A Digital Approach to Mess Management System: The Role of Chilekotha

by

Md. Nure Alam
ID: CSE2201025181

Md. Parvez
ID: CSE2201025091

Md. Aibur Rahman
ID: CSE2201025024

Sajib Hasan
ID: CSE1903018144

Rokaya Akter
ID: CSE2201025087

Supervised by
Mohammad Naderuzzaman

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



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
SONARGAON UNIVERSITY (SU)
September 2025

A Digital Approach to Mess Management System: The Role of Chilekotha

by

Md. Nure Alam
ID: CSE2201025181

Md. Parvez
ID: CSE2201025091

Md. Aibur Rahman
ID: CSE2201025024

Sajib Hasan
ID: CSE1903018144

Rokaya Akter
ID: CSE2201025087

Supervised by
Mohammad Naderuzzaman

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



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
SONARGAON UNIVERSITY (SU)

September 2025

APPROVAL

The Project titled "**A Digital Approach to Mess Management System: The Role of Chilekotha**" submitted by Md. Nure Alam (CSE2201025181), Md. Parvez (CSE2201025091), Md. Aibur Rahman (CSE2201025024), Sajib Hasan (CSE1903018144) and Rokeya Akter (CSE2201025087) to the Department of Computer Science and Engineering, Sonargaon University (SU), has been accepted as satisfactory for the partial fulfillment 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

Mohammad Naderuzzaman — **Supervisor**
Associate Professor
Department of Computer Science and Engineering
Sonargaon University (SU)

(Examiner Name and Signature) — **Examiner 1**
Department of Computer Science and Engineering
Sonargaon University (SU)

(Examiner Name and Signature) — **Examiner 2**
Department of Computer Science and Engineering
Sonargaon University (SU)

(Examiner Name and Signature) — **Examiner 3**
Department of Computer Science and Engineering
Sonargaon University (SU)

DECLARATION

We, hereby, declare that the work presented in this report is the outcome of the investigation performed by us under the supervision of **Mohammad Naderuzzaman, Associate Professor,** 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
----------	----------
Mohammad Naderuzzaman	Md. Nure Alam
Supervisor	ID: CSE2201025181

	Md. Parvez
	ID: CSE2201025091

	Md. Aibur Rahman
	ID: CSE2201025024

	Sajib Hasan
	ID: CSE1903018144

	Rokaya Akter
	ID: CSE2201025087

ABSTRACT

This project, titled '**A Digital Approach to Mess Management System: The Role of Chilekotha**', is developed to simplify and modernize the lifestyle of living in a student mess or hostel is a common part of university and college life, but it often brings many challenges. Students who live together face issues such as food management, fair distribution of expenses, cleaning and duty schedules, and maintaining proper records of shared responsibilities. Misunderstandings and lack of coordination can lead to conflicts, stress, and an unorganized environment. These difficulties highlight the need for a structured approach to make collective living smoother and more efficient.

To address these problems, this project proposes **Chilekotha**, a conceptual framework for effective mess management. Chilekotha focuses on organizing the daily activities of student life in a transparent and systematic way. It aims to simplify common tasks such as meal planning, bazar lists, expense tracking, duty allocation, and communication among members. By using clear rules and organized methods, Chilekotha reduces confusion, increases accountability, and creates a sense of fairness within the mess.

By combining practical needs with modern management ideas, **Chilekotha** provides a framework that is easy to use, inclusive, and sustainable. Ultimately, it seeks to transform the traditional mess experience into a more harmonious, efficient, and student-friendly environment.

The system is implemented using React.js, Node.js, Express.js, and MongoDB. The expected outcome is efficient time management, easy calculation of expenses, and improved transparency among mess members.

In conclusion, Chilekotha is not only a management system but also a vision to improve student mess life. It provides a structured and innovative approach that transforms the challenges of collective living into opportunities for harmony, efficiency, and shared responsibility.

ACKNOWLEDGMENT

First and foremost, I am grateful to Almighty Allah for providing me with the strength, patience, and perseverance to complete this research work. I extend my heartfelt gratitude to my respected supervisor, **Mohammad Naderuzzaman**, Associate Professor, Department of Computer Science and Engineering, Sonargaon University, whose hearted and valuable support with the best concern and direction acted as a necessary recourse to carry out our project.

We would like to convey our special gratitude to **Prof. Bulbul Ahamed,** 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

CPU	Central Processing Unit
RAM	Random Access Memory
ROM	Read Only Memory
HTML	Hyper Text Markup Language
CSS	Cascading Style Sheets
JS	JavaScript
SQL	Structured Query Language
Mongo dB	Mongo Database
Express JS	Express JavaScript
Node JS	Node JavaScript
DFD	Data Flow Diagram
CRC	Class Responsibilities Collaboration

TABLE OF CONTENTS

Title A Digital Approach to Mess Management System: The Role of Chilekotha Page No.

DECLARATION ……..…….. ii

ABSTRACT ……...……...…..…….…. iii

ACKNOWLEDGEMENT ……..…… …….. iv

LIST OF ABBREVIATION …….. …….. v

CHAPTER 1 1-1

INTRODUCTION TO MESS MANAGEMENT SYSTEM

1.1 Project Overview …..……...……..... 1

1.2 Project Purpose ……...…..……. 1

1.2.1 Background ..……..……...……..……...… 1

1.2.2 Benefits ……..……..……..……..... 1

CHAPTER 2 2-5

SOFTWARE REQUIREMENT SPECIFICATION

2.1 Functional Requriements ……..……..…….…..... 2

2.2 Performance Requirements ……..……..…….…. 2

2.2.1 Speed and Latency Requirements ……. 2

2.2.2 Accuracy Requirements ……...…….…… 3

2.2.3 Capacity Requirements ……..…… 3

2.3 Dependability Requirements ……..……..... 3

2.3.1 Reliability Requirements ……..……...…… 3

2.3.2 Availability Requirements ……..……...……. 4

2.3.3 Robustness or Fault Tolerance Requirements ….……..…… 4

2.4 Maintainability and Supportability Requirements ….……...……..… 4

2.5 Maintenance Requirements ……..……. 4

2.5.1 Supportability Requirements ……. 4

2.6 Security Requirements ……..……. 4

2.6.1 Access Requirements …… 4

2.6.2 Integrity Requirements ……..……..……...…….. 4

2.6.3 Privacy Requirements ……..……. 4

2.7 Usability and Human-Interaction Requirements …….. 5

2.7.1 Ease of Use Requirements …..……. 5
2.7.2 Personalization and Internationalization Requirements …… 5
2.7.3 Understandability and Politeness Requirements……...…. 5
2.7.4 Accessibility Requirements ...……..…….. 5
2.7.5 User Documentation Requirements …….……. 5
2.7.6 Training Requirements ……..…… 5
2.8 Look and Feel Requirements …… 5
2.8.1 Apperance Requirements …...……...……. 5
2.8.2 Style Requirements ……... 5

CHAPTER 3 6-10

SYSTEM ANALYSIS

3.1 Use Case Diagram ….……..... 6
3.2 Use Case Description …….…...…….……..…. 6
3.2.1 Add Members ..…….……..… 6
3.2.2 Manage User …….…… 7
3.2.3 Manage Account ……..……..… 7
3.2.4 Manage Meal List …….…….…….. 7
3.2.5 Manage Bazar Schedule …….…….…….. 8
3.2.6 Manage Bazar List ……...…….…… 8
3.2.7 Manage Other Cost …….……..……. 8
3.3 Activity Diagram …….…… 9
3.3.1 Activity Diagram for Mess Manager Application …… 9
3.4 Data Flow Diagram …….…….…… 9
3.4.1 0 Level DFD …….…….…….…….…….. 10
3.4.2 1 Level DFD …….…….…… 10
3.4.3 2 Level DFD …….…….…… 10

CHAPTER 4 11-13

SYSTEM DESIGN SPECIFICATION

4.1 Class Responsibilities Collaboration (CRC) Cards ……..…. 11
4.2 Sequence Diagram ……...…… 11
4.3 Entity Relationship Diagram …….……... 12
4.4 Development Tools & Technology ……..……. 12
4.4.1 User Interface Technology ……...……..... 13

CHAPTER 5 14-17

SYSTEM TESTING

5.1 Testing Features ……..…….…….……. 14

5.1.1 Features to be Tested ……...……..…..…… 14

5.1.2 Features to not be Tested ..…… …… 14

5.2 Testing Strategies …….. 14

5.2.1 Test Approch ……..……...…….. 14

5.2.2 Pass/Fall Criteria ……...……. 14

5.2.3 Suspension & Resumption ……..……..…… 15

5.3 Testing Environment ……. 15

5.4 Test Case Description ……...…… 15

5.4.1 Test Case for Admin Login ……..……...… 16

5.4.2 Test Case for Add Bazar List ……..…….. 17

5.4.3 Test Case for Add Other Cost ……...……. 17

CHAPTER 6 18-28

USER MANUAL

6.1 Create Account ……..……..….. 18

6.1.1 Verify Account ……...……..…….……. 18

6.1.2 Create Profile ……..……..……. 19

6.2 Login ……. 20

6.2.1 Create a New Password ……..……... 20-21

6.3 Overview Home Pagen ……. 22

6.4 Meal Calendar …...…… 22

6.4.1 Meal Setting ……... 23

6.4.2 Auto and Last Meal Request Daily and Weakly ……....…….. 23

6.4.3 Meal Schedule Time ……..…….…….. 24

6.4.4 Set Meal List ….……...…….. 24

6.5 Bazar ……. 25

6.5.1 New Bazar List ……..……. 26

6.5.2 Money Receives ……..…….. 26

6.5.3 Add New Receives Entry ……..…… 27

6.6 Expense ….…… 27

6.6.1 Add New Expense with Chart ……...…… 28

6.6.2 Add New Expense Entry ……..……. 29
6.6.3 Add New Account ……..…… 30
6.7 Residents ……..……..…… 30
6.8 Conclusion …… 31

LIST OF TABLES

Table No.	Title	Page No.
Table 2.1	Functional Requirement	2
Table 2.2	Speed and Latency Requirements	2
Table 2.3	Accuracy Requirements	3
Table 2.4	Capacity Requirements	3
Table 2.5	Reliability Requirements	3
Table 2.6	Availability Requirements	4
Table 2.7	Robustness or Fault Tolerance Requirements	4
Table 3.1	Add members	6
Table 3.2	Use Case Description of Manage User	7
Table 3.3	Use Case Description of Manage Amount	7
Table 3.4	Use Case Description of Manage Meal-List	7
Table 3.5	Use Case Description of Manage Bazar Schedule	8
Table 3.6	Use Case Description of Manage Bazar List	8
Table 3.7	Use Case Description of Manage Other Cost	8
Table 4.1	Admin Responsibilities and Collaboration	11
Table 4.2	User Responsibilities and Collaboration	11
Table 5.1	Test Case for Admin Login	15
Table 5.2	Test Case for Member Add	15
Table 5.3	Test Case for Add Amount	16
Table 5.4	Test Case for Add Meal	16
Table 5.5	Test Case for Add Bazar Date	16
Table 5.6	Test Case for Add Bazar List	17
Table 5.7	Test Case for Add Others Cost	17

LIST OF FIGURES

Figure No.	Title	Page No.
Fig.3.1	Use Case Diagram	6
Fig.3.3.1	Activity Diagram	9
Fig.4.1	Sequence Diagram	11
Fig.4.2	Entity Relation Ship Diagram	12
Fig.6.1	Create Account	18
Fig.6.1.1	Verify Account	18
Fig.6.1.2	Create Profile	19
Fig.6.2	Login	20
Fig.6.2.1	Password identification	20-21
Fig.6.3	Overview Home page	22
Fig.6.4	Meal Calendar	22
Fig.6.4.1	Meal Setting	23
Fig.6.4.2	Auto and Last Meal Request Daily and Weakly	23
Fig.6.4.3	Meal Schedule Time	24
Fig.6.4.4	Set Meal List	24
Fig.6.5.1	Bazar List	25-26
Fig.6.5.2	Money Receives	26
Fig.6.5.3	New Receives Entry	27
Fig.6.6	Expense	27
Fig.6.6.1	New Expense Chart	28
Fig.6.6.2	Add New Expense Entry and View Dashboard	29
Fig.6.6.3	Add New Account	30
Fig.6.7	Residents	30

CHAPTER 1

INTRODUCTION TO MESS MANAGEMENT SYSTEM

1.1 Project Overview:

Mess Manager App is a powerful tool for simplifying and optimizing the management of mess facilities. It improves efficiency, reduces administrative work, and enhances the overall user experience. By providing features for menu planning, inventory management, meal distribution, payment tracking, and user feedback, the app empowers mess managers to deliver high-quality services and make data-driven decisions for continuous improvement

1.2 Project Purpose:

The main purpose of your project is to simplify menu planning, inventory management, meal distribution, and payment tracking processes.

Where manager can manage mess easily and efficiently.

