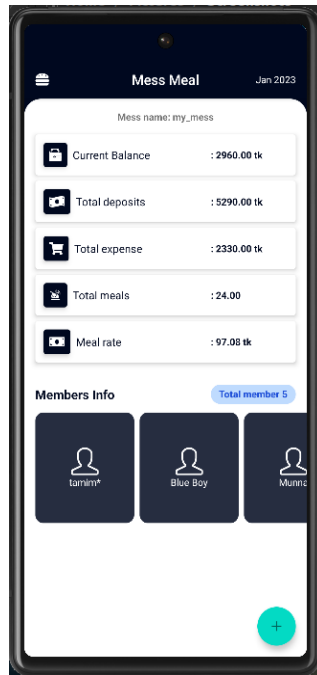
Join Mess



Member Request

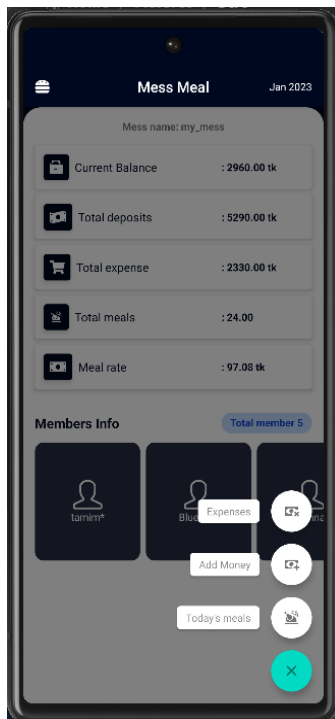


Mess information

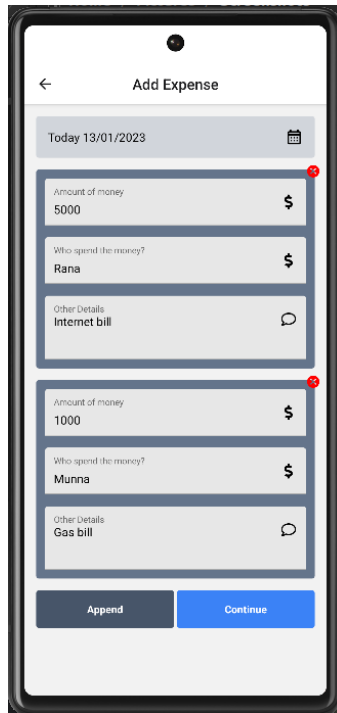


2.2.2 Software Product Features:

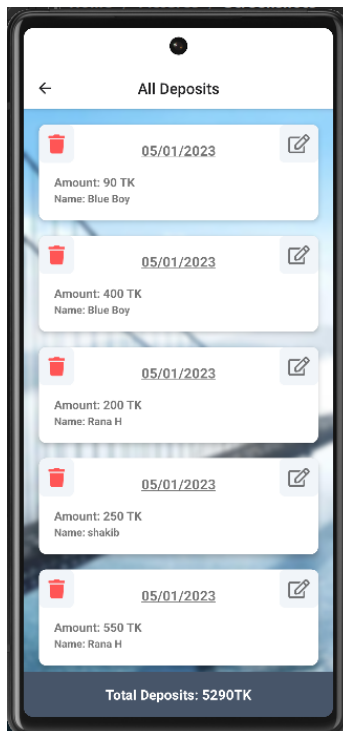
Input data



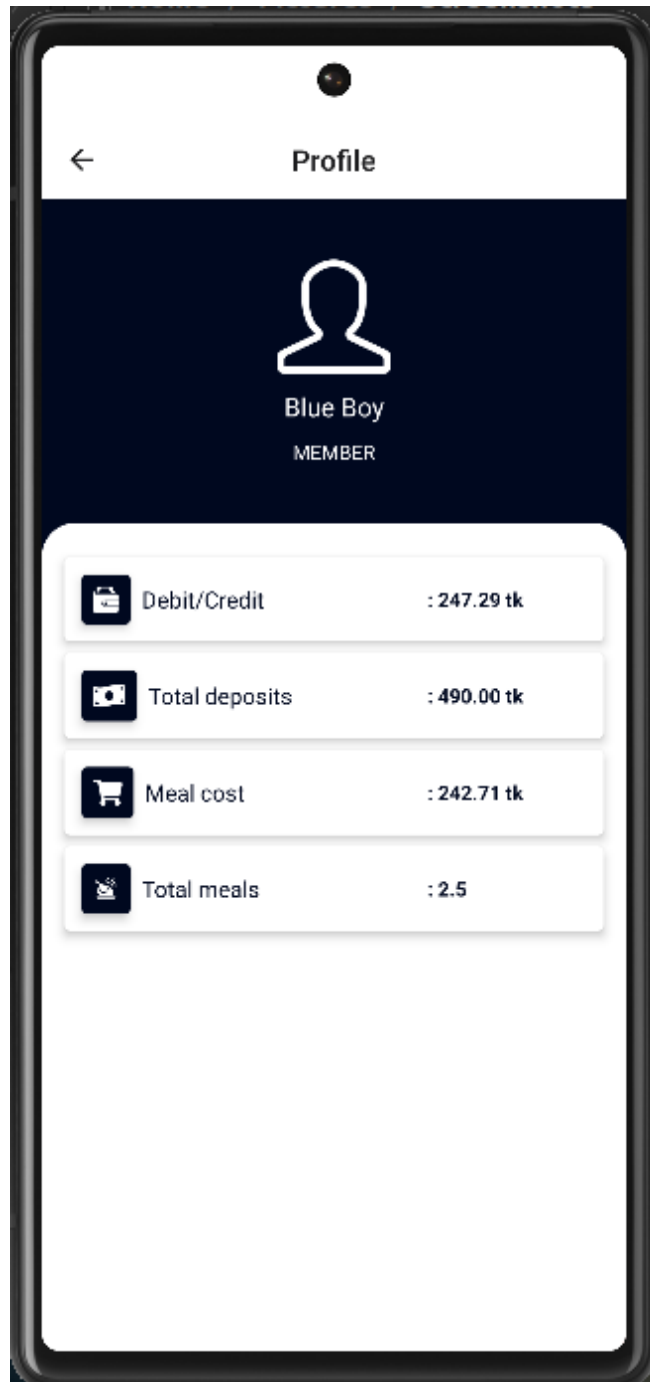
Add expense



All deposits



☐ Member profile



CHAPTER 3

SYSTEM DESIGN

3.1 Table design

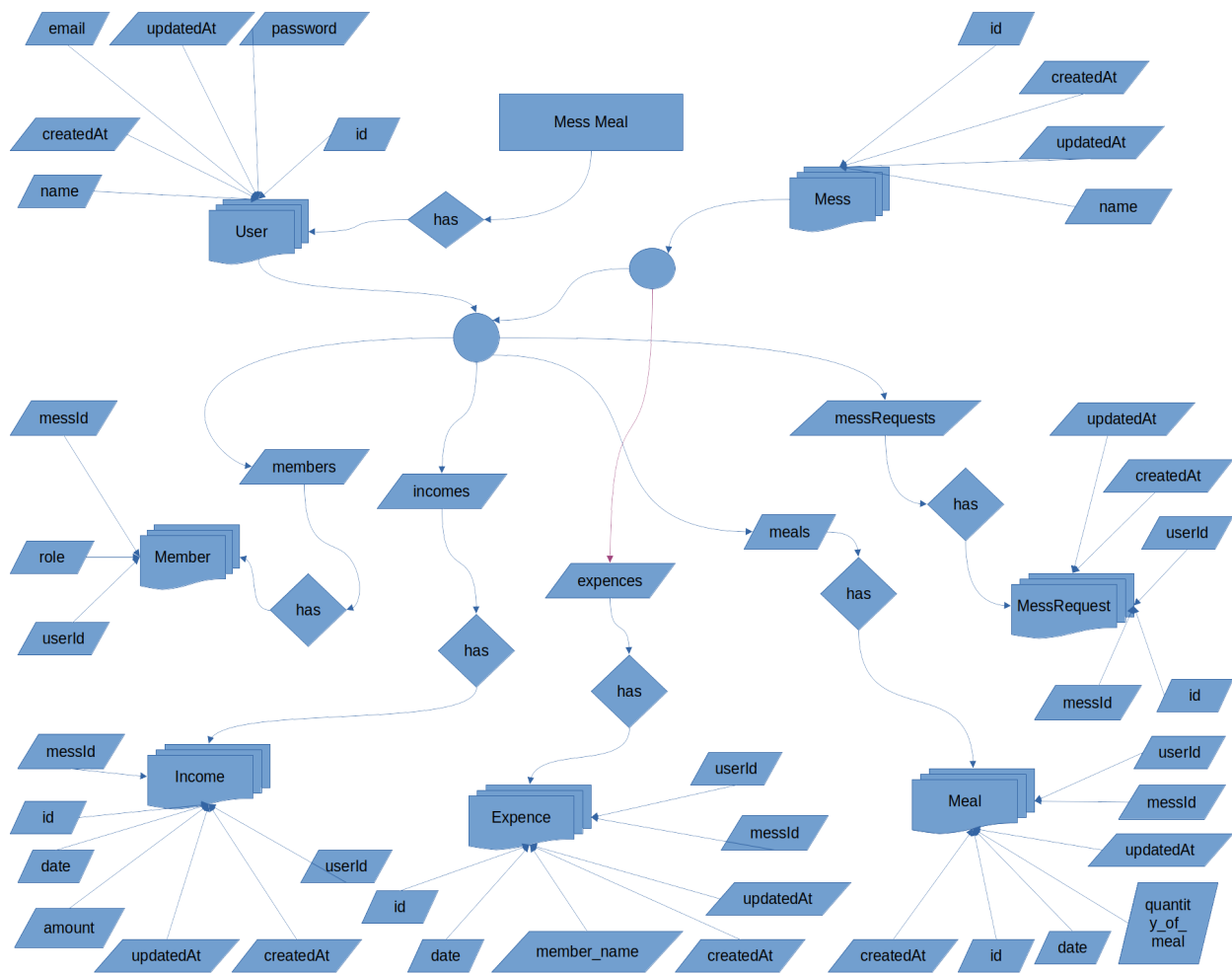


Figure 3.1: Table Design of Mess Meal Application

3.2 Use case diagram

3.2.1 Members Use Case Diagram:

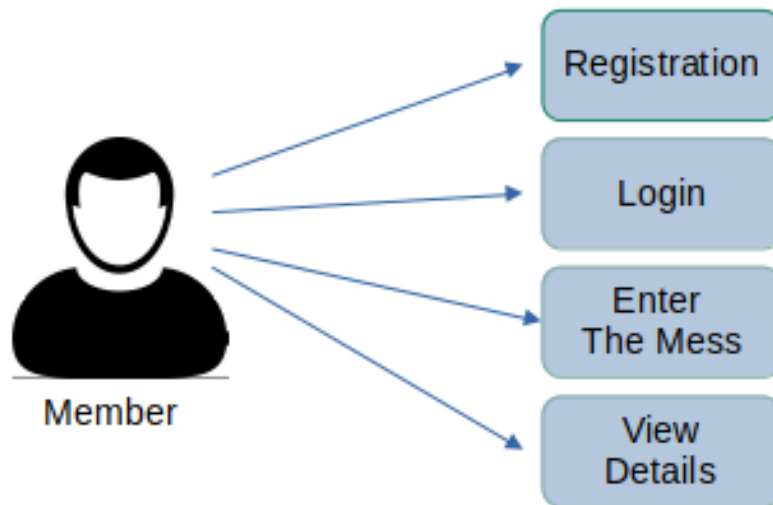


