Design Document

for

Mess Automation System

Version 0.1

Prepared by

Group:7 Group Name:The Software Savants

Ashutosh Agrawal	210219	ashutosha21@iitk.ac.in
Sawan H N	210952	sawanhn21@iitk.ac.in
Mansi Singh	210590	mansisingh21@iitk.ac.in
Prapti Dagli	210748	praptid21@iitk.ac.in
Sanjay Kumar	210080	askumar21@iitk.ac.in
Archisha Singh	210181	archishas21@iitk.ac.in
Mokshagna Prattipati	210761	mokshagna21@iitk.ac.in
Charugundla Chethan	210279	chethan21@iitk.ac.in
Chandaka Hemanth kumar	210277	chandaka21@iitk.ac.in

Course: CS253

Mentor TA: *Archit Gupta (garchit@cse.iitk.ac.in)*

Date: *10 Feb 2023*

Contents II

Revisions II

1 Context Design 1

1.1 Context model 1

1.2 human Interface Design 1-13

2 Architecture design 14

3 object-oriented design 15

3.1 Use case diagram 15

3.2 Class diagram 16

3.3 sequence diagram 17-19

3.4 State diagram 20

4 Project Plan 21-23

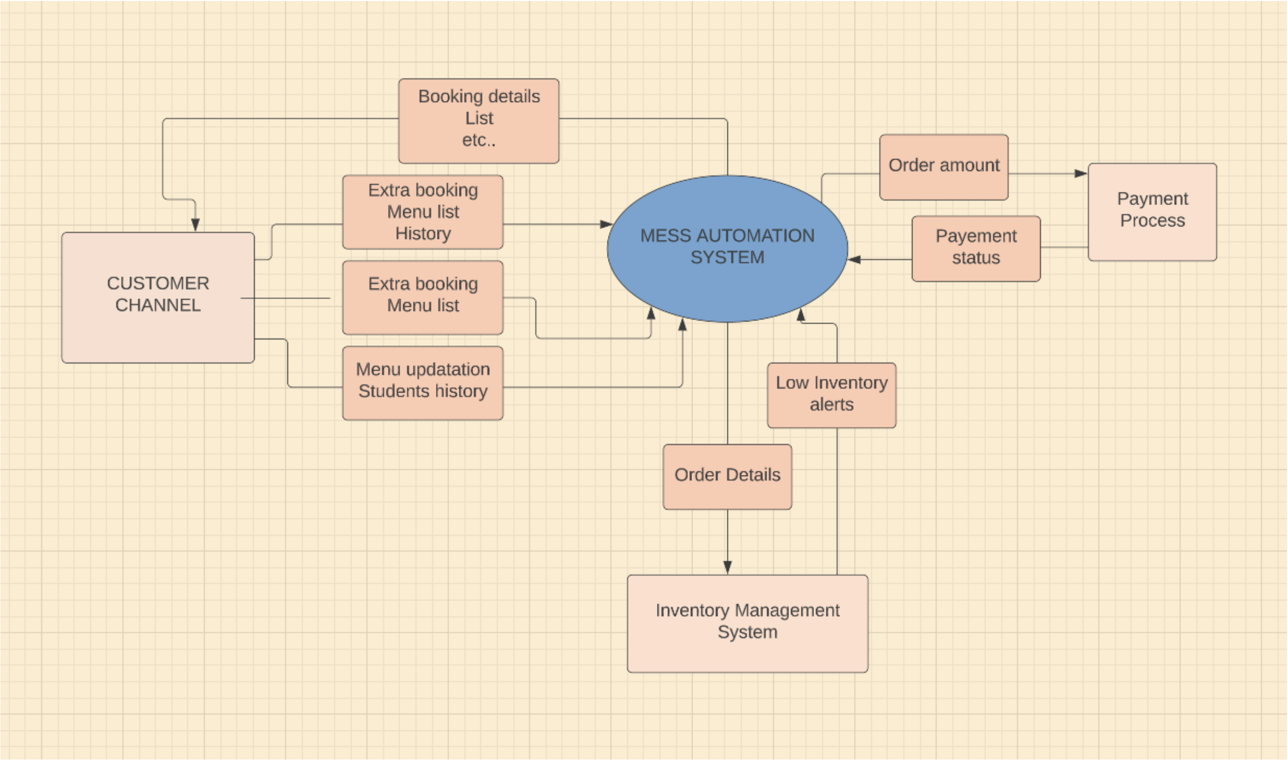
Appendix A - Group Log 24

Revisions

Version	Primary Author(s)	Description of Version	Date Completed
0.1	Ashutosh Agrawal Sawan H N Mansi Singh Prapti Dagli Sanjay Kumar Archisha Singh Mokshagna Prattipati Charugundla Chethan Chandaka Hemanth Kumar	First Draft	10/02/2023