1.2.1 Background:

The background of mess manager apps can be traced to the growing demand for improved management and organization of mess facilities. Traditional methods of managing mess operations, such as manual record-keeping, paper-based menu planning, and cash-based payment systems, were often time-consuming, error-prone, and inefficient. Mess manager apps have revolutionized the way mess facilities are managed, providing a modern and convenient solution for both administrators and end-users. With the advancement of technology and increasing demand for streamlined food service operations, the popularity and adoption of mess manager apps are likely to continue to grow.

1.2.2 Benefits:

There are several benefits that I have listed below.

1. Efficient management of mess operations, ordering, billing and inventory, leading to improved productivity and reduced administrative burden.

2. Enhanced transparency and convenience for users through features like real-time menu updates, digital ordering, and transparent billing and payment systems.

3. Ensures a fair distribution of responsibilities, reduces conflicts.

4. This system will reduce paper uses for us.

CHAPTER 2

SOFTWARE REQUIREMENT SPECIFICATION

Requirement analysis is the first step of identifying user needs in order to satisfy the users. This is the most important part of any software development.

2.1 Functional Requirements

Systems Functional Requirements are listed below:

Table 2.1: Functional Requirement

Id	Requirement Name	Description	Priority
01	Log in	Manager must have valid email and password to get into the system	High
02	Add User	Admin can add new user	Medium
03	Add/Remove Member	Admin can Add/Remove Member	High
04	Add/remove Money	Admin Add/Remove member Money	High
05	Add/remove meal	Admin Add/Remove meal	High
06	Add/remove Bazar Day	Admin Add/Remove Bazar Day	High
07	Add/remove Daily Bazar-List	Admin Add/Remove Bazar-List	High
08	Add/remove Other Cost	Admin Add/Remove Other Cost	High
09	Update Password	Admin update their own password	High

2.2 Performance Requirements

It's very necessary to sustain the performance of the project. To assure the better performance, this project has to meet some requirements which will provide the better performance.

2.2.1 Speed and Latency Requirements

While user will run our project in the browser, system needs a minimum amount of speed to perform the task.

Table 2.2: Speed and Latency Requirements:

SLR-01	Processing Speed of the System will be Faster.
Description	When user will run our system speed will be dependent on their internet speed & the server bandwidth speed.
Stakeholders	Users as Admin/Mess manager etc.

2.2.2 Accuracy Requirements

System has to confirm the Legibility and Accuracy of the data.

Table 2.3: Accuracy Requirements:

AR-01	Required Proper Information of Every User
Description	The input data should be correct and right pattern data, otherwise the input field show error message, like personal information, phone number, email address, password etc. the input information is not valid, the data never saved. Or the input data pattern is not match, then users will not be able to go forward step with the system.
Stakeholders	Users as student/mess member, Admin etc.

2.2.3 Capacity Requirements

The system should maintain all inserting data

Table 2.4: Capacity Requirements:

CR-01	Manages all the Data in Database Properly
Description	All data like, Users information data, Amount data, Meal data, Bazar Date data, Bazar list data, Other Cost data notice etc. are stored in the database.
Stakeholders	Users as student/mess member, Admin etc.

2.3 Dependability Requirements

Dependability means, it measures of a system reliability, availability, robustness, safety etc. Here, dependability means the running time of this project.

2.3.1 Reliability Requirements

Table 2.5: Reliability Requirements:

RR-01	The system is reliable.
Description	User information, Amount information, Bazar list information, time schedule and others information will be safe and secured in our system.
Stakeholder	Users as student/mess member, admin etc.

2.3.2 Availability Requirements

Table 2.6: Availability Requirements:

AR-01	The system will be available 24x7.
Description	It is available 24 hours in a day and 7 days in a week. And The system will be updated regularly.
Stakeholder	Users as student/mess member, admin etc.

2.3.3 Robustness or Fault Tolerance Requirements

Table 2.7: Robustness or Fault Tolerance Requirements:

FTR-01	Well robustness of the system
Description	If any problem occurs the system will show error message and the fault tolerance are handled properly.
Stakeholder	Users as student/mess member, admin etc.

2.4 Maintainability and supportability requirements

Authority would have assigned some Admins and specialist for the maintenance of the system and support the system.

2.5 Maintenance Requirements

Authority would have to assign some specialists for maintaining the system. System can produce wrong results and the specialists must be able to reproduce the data flow through the system.

2.5.1 Supportability Requirements

Authority would have to recruit some admins for supportability of the system.

2.6 Security requirements

There are two securities to get access to this system or a specific module of the system must provide an authentication mechanism.

2.6.1 Access Requirements

This system provides accesses the different module, by access in an authentication way to the authentic user.

2.6.2 Integrity Requirements

Credential information of users will be kept safe.

2.6.3 Privacy Requirements

All the passwords and personal information will be kept safe and securely in the system.

2.7 Usability and human-interaction requirements

This system is very user friendly with simple user-interface.

2.7.1 Ease of Use Requirements

Users can use the system very easily with satisfaction and efficiency.

2.7.2 Personalization and Internationalization Requirements

I have developed this system in order to help the Software department sports event. So, there are no internationalization requirements.

2.7.3 Understandability and politeness Requirements

This system is very easy to use and understand with simple user-interface.

2.7.4 Accessibility Requirements

This system is very easily accessible for definite user in definite module.

2.7.5 User Documentation Requirements

Events information and user records will be stored and kept in the system properly.

2.7.6 Training Requirements

There is not needed any extensive training for operating this digital system.

2.8 Look and feel Requirements

According to client's expectation look means the graphical user interface, design, layout, shapes, colors etc. and feel means the behavior of different elements like menu bar, buttons etc.

2.8.1 Appearance Requirements

Needed knowledge about Node.js, MongoDB.

2.8.2 Style Requirements

Knowledge about HTML, Tailwind CSS, JavaScript and ReactJS.

CHAPTER 3

SYSTEM ANALYSIS

3.1 Use Case Diagram:

Use case Diagram is the simplest way of representing in a graphical representation of a complex systems user interactions among the elements of the system. Sports Event Reminders use case diagram is given below:

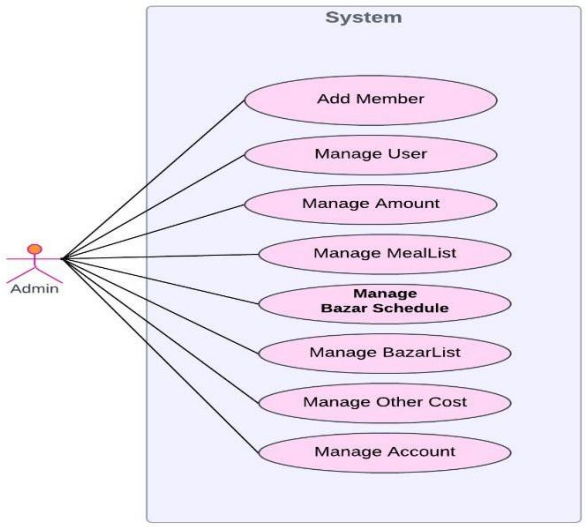


Figure 3.1: Use Case Diagram (adapted from lucid.app, 2025)

3.2 Use Case Description:

This part will describe about the task briefly about how the users needed to interact with the system.

3.2.1 Add Members:

Table 3.1: Add members of User Authentication:

Use case name	Add members	
Scenario	Admin added members	
Brief	Manager/Admin must create user account. Here stores all	
Actor	Admin/Manager	
Preconditions	Name, Email, Mobile Number etc.	
Post conditions	New User must be added.	
Flow of events	Actor	System
	Admin	Account Created successfully
Failure results	If system not response, then show an error message.	

3.2.2 Manage User:

Table 3.2: Use Case Description of Manage User:

Use case name	Manage User	
Scenario	Admin have to manage the user list.	
Brief	Admin will have the privileges to manage the user such as check the user list, delete /remove user from the list, block user etc.	
Actor	Admin	
Preconditions	Admin must exist.	
Post conditions	User list will be displayed	
Flow of events	Actor	System
	Enter Email and Password	Response with displaying proper information.
Failure results	If system not response, then show an error message.	

3.2.3 Manage Amount:

Table 3.3: Use Case Description of Manage Amount:

Use case name	Manage Amount	
Scenario	Admin have to manage the Amount list.	
Brief	Admin will have the privileges to manage the user amount such as check the user amount list, delete /remove user amount from the list etc.	
Actor	Admin	
Preconditions	Admin must exist.	
Post conditions	User amount list will be displayed	
Flow of events	Actor	System
	Manager	Response with displaying proper information.
Failure results	If system not response, then show an error message.	

3.2.4 Manage Meal List:

Table 3.4: Use Case Description of Manage Meal-List:

Use case name		Manage Meal-List	
Scenario		Admin have to manage Meal-List.	
Brief		Admin will have the privileges to manage Meal-List such as How much meal each one eats.	
Actor		Admin	
Preconditions		Admin must exist.	
Post conditions		Meal-List will be displayed	
Flow of events		Actor	System
		Admin	Response with displaying proper information.
Failure results		If system not response, then show an error message.	

3.2.5 Manage Bazar Schedule

Table 3.5: Use Case Description of Manage Bazar Schedule:

Use case name	Manage Bazar Schedule	
Scenario	Admin have to manage bazar schedule.	
Brief	Admin will have the privileges to manage bazar day schedule such as check Who will market this month.	
Actor	Admin	
Preconditions	Admin must exist.	
Post conditions	Bazar schedule list will be displayed	
Flow of events	Actor	System
	Admin	Response with displaying proper information.
Failure results	If system not response, then show an error message.	

3.2.6 Manage Bazar List

Table 3.6: Use Case Description of Manage Bazar List:

Use case name	Manage Bazar List	
Scenario	Admin have to manage bazar list.	
Brief	Admin will have the privileges to manage bazar list schedule such as the members will do the market and then the admin will keep the list of that market according to the day.	
Actor	Admin	
Preconditions	Admin must exist.	
Post conditions	Bazar list will be displayed	
Flow of events	Actor	System
	Admin	Response with displaying proper information.
Failure results	If system not response, then show an error message.	

3.2.7 Manage Other Cost:

Table 3.7: Use Case Description of Manage Other Cost:

Use case name	Manage Other Cost	
Scenario	Admin have to manage other cost	
Brief	Admin will have the privileges to manage the other cost such as check the user other cost list, delete /remove other cost from the list etc..	
Actor	Admin	
Preconditions	Admin must exist.	
Post conditions	Other cost list will be displayed	
Flow of events	Actor	System
	Admin	Response with displaying proper information.
Failure results	If system not response, then show an error message.	

3.3 Activity Diagram

In Unified Modeling Language Activity diagram is an important diagram. Basically, it is a flow chart. It can describe the flow of the activity in a graphical representation.

3.3.1 Activity Diagram for Mess Manager Application

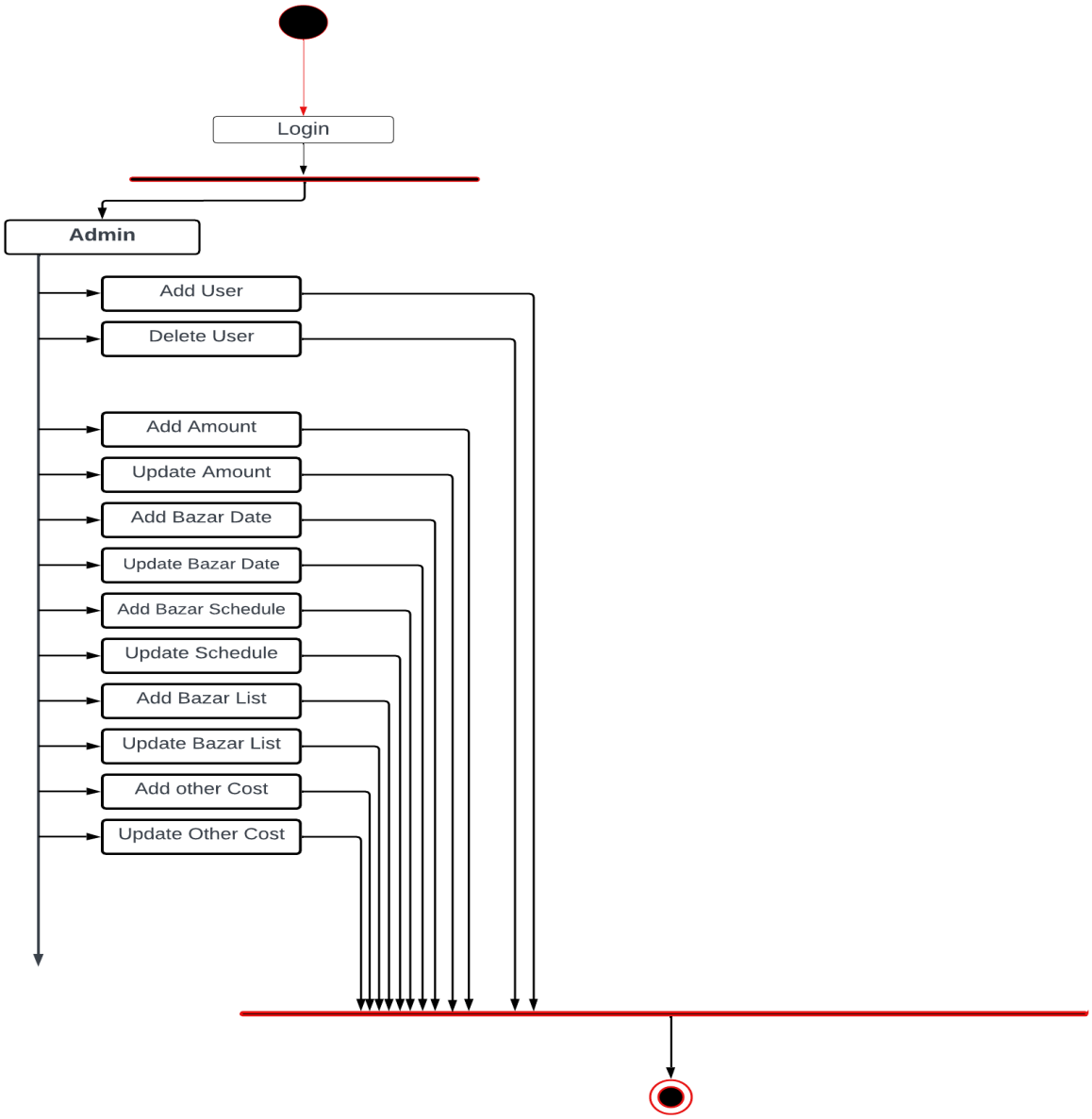
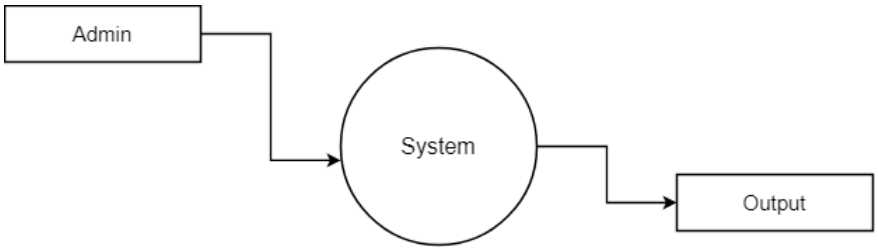