Figure 3.2: Members Use Case Diagram

3.2.2 Admin Use Case Diagram:

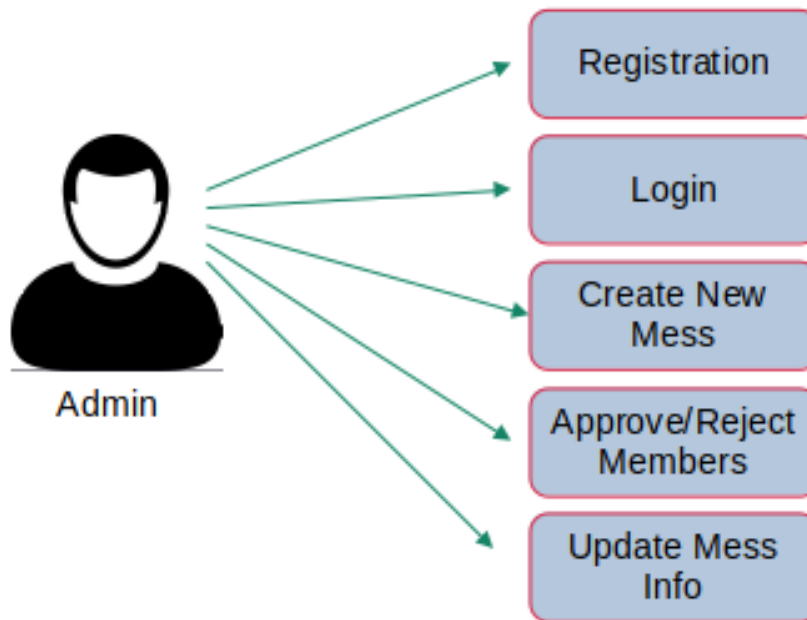


Figure 3.3: Admin Use Case Diagram

3.3 Flow chart

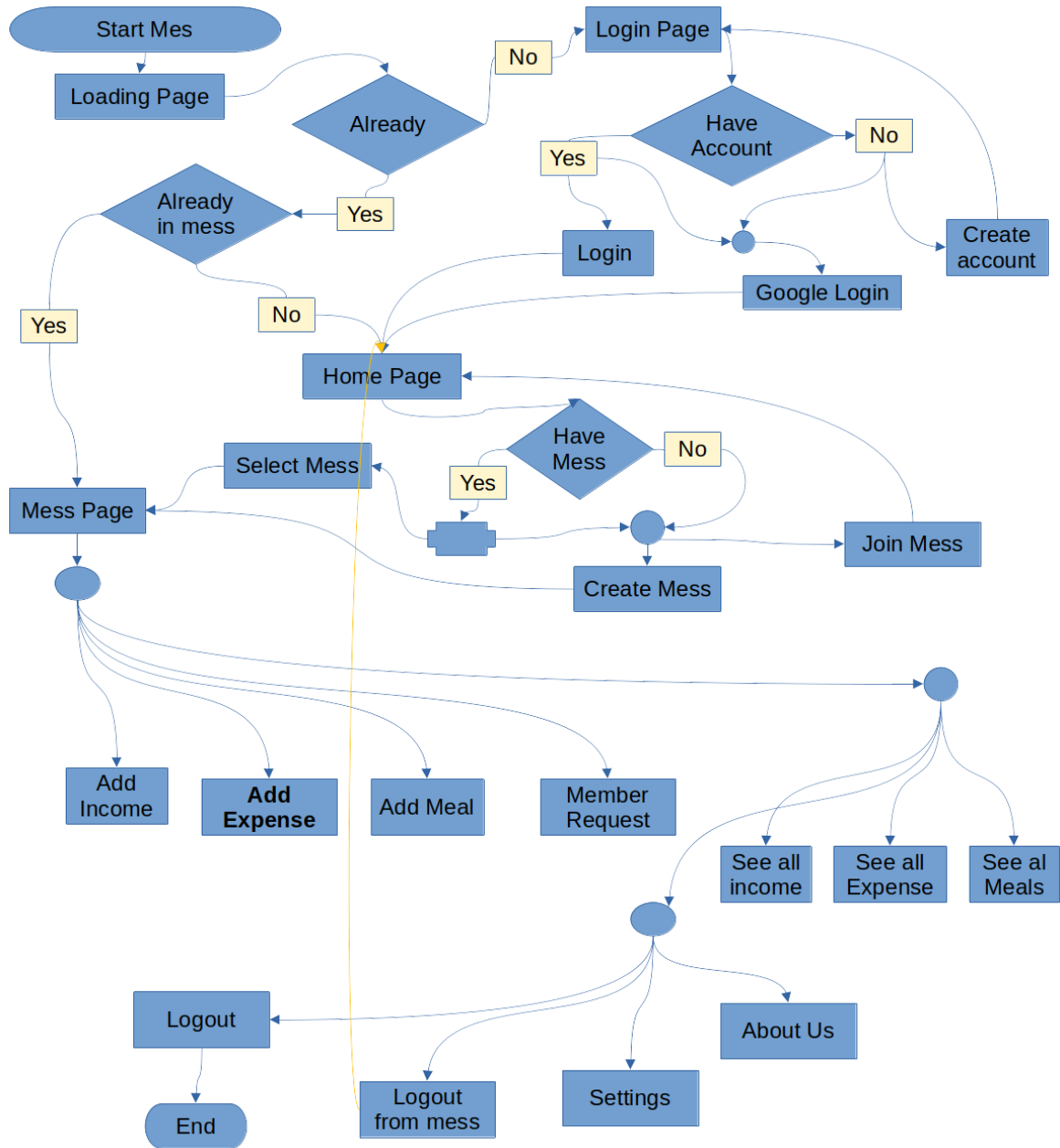


Figure 3.4: Flowchart of Mess Meal Application

CHAPTER 4

TASK AND ACTIVITIES PERFORMED

4.1 Advantage

The mess meal system is a dynamic as well as practical approach as it makes managing the boarding and other related facilities a lot easier. One of the key benefits of a mess meal system is that it reduces the burden on the mess manager and simplifies roles and responsibilities as most of the manual tasks and mundane paperwork can be done through the online system. An online mess meal system ensures record management and flawless execution of administrative tasks such as managing mess meal applications, registration, fees management, mess payment management, disciplinary issues and easy communication with mess members. The mess meal administration team can access all the mess meal data and manage the workflow from their device at all times – ensuring that updates are made in real time and notifications to mess members can also be sent in real-time. It also helps securely store and manage the data of the mess member.

4.2 Structure of the project

- After Administrator Login:
 - Approve or Reject member's entry request
 - Input expense
 - Add money
 - Input meal status
 - Update all information
 - Give administrative power to a member

- After Member login:
 - Request admin to join a mess
 - Can view of all information of the mess
 - Leave the mess
 - Can create a new mess as an admin

4.3 Feasibility Study

A feasibility study is an analysis that takes all parts of a project’s relevant factors into account – including economic, technical, legal and scheduling considerations – to ascertain the likelihood of completing the project successfully. (Will Kenton, 2020). Feasibility studies are necessary because they help the project, concept or plan inside and out and potential problems are easily highlighted.

4.3.1 Economic Feasibility

The computerised system will help in automating the selection leading to the cost and details of the mess members. For these services, this application can be used to save precious time.