1 Context Design

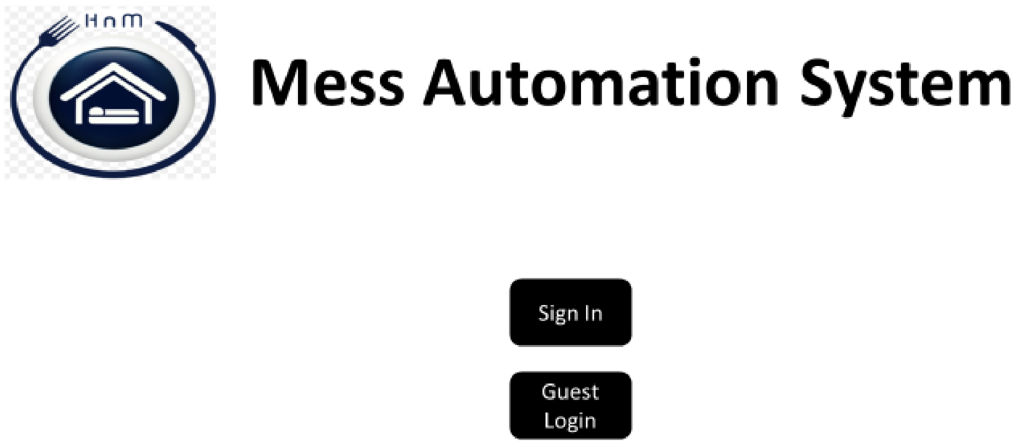
1.1 Context Model



1.2 Human Interface Design

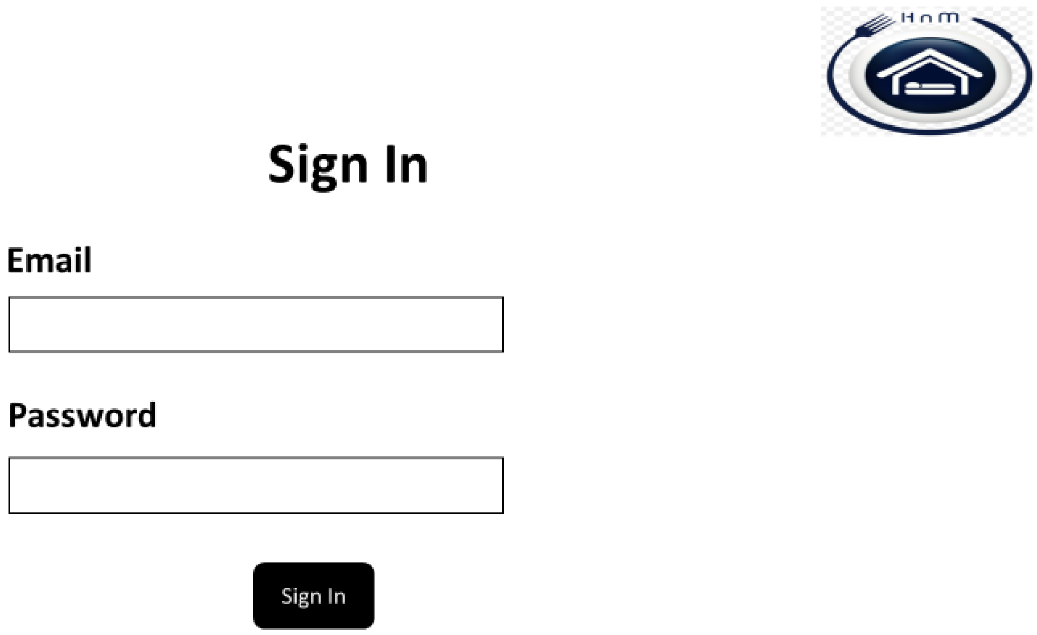
There are eighteen interfaces in our web-app, each having its own utility. Each interface is meant to be used for either hall residents, mess manager or guests and facilitates a smooth user experience with the clients. The interface design of the mess automation system with all eighteen interfaces is given below.