Figure 3.3.1: Activity Diagram (adapted from lucid.app, 2025)

3.4 Data Flow Diagram

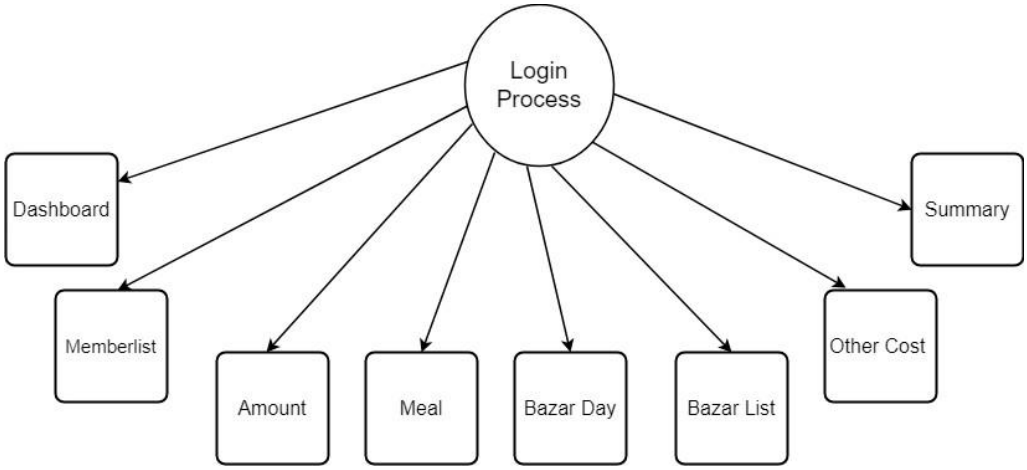
A data flow diagram (DFD) maps out the flow of information for any process or system. It uses defined symbols like rectangles, circles and arrows, plus short text labels, to show data inputs, outputs, storage points and the routes between each destination.

3.4.1 0-Level DFD



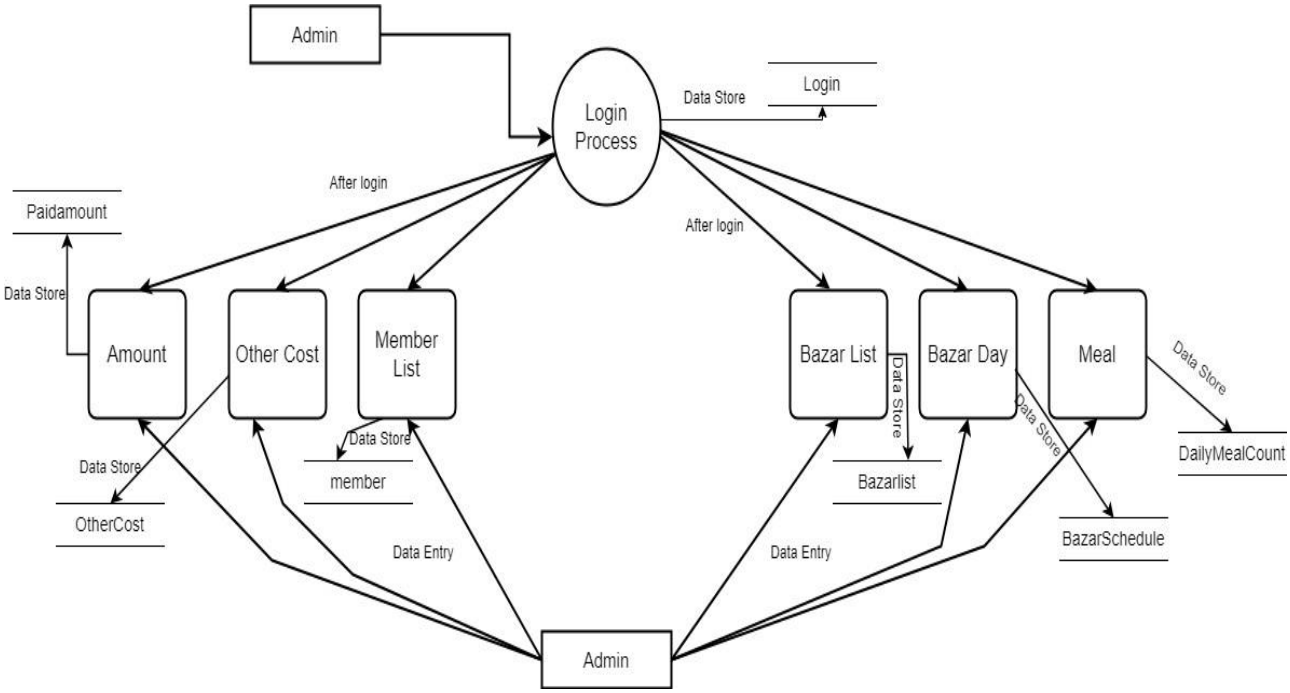
Source: lucid.app, 2025 (adapted)

3.4.2 1-Level DFD



Source: lucid.app, 2025 (adapted)

3.4.3 2-Level DFD:



Source: lucid.app, 2025 (adapted)

CHAPTER 4

SYSTEM DESIGN SPECIFICATION

4.1 Class Responsibilities Collaboration (CRC) Cards

It's a brainstorming tool used to design any kind of object-oriented software. Following tables are given blew:

Admin:

Table 4.1: Admin Responsibilities and collaboration:

Responsibilities	Collaboration
Monitoring the system	Manage Account, Manage Amount, Manage Bazar Schedule.

User/Participants:

Table 4.2: User Responsibilities and collaboration:

Responsibilities	Collaboration
Log in with proper information.	Authentication Process

4.2 Sequence Diagram

Sequence diagram is another very important UML diagram of a software. Sequence diagram can describe the working process of a system as how it works and what order is following it to complete a process.

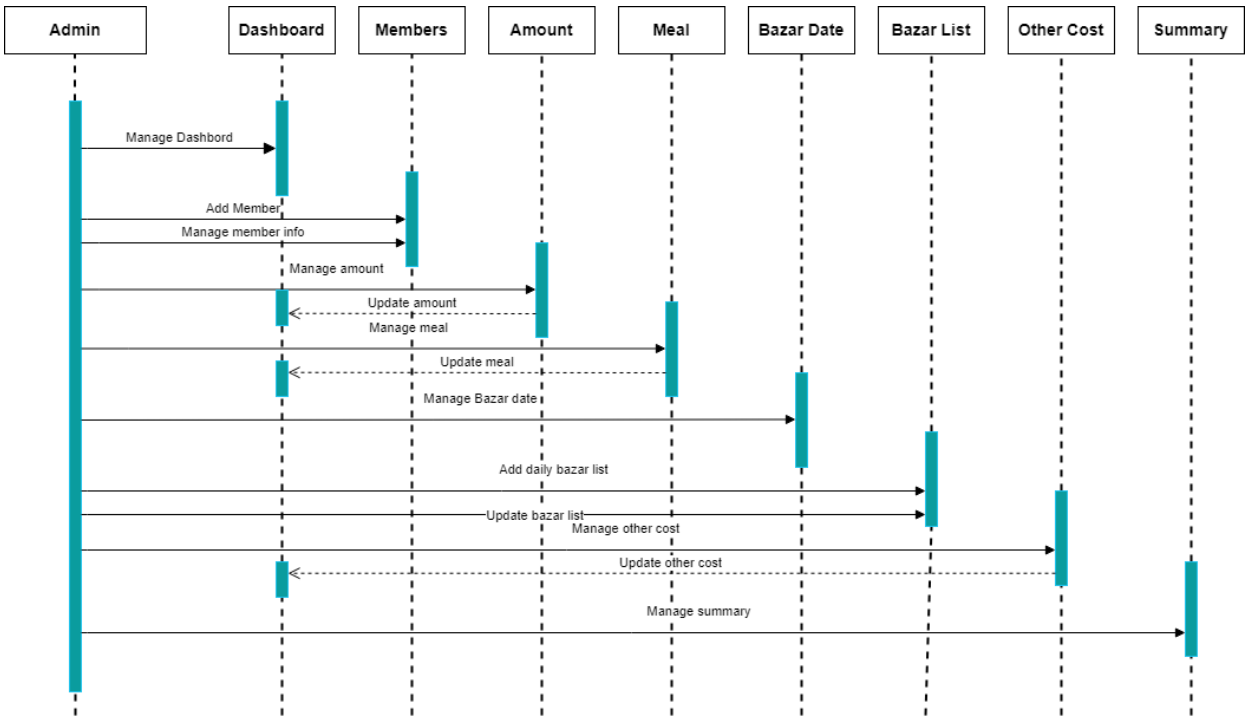


Figure 4.1: Sequence Diagram (adapted from diagrams.net, 2025)

4.3 Entity Relationship Diagram

Database is the most important part of any system. We have tried to design a dynamic database for our system. Our database design diagram is given below:

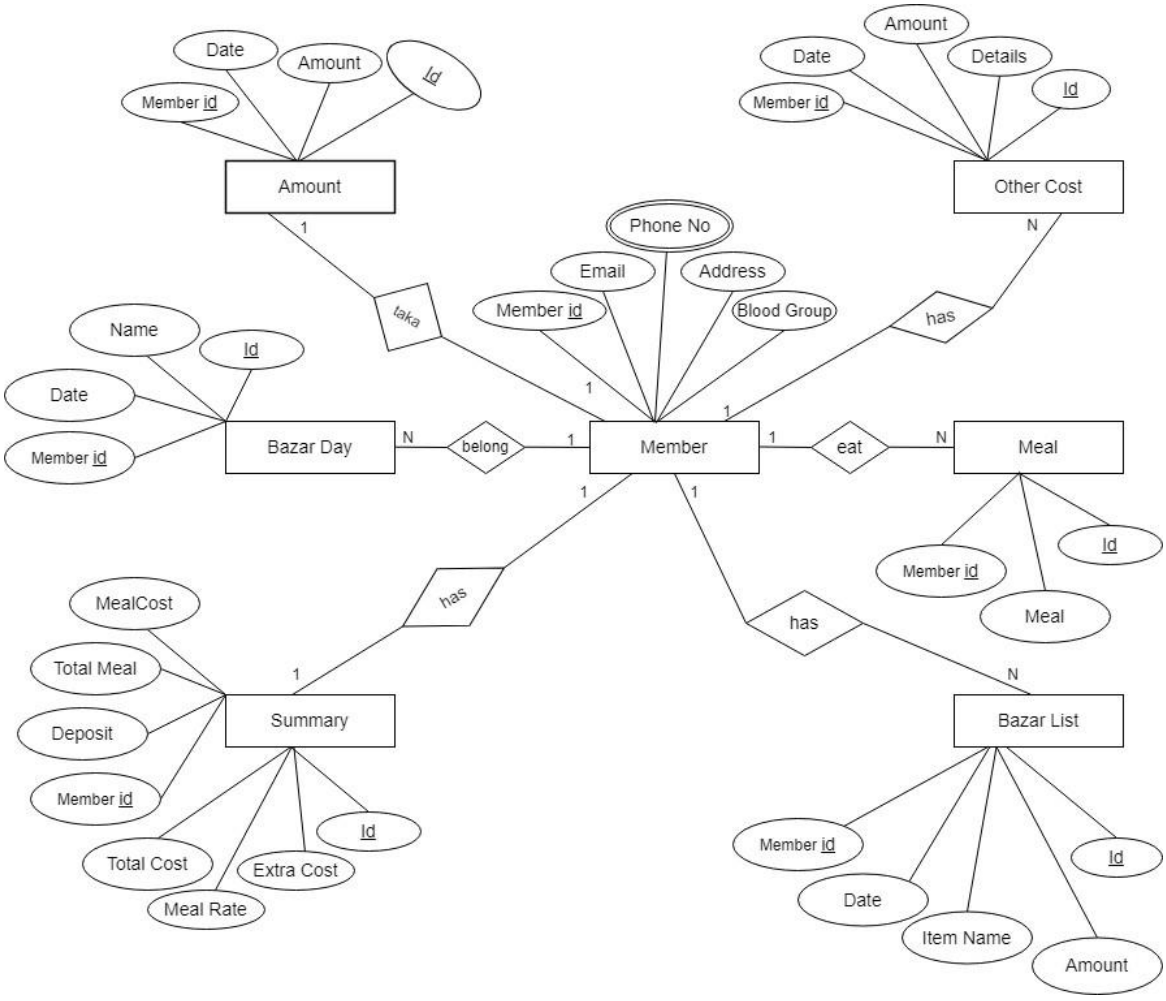


Figure 4.2: Entity Relation Ship Diagram (adapted from diagrams.net, 2025)

4.4 Development Tools & Technology

For developing a software system, selecting development tools and technology is a vital fact and challenge.