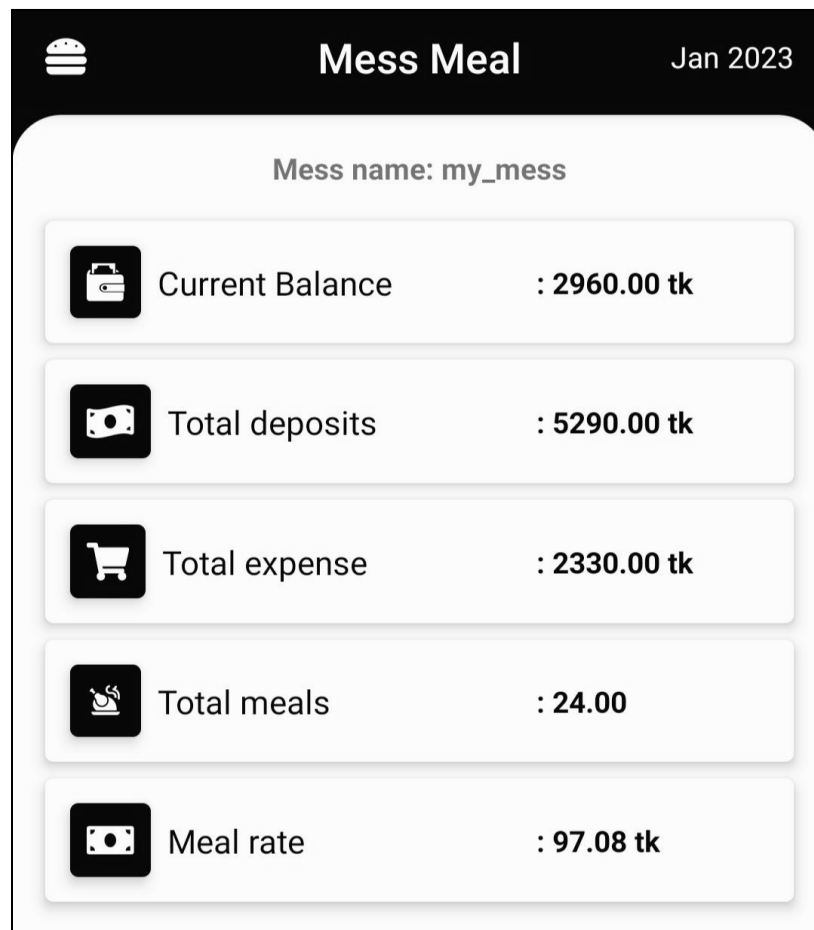


Figure 4.2: Cost and details

4.3.2 Technical Feasibility

The project is developed in React Native which is a cross-platform mobile application development platform that supports both Android and iOS devices. They provide a high level of reliability, availability and compatibility. All these qualities make the chosen framework appropriate for this project.

4.4 System Analysis

A system is a collection of different parts to form a functional unit that achieves a particular goal or outcome. Therefore, system analysis is a problem-solving approach that breaks a larger system into these previously mentioned parts to figure out how it works so that a specific goal can be achieved. This problem-solving method can help a person learn how to use systems that were created by someone else and plan for new systems. System analysis also reduces errors when using systems to solve problems. For example, a person may resolve a problem within a few minutes but then weeks later find out that what they fixed caused other errors to occur. The better you understand a system, the lower the chances of such occurrences.

4.5 System Design

Systems design is an interdisciplinary engineering activity that enables the realization of successful systems. A system may be defined as an integrated set of components that accomplish a defined objective. The process of systems design includes defining software and hardware architecture, components, modules, interfaces, and data to enable a system to satisfy a set of well-specified operational requirements.

In general, systems design, systems engineering, and systems design engineering all refer to the same intellectual process of being able to define and model complex interactions among many components that comprise a system and being able to implement the system with proper and effective use of available resources. Systems design focuses on defining customer needs and required functionality early in the development cycle, documenting requirements, and then proceeding with design synthesis and system validation while considering the overall problem consisting of Systems design integrates all of the engineering disciplines and speciality groups into a team effort forming a structured development process that proceeds from concept to production to operation.

- Operations
- Performance
- Test and integration
- Manufacturing
- Cost and schedule
- Deployment
- Training and support
- Maintenance
- Disposal

4.6 Implementation

The main aim of any mess is to provide clean and fresh food to every member of the team. In many teams, entire mess management and billing calculations are done manually to date. It is very time-consuming and increases the chances of performing calculation mistakes. It would be possible to do the same work within a short period and without using much effort and manpower if there existed software for the same. Thus, there arises a need to create software for the same. Such software would make the entire Mess-related management an automated system. The software is not only restricted to food items and their billing manipulations but maintaining diets routines in mess is also possible through the apps. Thus, such a combination in a single software is of great benefit.

4.7 Test Generation

This activity generates a set of test data, which can be used to test the new system before accepting it. In the test generation phase, all the parts come which are to be tested to ensure that the system does not produce any errors. If there are some errors then we remove them and let the system provide service.

CHAPTER 5

LIMITATIONS, POST IMPLEMENTATION & CONCLUSION

5.1 Limitation

Time constraints were the major factors that hindered the progress of this research. The study involved a lot of financial obligations such as the cost of stationery, printing, and photocopying. Moreover, lectures to produce a comprehensive research report within the time limit were tiresome. Nevertheless, the quality of this study was not compromised.

There is the most common limitation-

1. A member must have a smartphone to run the application.
2. A member has to have internet access to run the application.