Home Page



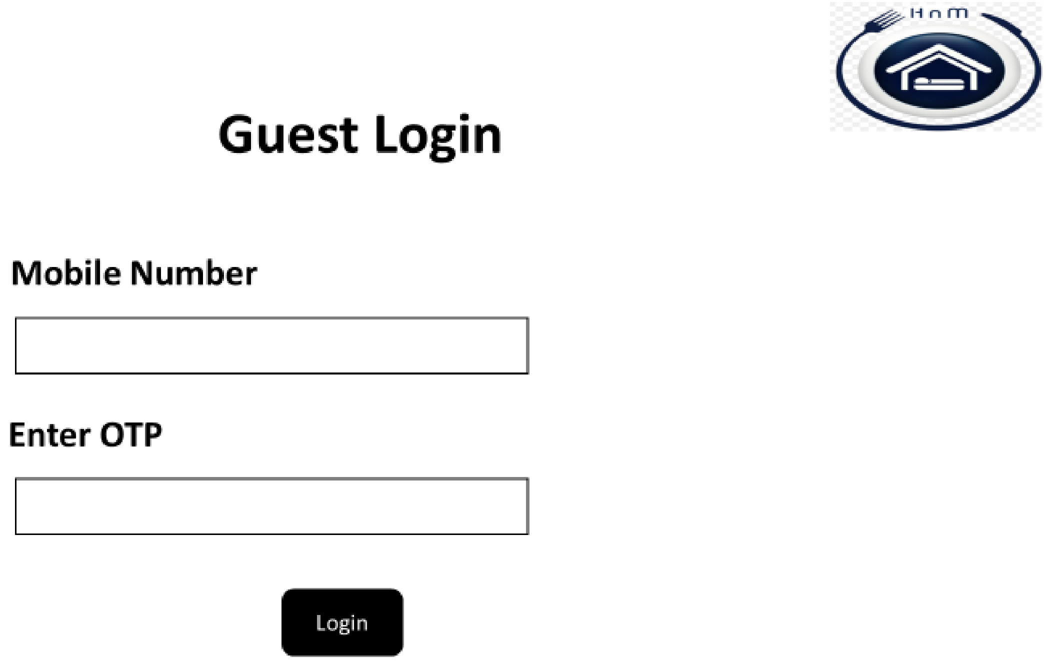
Mess Automation System

Sign in(Manager and Resident's Dashboard)
sign_in(email-id,pswd)