Development Tools

Server: MongoDB

Operating System: Windows 10

Development Technology

Programming Language: Node.js

4.4.1 User Interface Technology

- **HTML**
  The backbone of any website. Defines the **structure** of web pages (headings, paragraphs, tables, forms). In this project, HTML is used for creating forms (login, signup, meal entry) and organizing content.

- **Tailwind CSS**
  Utility-first modern styling with flexibility. Ensures the site is visually attractive and easy to use. In this project: Tailwind CSS is used for dashboard layout, color themes, and responsive design.

- **JavaScript**
  The **programming language of the web**. Adds **interactivity** (validating forms, showing notifications, updating data dynamically). Example in this project: Live expense calculation, meal count updates, and dynamic charts.

- **React.js**
  Used for building component-based user interfaces. Example in this project: Dashboard, expense charts, and meal forms are all React components.

- **Mongo dB**
  MongoDB will store all mess-related data (users, meals, expenses, duties) in a fast, flexible, and scalable way.

- **Node JS**
  Node.js is the backend engine of Chilekotha. It connects the frontend **(**React**)** with the database (MongoDB**)**, handles user requests, stores data, and ensures smooth performance. With Express, it creates powerful APIs to manage meals, expenses, and duties.

- **Express JS**
  Express.js is the heart of the backend. It defines how requests (like adding meals, calculating expenses, assigning duties) are handled, and ensures smooth communication between React (frontend) and MongoDB (database)**.**

CHAPTER 5

SYSTEM TESTING

5.1 Testing Features

Testing is one of the most important parts for checking the functionality of a software or system. Testing every feature of a software we can ensure that the software or system is ready to provide its service.

5.1.1 Features to be Tested

There is given a list of the features we have tested

- Admin Login
- Manage User
- Manage Amount
- Manage Meal Count
- Manage Bazar Schedule
- Manage Bazar List
- Manage Other Cost
- Manage Summary

5.1.2 Features not to be Tested

We have tested every field of our system.

5.2 Testing Strategies

There is given the strategies of our system or software testing.

- Measure test
- Quality test

5.2.1 Test Approach:

To implementation a project test approach is very important. Test approach has two types of techniques:

Proactive - An approach in which the test design process is initiated as early as possible in order to find and fix the defects before the build is created.

Reactive - An approach in which the testing is not started until after design and coding are completed.

5.2.2 Pass/Fail Criteria:

- When the expected result is come after testing then it is called pass
- When unexpected result is come during testing then it is called fail.

5.2.3 Suspension & Resumption:

Our system is very smooth and easy to use and maintain. There is no suspension and resumption problem.

5.3 Testing Environment:

- Testing IDE: Visual Studio Code
- Browser: Google Chrome
- Server: MongoDB
- Operating Systems: Windows 10
- Possessor: CORE i5

5.4 Test Case Description:

We have tested our system many times. Now we are focusing on that our system will execute properly.

Table 5.1: Test Case for Admin Login

Test Scenario		Admin Login	Test Case ID	TC01	
Test Case Description		Login test case	Test Priority	High	
Pre-Requisite		A valid user account	Post-Requisite	NA	
Test Execution Steps:					
SL No	Action	Inputs	Expected Result	Actual Result	Test Result
01	Enter Correct Email and password	Email: nadeem@gmail.com Password: ******	Login Successful	Login Successful	Pass

Table 5.2 Test Case for Member Add

Test Scenario		Add member	Test Case ID	TC02	
Test Case Description		Member added in mess app	Test Priority	High	
Pre-Requisite		A user authentication	Post-Requisite	NA	
Test Execution Steps:					
SL. No	Action	Inputs	Expected Result	Actual Result	Test Result
01	Enter Name, Email Address Phone, and Blood group	Nadeem nadeem@gmail.com 01537515034 O+	Amount added Successful	Added Successful	Pass

Table 5.3 Test Case for Add Amount

Test Scenario		Add amount		Test Case ID	TC03
Test Case Description		Login test case		Test Priority	High
Pre-Requisite		A valid user info		Post-Requisite	NA
Test Execution Steps:					
SL No	Action	Inputs	Expected Result	Actual Result	Test Result
01	Enter Name And Amount	Nadeem 1500tk	Amount added Successful	Added Successful	Pass

Table 5.4: Test Case for Add meal

Test Scenario		Add meal		Test Case ID	TC04
Test Case Description		Add meal		Test Priority	High
Pre-Requisite		An authenticated Admin		Post-Requisite	NA
Test Execution Steps:					
SL No	Action	Inputs	Expected Result	Actual Result	Test Result
01	Enter Meal, Date and Name	Meal:3 Date:23-5-23 Md. Forid	Successful	Successful	Pass

5.4.1 Test Case for Add Bazar Date:

Table 5.5: Test Case for Bazar Date

Test Scenario		Add Bazar Date		Test Case ID	TC04
Test Case Description		Add Bazar Date		Test Priority	High
Pre-Requisite		An authenticated Admin		Post-Requisite	NA
Test Execution Steps:					
SL No	Action	Inputs	Expected Result	Actual Result	Test Result
01	Enter Date and Name	Date:23-5-23 Md. Forid	Successful	Successful	Pass

5.4.2 Test Case for Add Bazar List:

Table 5.6: Test Case for Add Bazar List

Test Scenario		Add Bazar List	Test Case ID	TC04	
Test Case Description		Add Bazar List	Test Priority	High	
Pre-Requisite		An authenticated Admin	Post-Requisite	NA	
Test Execution Steps:					
SL. No	Action	Inputs	Expected Result	Actual Result	Test Result
01	Enter Date Details Amount and Name	Date:23-5-23 Fish-1kg 280 taka Md.Forid	Successful	Successful	Pass

5.4.3 Test Case for Add Other Cost

Table 5.7: Test Case for Add Other Cost

Test Scenario		Add Other cost	Test Case ID	TC04	
Test Case Description		Add Other Cost	Test Priority	High	
Pre-Requisite		An authenticated Admin	Post-Requisite	NA	
Test Execution Steps:					
SL No	Action	Inputs	Expected Result	Actual Result	Test Result
01	Amount Date and Details Enter Name	Light-1pc 120tk 23-5-23 Md. Forid	Successful	Successful	Pass

CHAPTER 6

USER MANUAL

6.1 Create Account

A mess account is a record of all financial transactions related to the daily meals system of a hostel, dormitory, or shared accommodation where members contribute to food expenses.

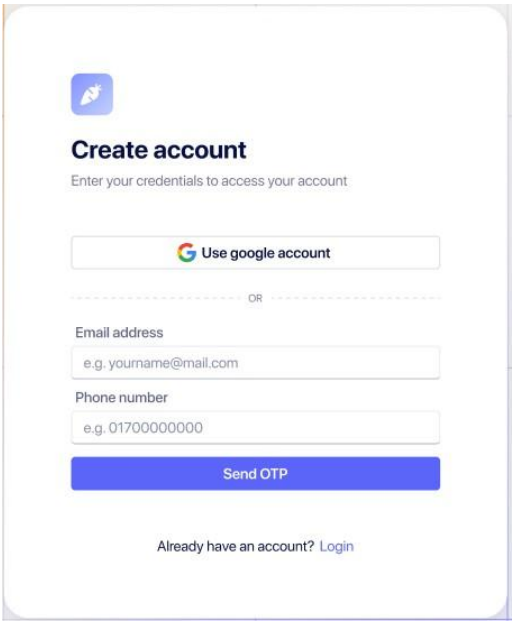


Figure 6.1: Create Account (adapted from figma.com, 2025)

6.1.1 Verify Account

Verification means checking whether the income, expenses, and balances recorded in the mess account are accurate, complete, and transparent

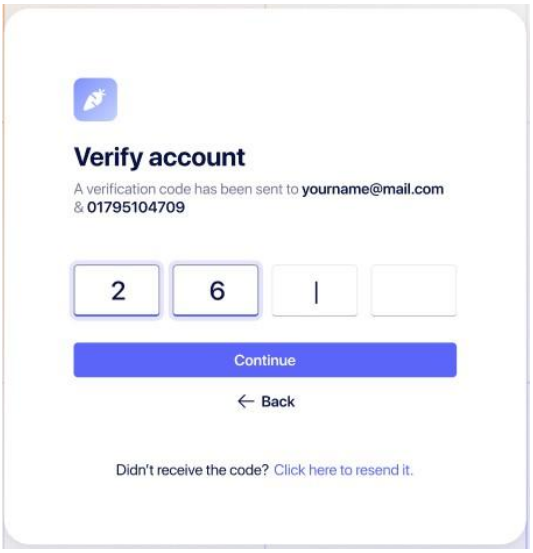


Figure 6.1.1: Verify Account (adapted from figma.com, 2025)

6.1.2 Create Profile

A profile is simply a structured record of someone's personal, professional details that can be used for identification, communication, or reference.

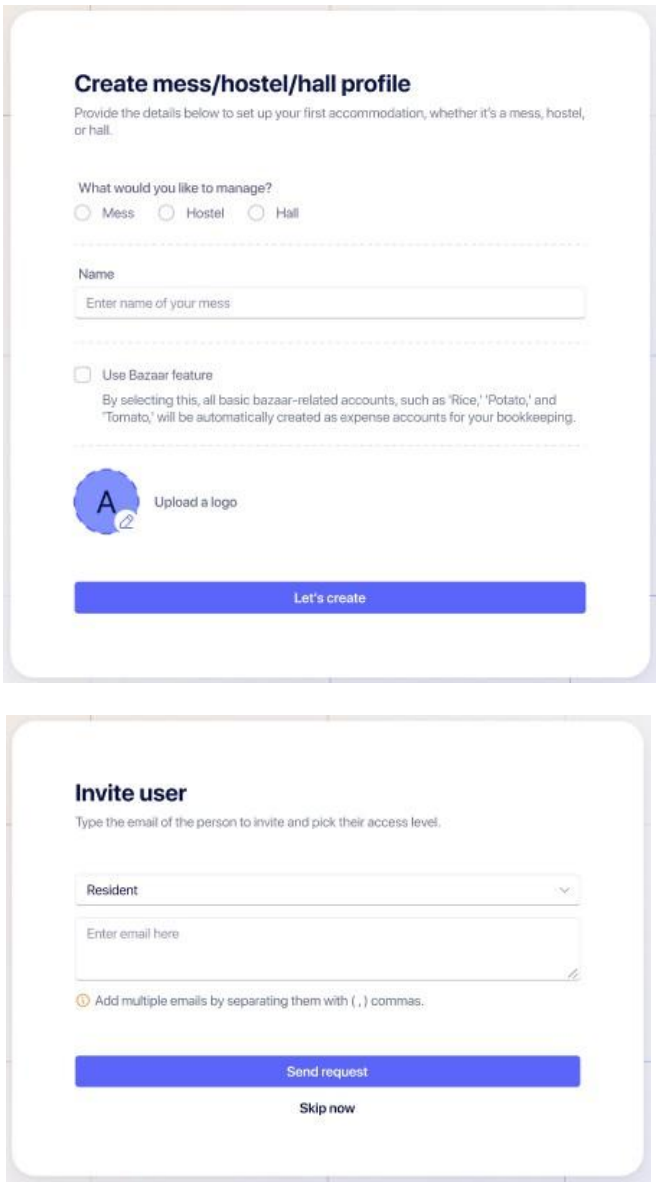


Figure 6.1.2: Create Profile (adapted from figma.com, 2025)

6.2 Login

Login is the process of accessing a computer system, website, or application by providing valid credentials