5.2 Future Work

Since there are no applications in the android app store or apple store to maintain a dining account, this project idea in our brains for a very long time. People used to do their maintained calculations in hand notes or excel files. We did market research and seems that this application demand is high in this field. We would like to add more features to use it comfortably. Will Add more responsibility to admin and members so that they can utilize it more efficiently.

The Application is currently hosted on a live server. We are working on the enhancement of this application concept and want to make it cosier. And also we will solidify and enhance the project features, UI and UX more commercially.

5.3 Conclusion

This app will be useful to any school/college hostel or in general to any institute maintaining a mess or canteen. Security is maintained as the complete control of the system is only under the hands of an authorised person. It will enable mess managers to refer to diet charts and plan diets accordingly for the cadets. Since easy calculations and bill generations are done in moments, therefore, it will lessen the work of mess managers.

Using this app, alerts regarding the stock of commodities will be easily generated. It will never let scarcity prevail in the mess. Backup of data can be easily taken using this app with the click of a single button. Restoring the data to the system is also possible in case of any failure. Also, this app will reduce manpower, thus reducing the capital being invested. This project can be merged with any major projects in future where meals and their monthly cost management can be done.

REFERENCES

- [1] React Native and native application development PAPER WITHIN Computer Science
Author: Oskar Svensson & Marcus Presa Kåld, Tutor: Peter Larson-Green, JÖNKÖPING April 2021
- [2] Bonnie Eisenman. Learning React Native; Chapter 1; 2018
- [3] D. Flanagan. Javascript: The Definitive Guide - Master the World's Most-Used Programming Language. Sebastopol: O'Reilly Media, 2020. ISBN: 978-1-491-95202- 3 (cit. on p. 19).
- [3] Kurniawan, A. (2014). Node.js Succinctly. Synfusion Inc.
- [4] Tom Hughes-Croucher, M. W. (2012). Node: Up and Running. In M. W. Tom Hughes-Croucher, Node: Up and Running. O'Reilly Media Inc.
- [5] Software Engineering Roger S.Pressman UK, c2000, 5th ed.
- [6] Rauch, G. (2012). Smashing Node.JS JavaScript Everywhere. John Wiley & Sons Inc.
- [7] Teixeira, P. (2013). Professional NodeJs: Building JavaScript-Based Scalable Software. John Wiley & Sons Inc.
- [8] Wandschneider, M. (2013). Learning Node.js A Hands-On Guide to Building Web Applications in JavaScript. Pearson Education.
- [9] Wilson, J. R. (2013). Node.js The Right Way Practical, Server-Side JavaScript That Sales. The Pragmatic Programmers LLC.
- [10] Flanagan, D. (2006). JavaScript The Definitive Guide. O'Reilly Media Inc.
- [11] Jianye Liu, Jianka Yu, 'Research on Development of Android Applications' in Fourth International Conference on Intelligent System -2011
- [12] Resham Shinde, Priyanka Thakare, Neha Dhomne, Sushmita Sarkar Design and Implementation of Digital dining in Restaurants using Android.

- [13] Sams, C. (2018, November 28). The Importance Of Restaurant Management Software Why You Need It. Retrieved November 27, 2020.
- [14] De Groef, W., Massacci, F., & Piessens, F. (2014). A security architecture for server-side JavaScript.
- [15] A. Dutt and J.R. Haritsa. Plan bouquets: Query processing without selectivity estimation. SIGMOD, 2014.
- [16] WAGH, Dr K; THOOL, Ravindra. A Comparative Study of SOAP vs REST web services provisioning techniques for the mobile host. Journal of Information Engineering and Applications. 2012, vol. 2, pp. 12–16.
- [17] Aebersold, K. (2018, June 17). Software Testing Methodologies. Retrieved March 20, 2021
- [18] Software project management: from concept to deployment / Kieron Conway. Scottsdale (Ariz.): Coriolis, c2001
- [19] Software project management / Bob Hughes and Mike Cotterell, London [etc.]: McGraw-Hill, c2002, 3rd ed.
- [20] Subversion version control: using the Subversion version control system in development projects / William Nagel, Upper Saddle River (N.J.): Prentice Hall/PTR, c2005