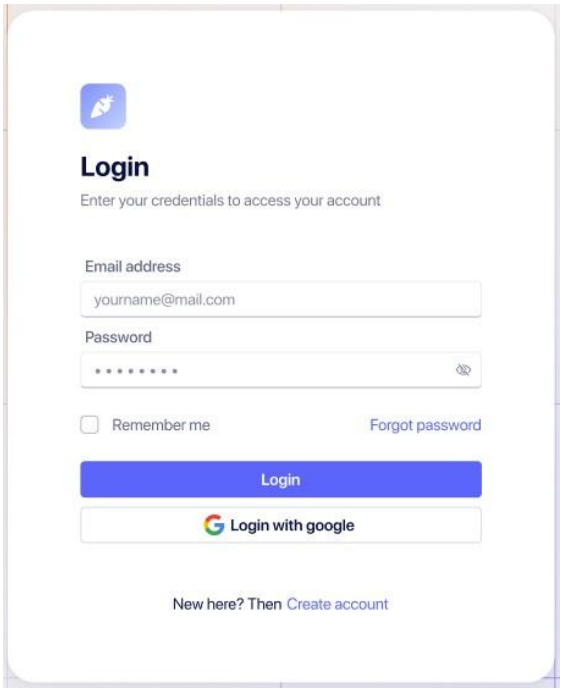
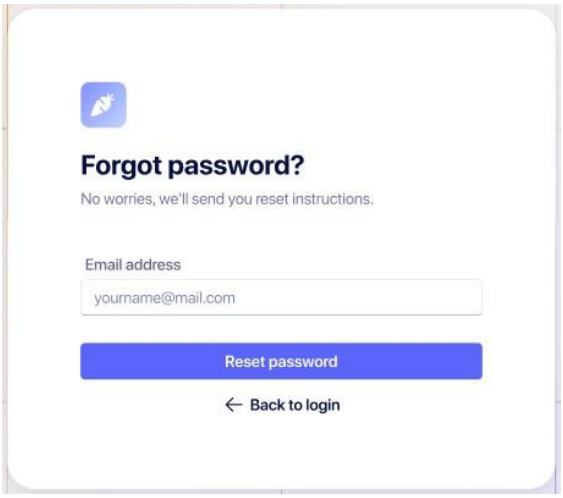


Figure 6.2: Login (adapted from figma.com, 2025)

6.2.1 Create a New Password

A **password** is a secret word, phrase, or combination of characters used to verify a user's identity during the **login process**.



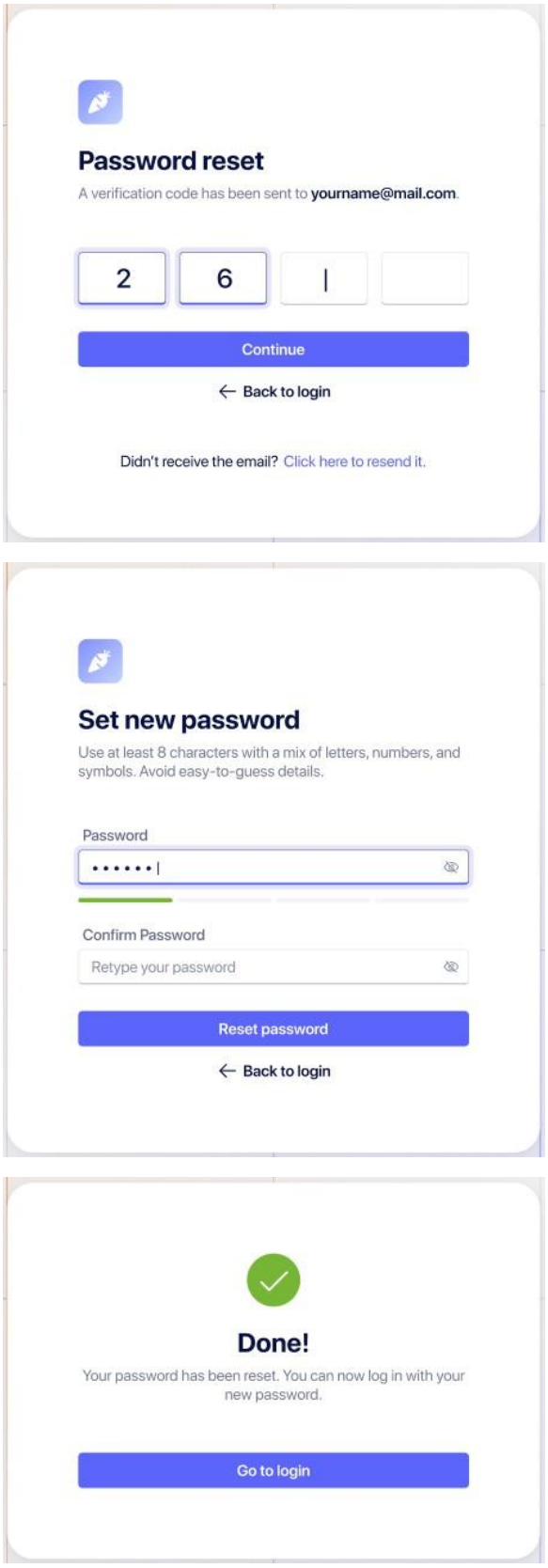


Figure 6.2.1: Password Identification (adapted from figma.com, 2025)

6.3 Overview Home Page

A Home Page is the main or first page of a website or application that appears when a user opens it.

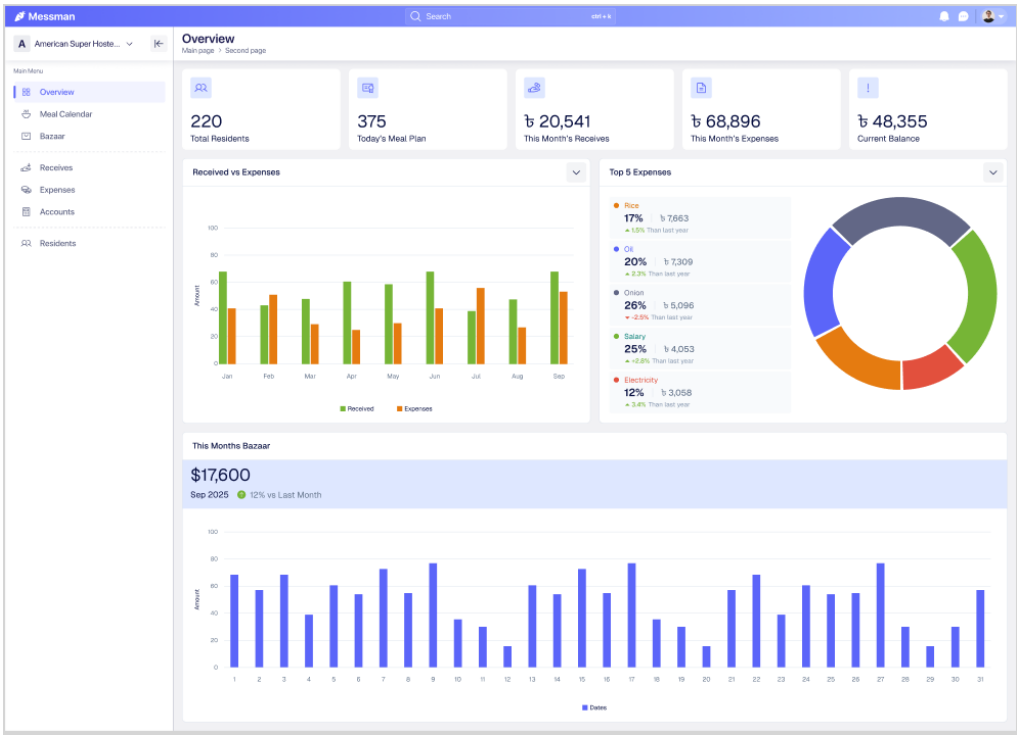


Figure 6.3: Overview Home Page (adapted from figma.com, 2025)

6.4 Meal Calendar

A Meal Calendar is a schedule that shows the daily or weekly plan of meals for members in a hostel, mess, or cafeteria

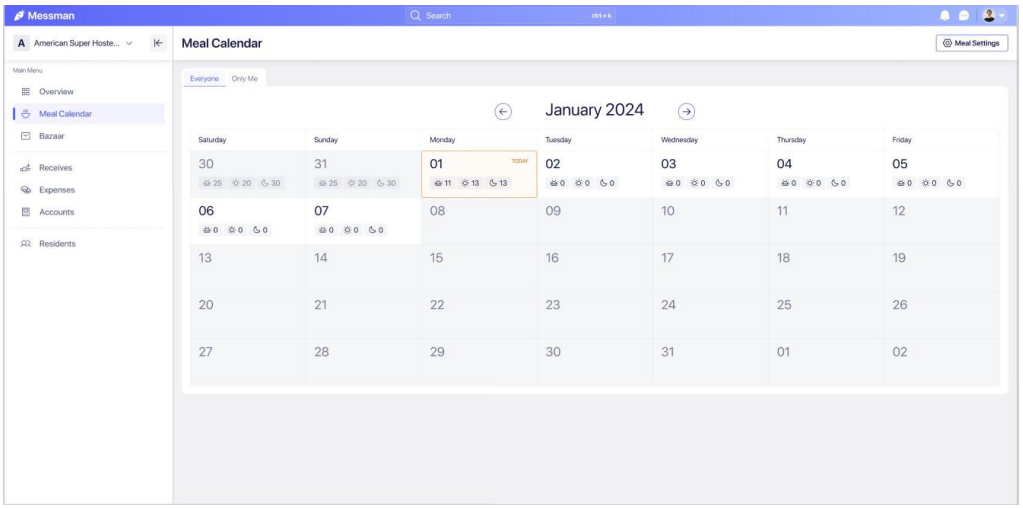


Figure 6.4: Meal Calendar (adapted from figma.com, 2025)

6.4.1 Meal Settings

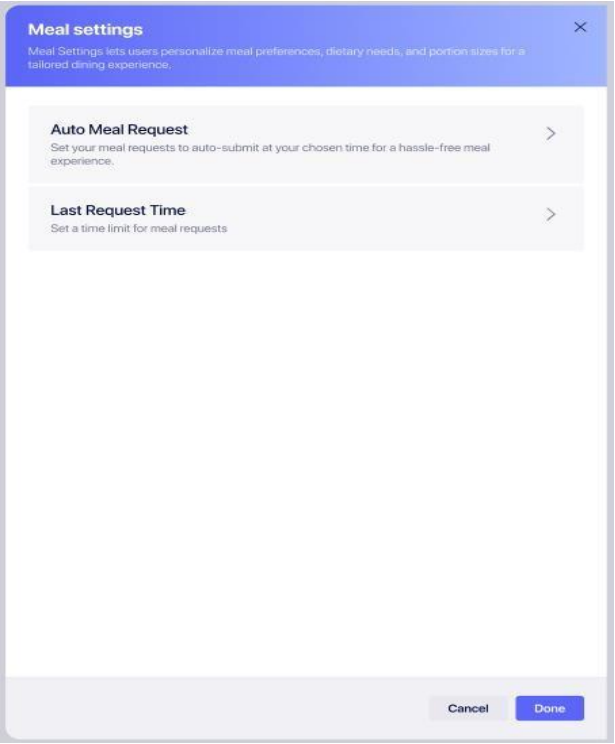


Figure 6.4.1: Meal Settings (adapted from figma.com, 2025)

6.4.2 Auto and Last Meal Request Daily and Weakly

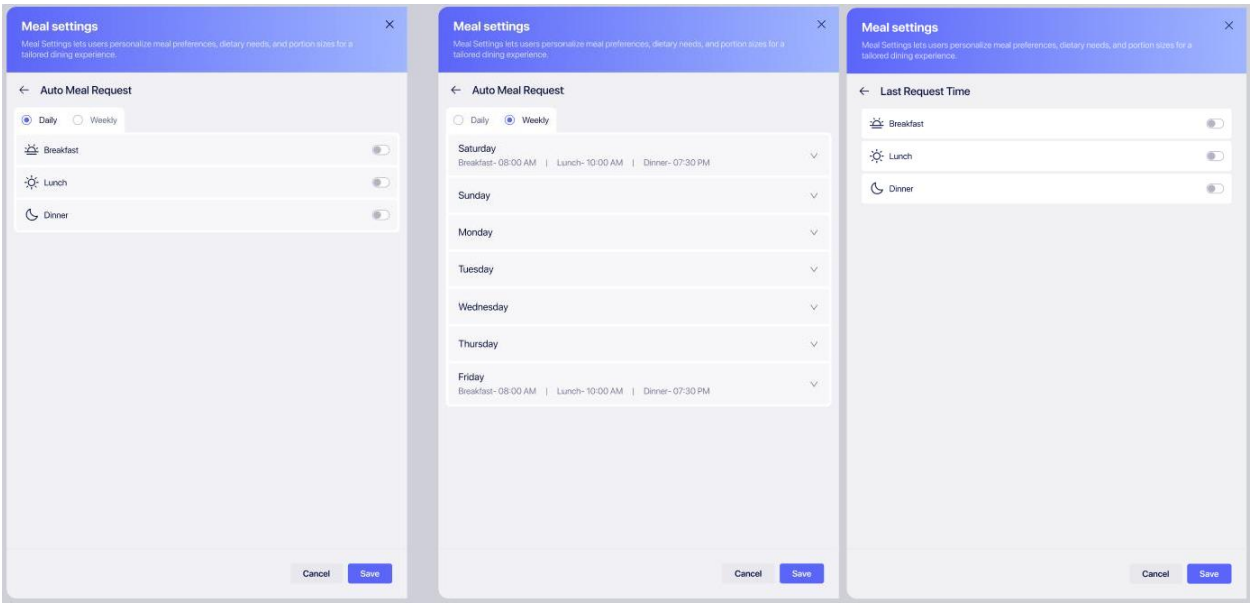


Figure 6.4.2: Auto and Last Meal Request Daily and Weakly
(adapted from figma.com, 2025)

6.4.3 Meal Schedule Time

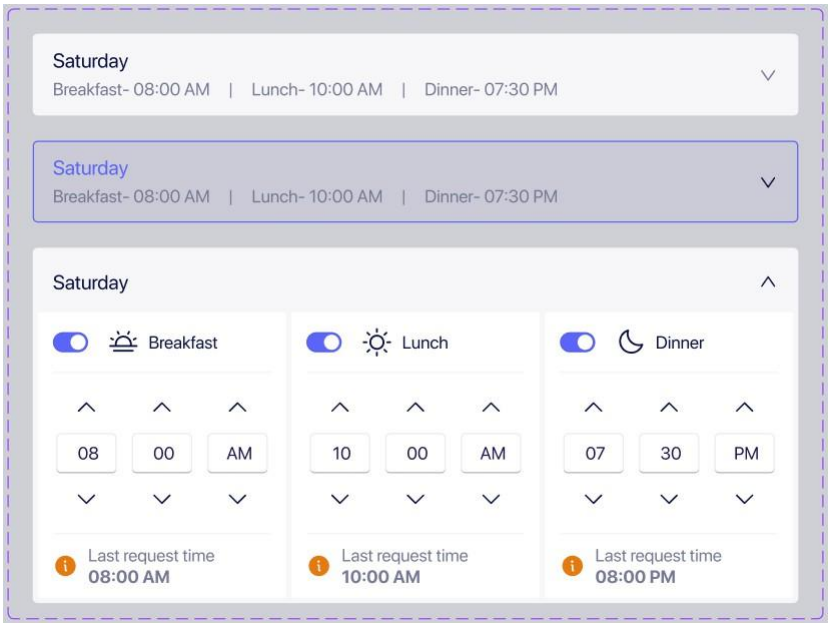


Figure 6.4.3: Meal Schedule Time (adapted from figma.com, 2025)

6.4.4 Set Meal List

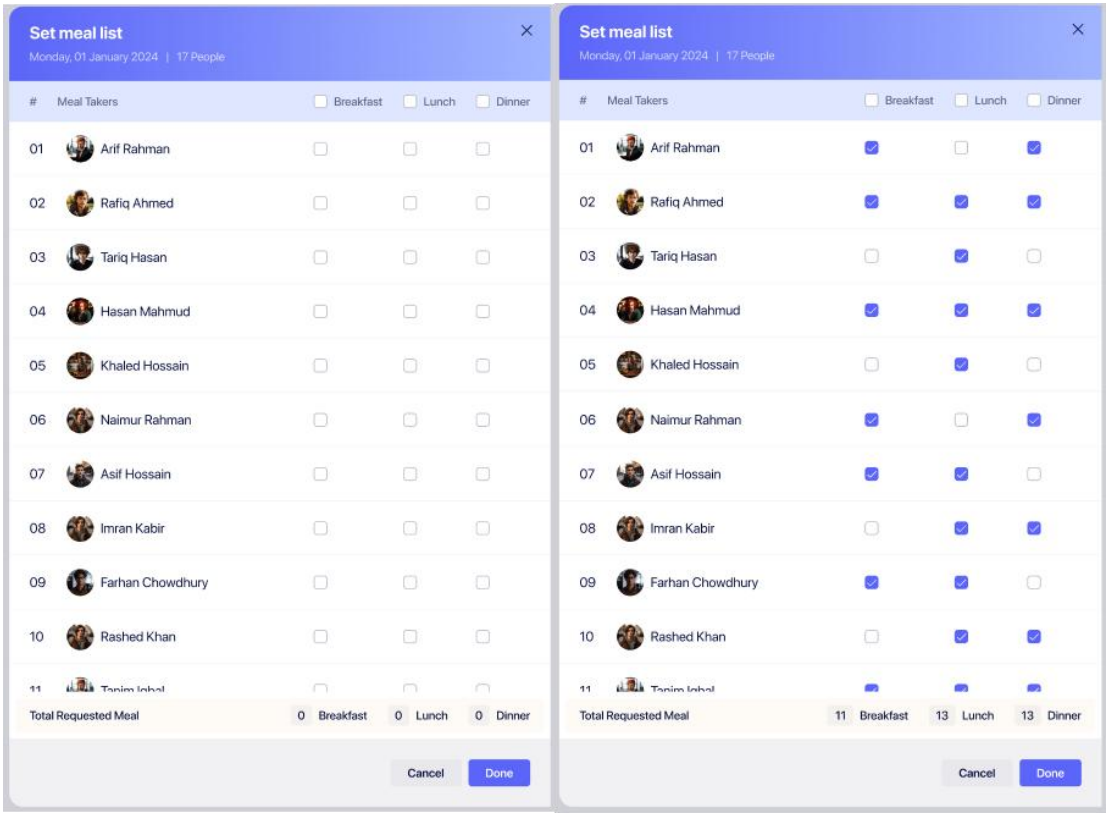
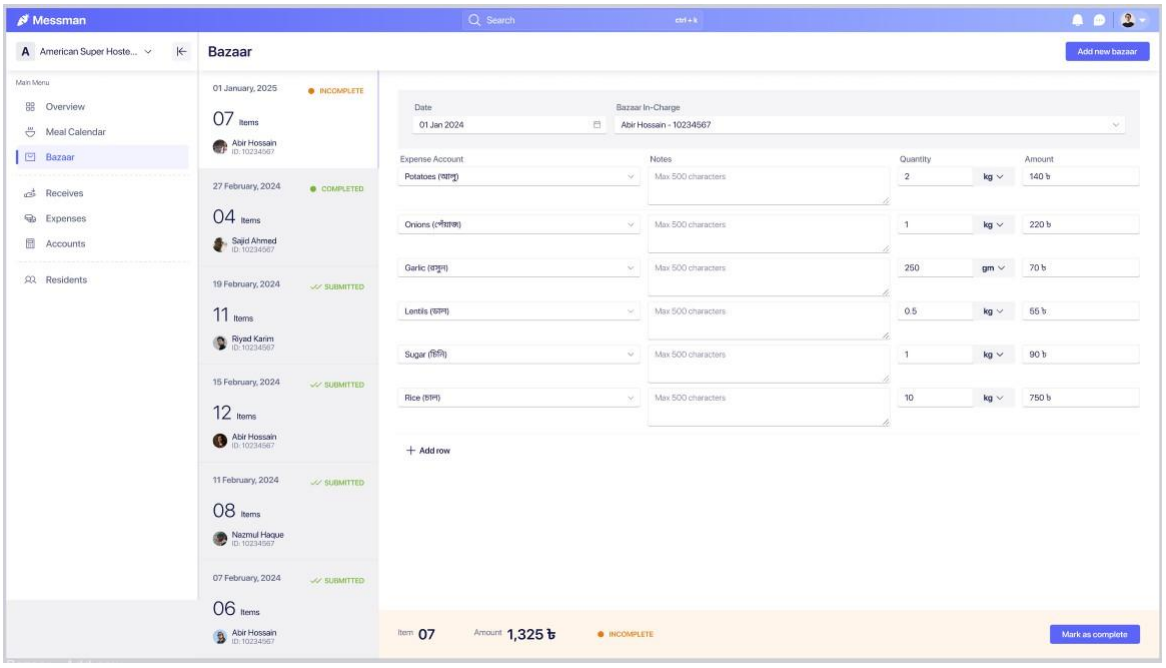
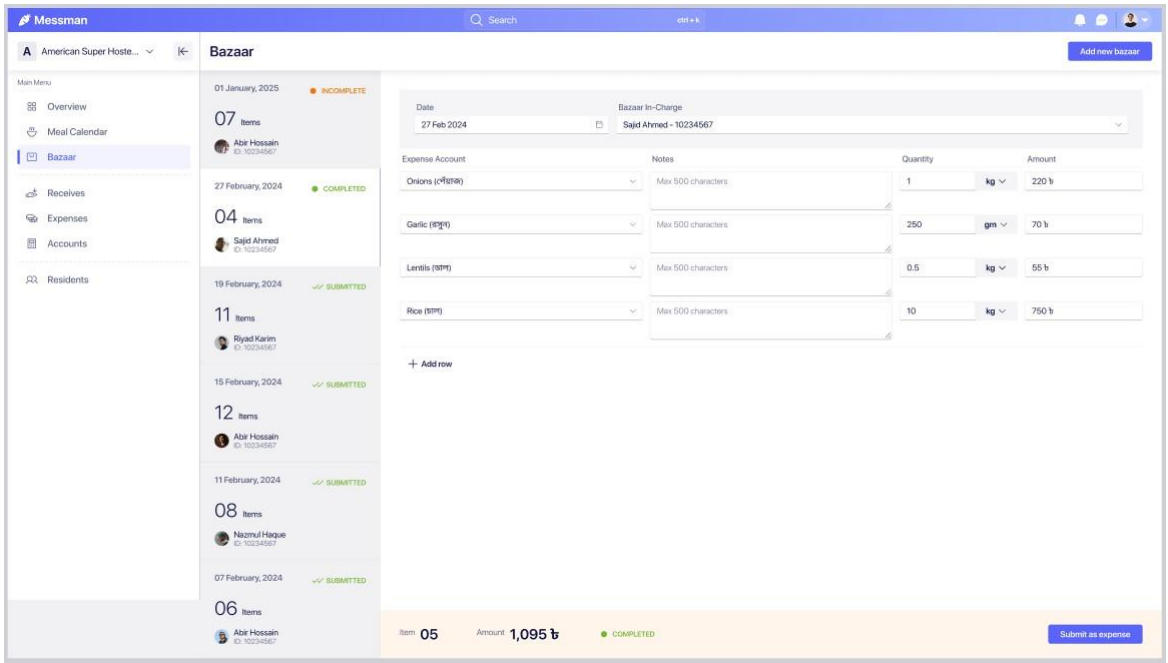


Figure 6.4.4: Set Meal List (adapted from figma.com, 2025)

6.5 Bazar List

The Bazar-day list of each month are recorded in this page.





6.5.1 New Bazar List

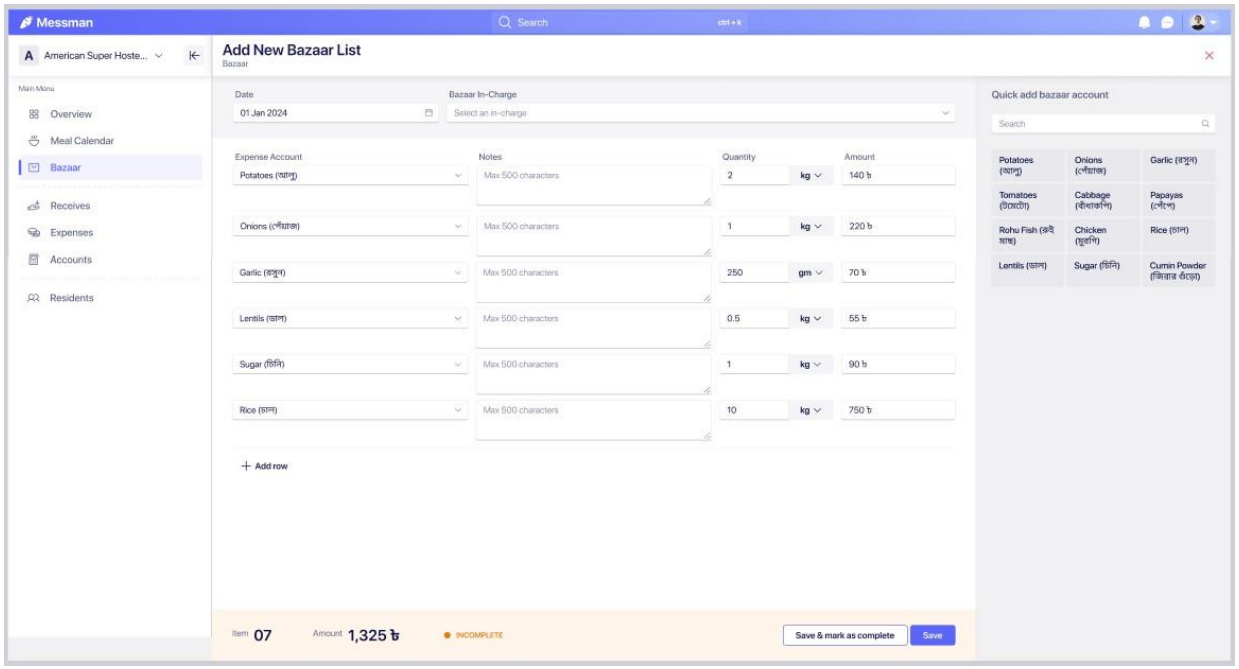


Figure 6.5.1: Bazar List (adapted from figma.com, 2025)

6.5.2 Money Receives

This table are received money.

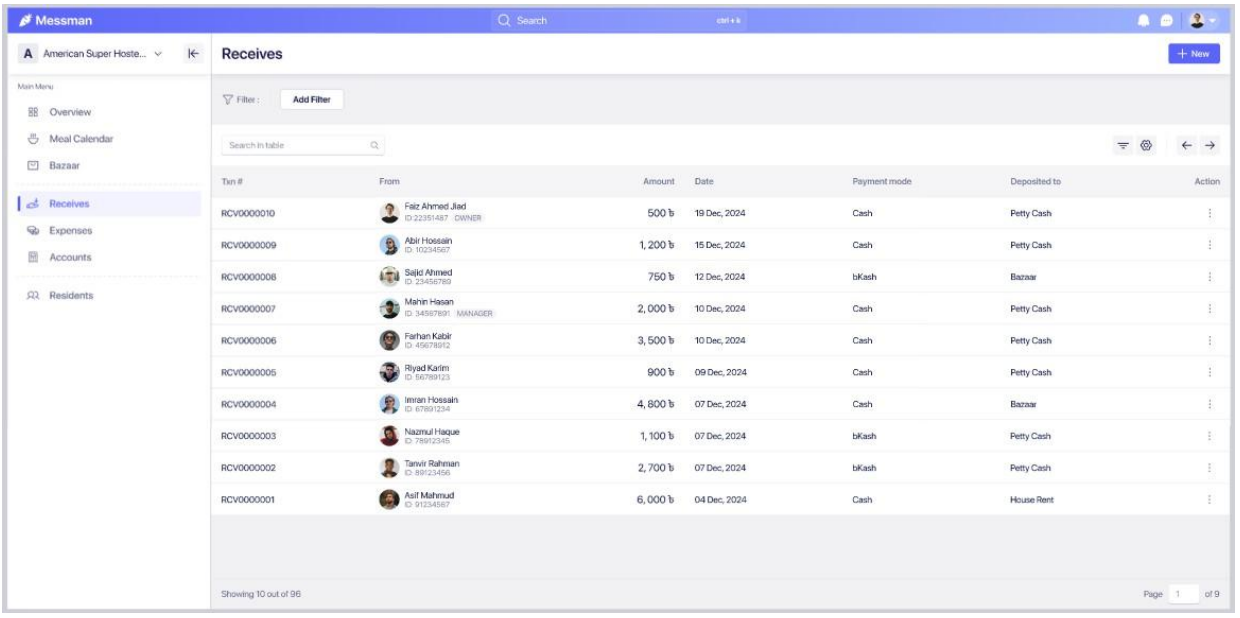


Figure 6.5.2: Receives Money (adapted from figma.com, 2025)

6.5.3 Add New Receives Entry

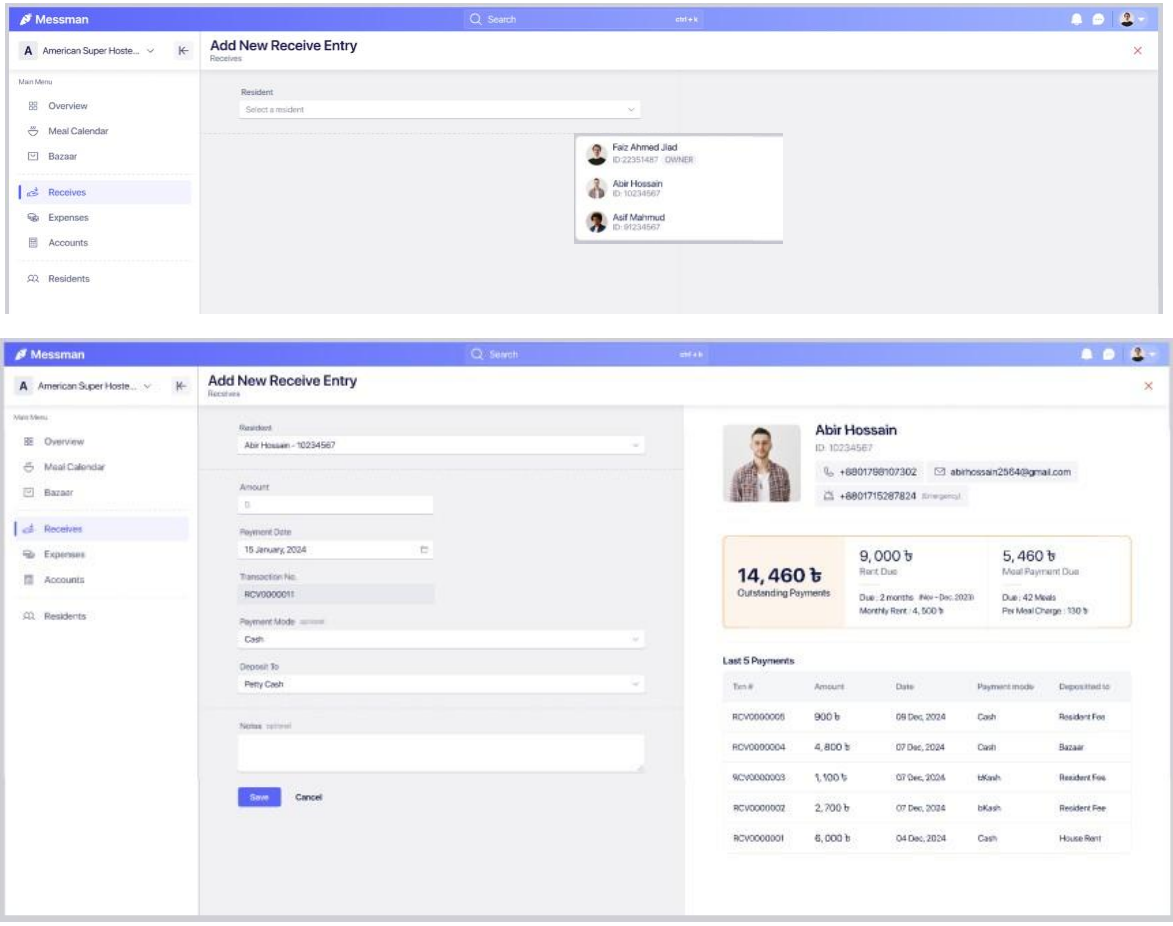


Figure 6.5.3: New Receives Entry (adapted from figma.com, 2025)

6.6 Expenses

What is marketed daily is mentioned here along with dates and costs. The manager takes the information from the members and writes it down.

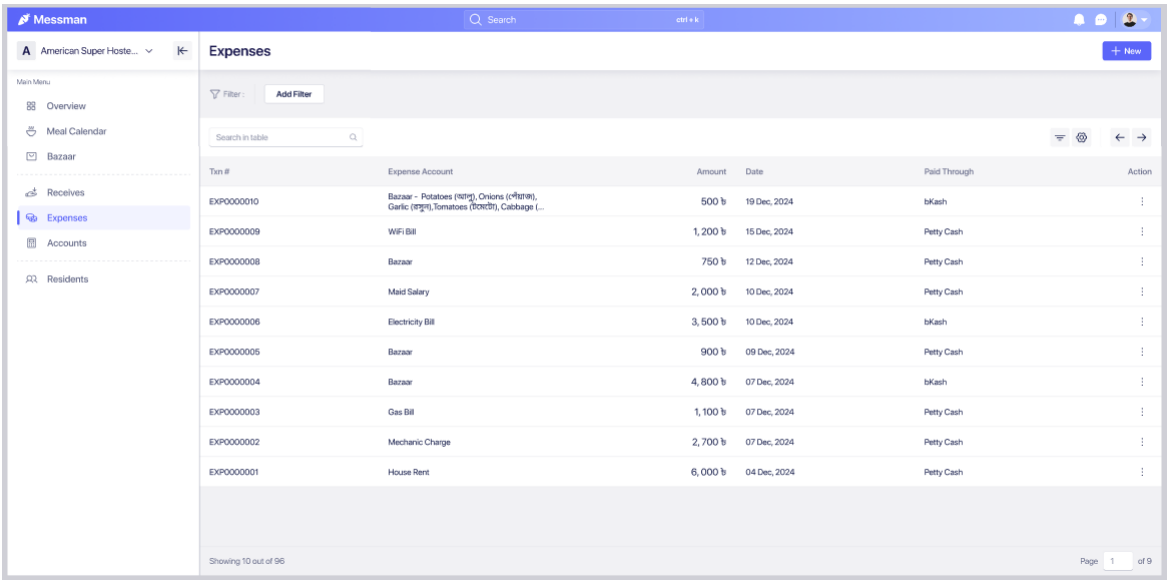
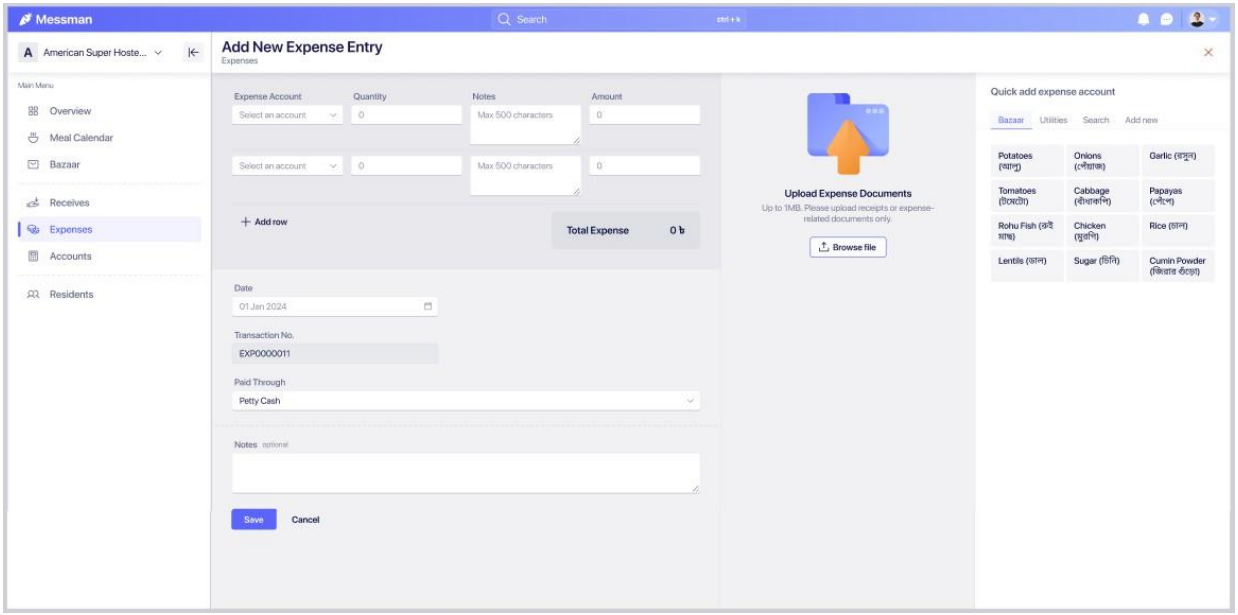


Figure 6.6: Expenses (adapted from figma.com, 2025)

6.6.1 Add New Expense with Chart



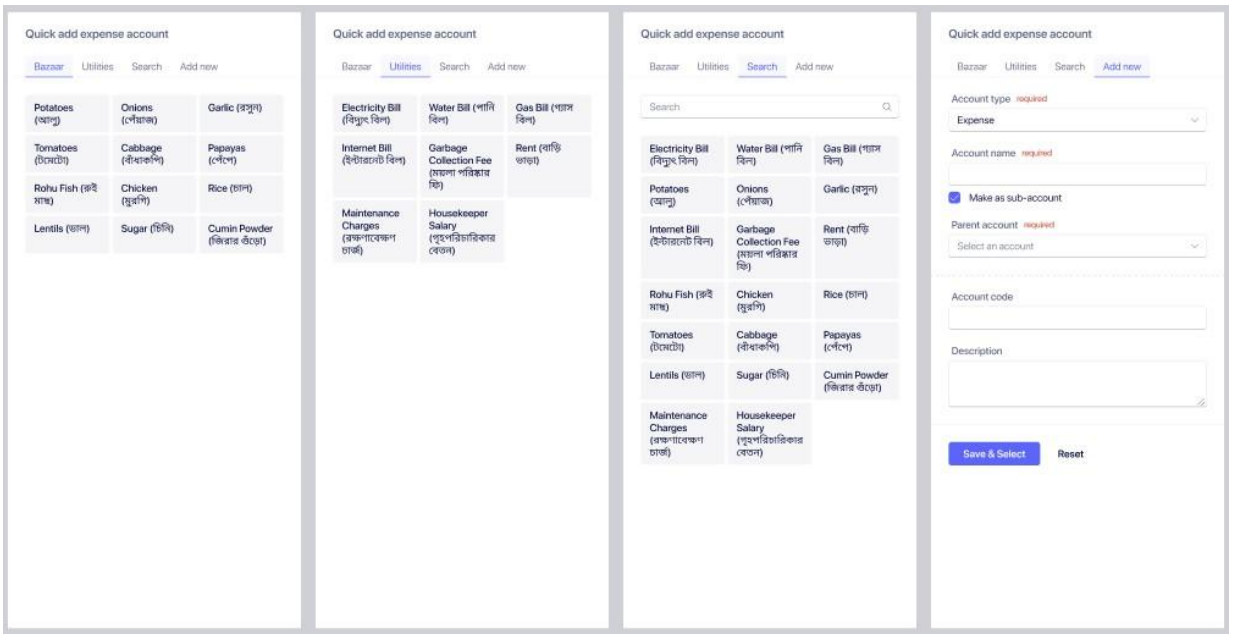


Figure 6.6.1: Add New Expense Chart (adapted from figma.com, 2025)

6.6.2 Add New Expense Entry and View Dashboard

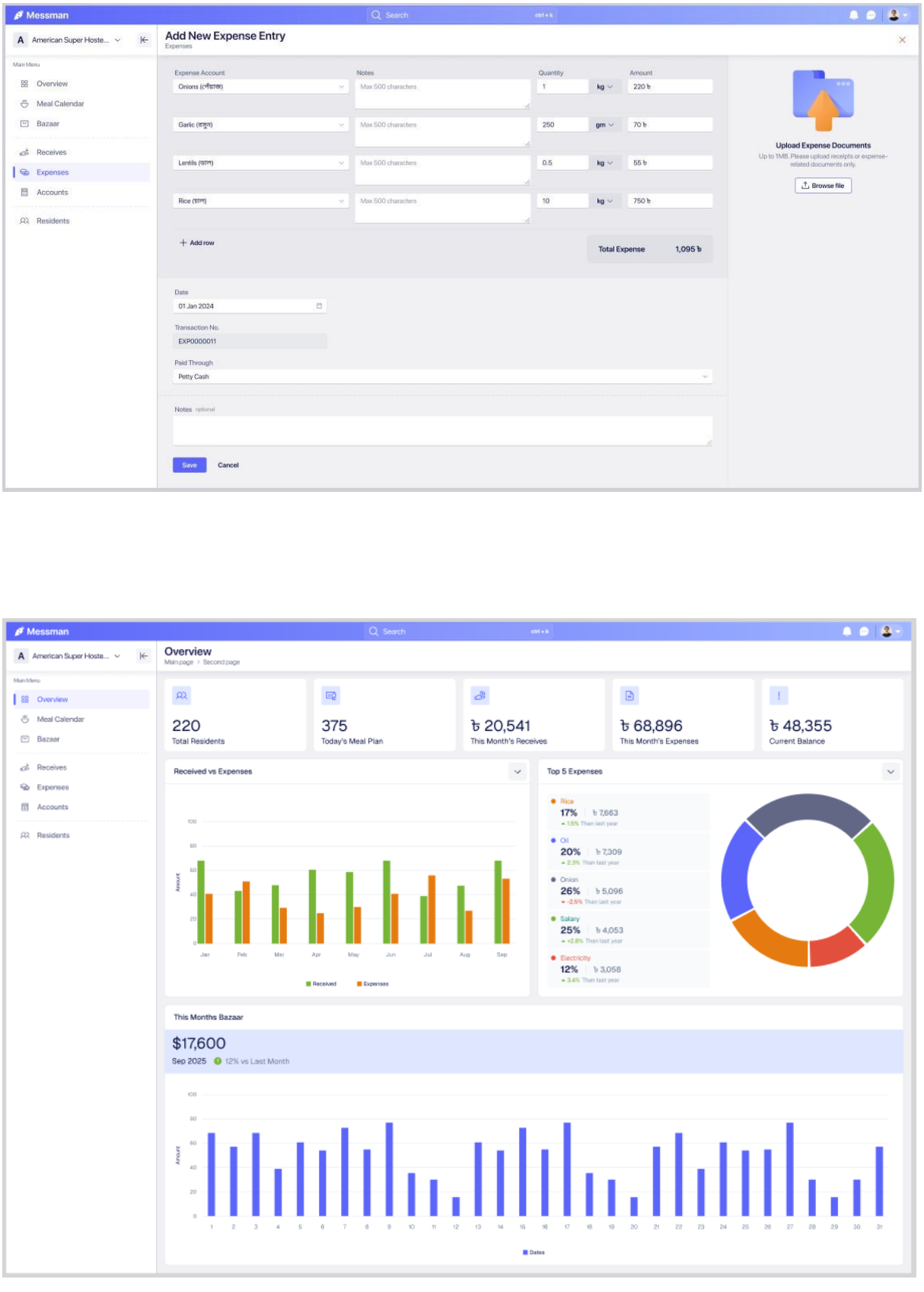


Figure 6.6.2: Add New Expense Entry and View Dashboard
(adapted from figma.com, 2025)

6.6.3 Add New Account

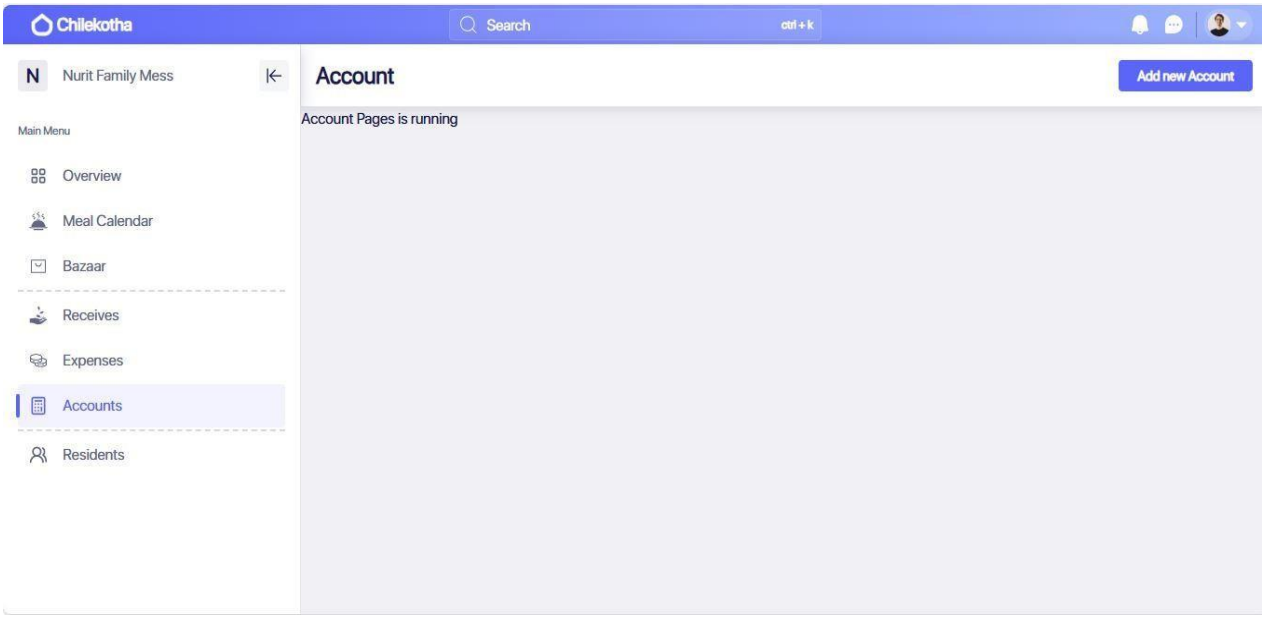


Figure 6.6.3: Add New Account (adapted from figma.com, 2025)

6.7 Residents

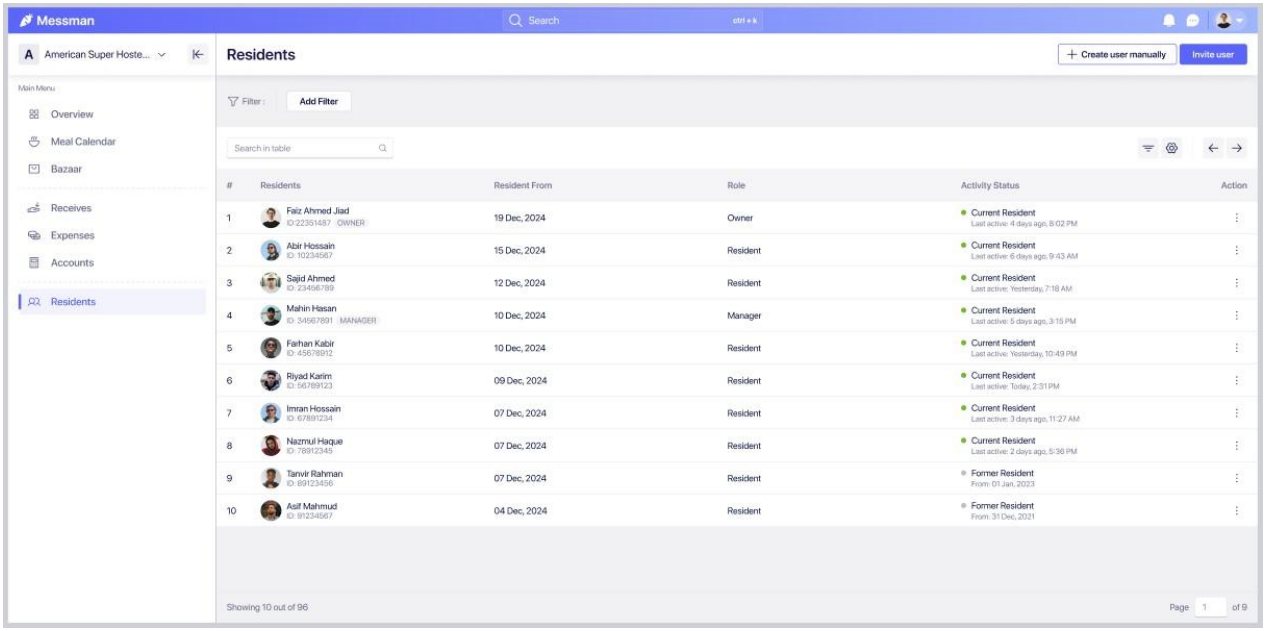


Figure 6.7: Residents (adapted from figma.com, 2025)

6.8 Conclusion

Living in a student mess or hostel brings unique opportunities for learning and growth, but it also creates many challenges in daily management. Issues such as food arrangements, expense sharing, duty distribution, and communication gaps often lead to conflicts and inefficiency. Through this study, the proposed Mess Management System – Chilekotha provides a structured framework to address these difficulties in a fair, transparent, and organized way.

The system emphasizes core principles such as cooperation, accountability, and sustainability. By introducing simple but effective methods for meal planning, expense tracking, task allocation, and record-keeping, Chilekotha transforms the student mess experience into one that is more harmonious and efficient.

This work also highlights how technology and innovative thinking can reshape collective living, making it easier for students to focus on their studies and personal growth rather than daily disputes. Although challenges remain in terms of adoption and improvement, Chilekotha represents a meaningful step toward smarter and more sustainable mass management.

In conclusion, A Digital Approach to Mess Management System – Chilekotha is not just a tool, but a vision for better student life. It promotes fairness, reduces conflicts, and builds a culture of cooperation, ultimately creating a healthier and more productive environment for all students living together.

CSE Reference Format

1. CHS Rao, "The Development of a Web-Based Integrated Hostel Management System for Andhra University," IJSAT, 2025. [Online]. Available: https://www.ijsat.org/papers/2025/2/3938.pdf. [Accessed: July. 16, 2025].

2. "Hostel Management System," IRJMETS, Vol. 07, Issue 03, Mar. 2025. [Online]. Available: Hostel Management System (https://tinyurl.com/3rzuh33j).

3. "Design and Implementation of a Web-based Student Hostel Management Application System for MUCE," Misheck Mutuzana, 2020, Bachelor Thesis, University of Zambia. [Online]. Available: Student Hostel Management (https://tinyurl.com/3venb24m). [Accessed: Aug. 11, 2025]

4. "An Intelligent Student Hostel Allocation System Based on Web Applications," A. Azeta, S. Misra, M. Odusami, U. Onyepunuka, Lecture Notes in Electrical Engineering, ICRIC 2020, pp. 779-791, Jan. 2021. [Online]. Available via ResearchGate. [Accessed: Aug. 26, 2025]

5. Rahman, M., & Sultana, T. (2019). Digital Solutions for Hostel and Mess Management: A Case Study in Dhaka. Presented at International Conference on Computer and Information Technology (ICCIT), Dhaka, Bangladesh.

6. Bangladesh University of Engineering and Technology (BUET). (2018). Challenges of Hostel and Mess Management for Students. Technical Report, Department of CSE.

7. Vishal Sharma (2025). A Digital Approach to Mess Management System, The Role of Chilekotha. Undergraduate Project Report, Sonargaon University, Department of Computer Science and Engineering.

8. Rahman, M., & Sultana, T. (2019). Digital Solutions for Hostel and Mess Management: A Case Study in Dhaka. Presented at International Conference on Computer and Information Technology (ICCIT), Dhaka, Bangladesh.