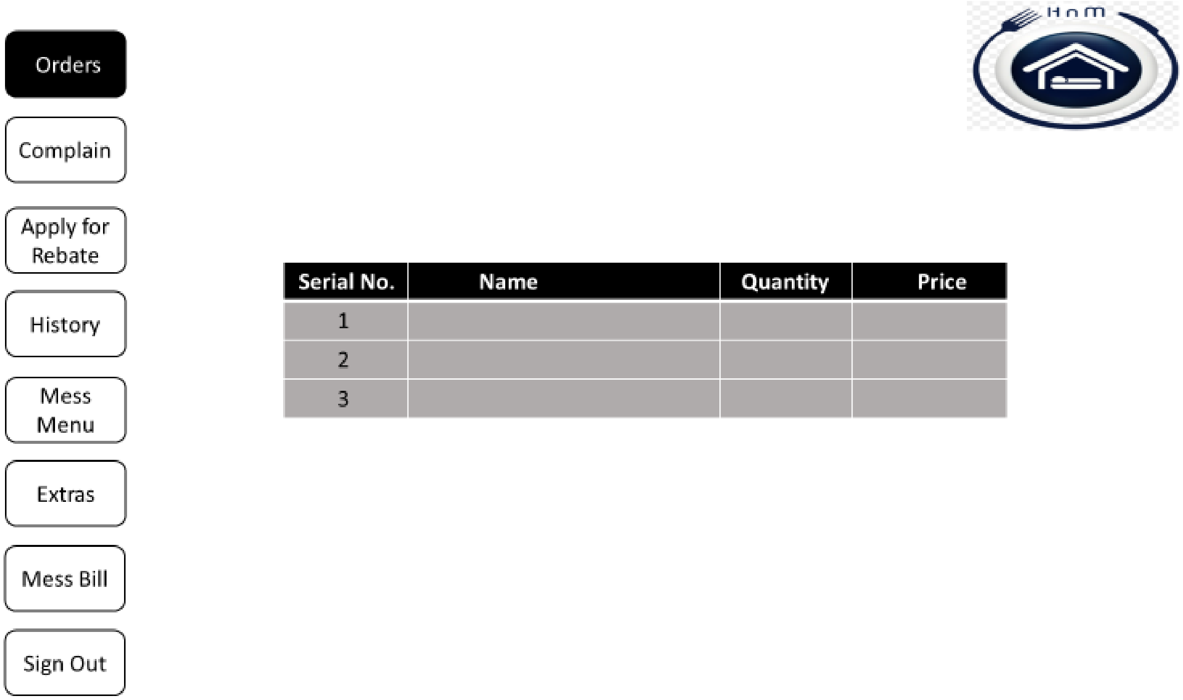


Sign In

Guest Login
Guest_login(mobile_number, OTP)

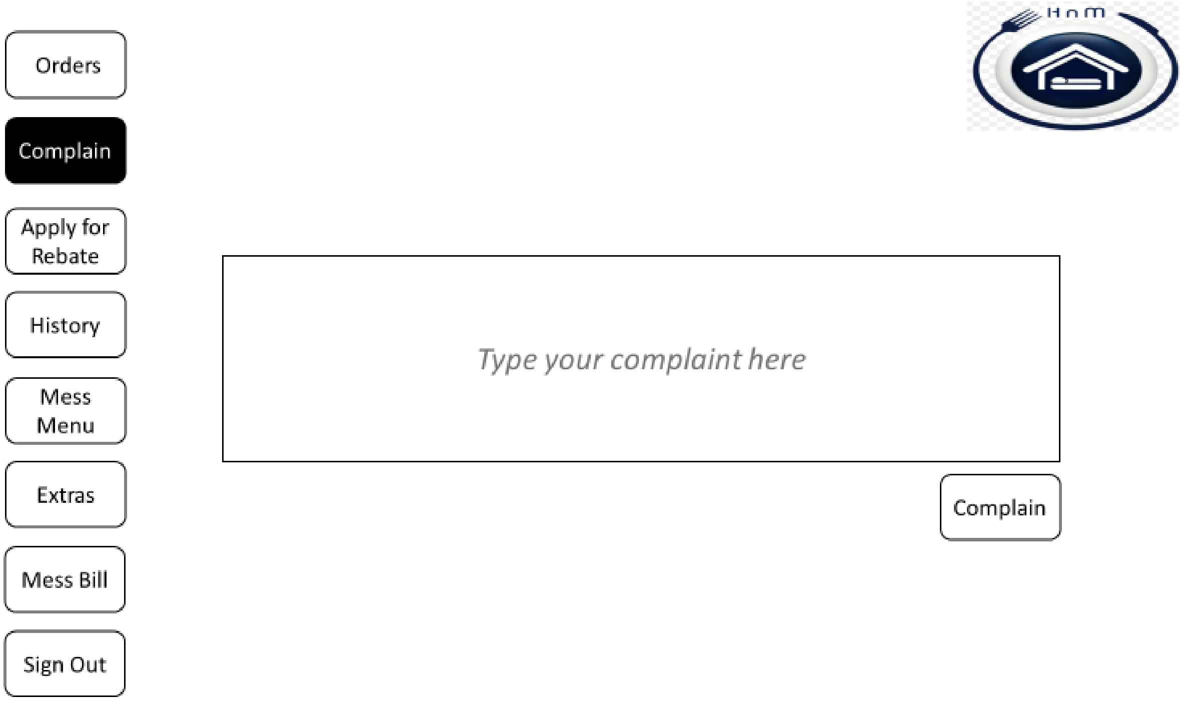


Orders(Resident's Dashboard)
Book_extras(extra_item_name)

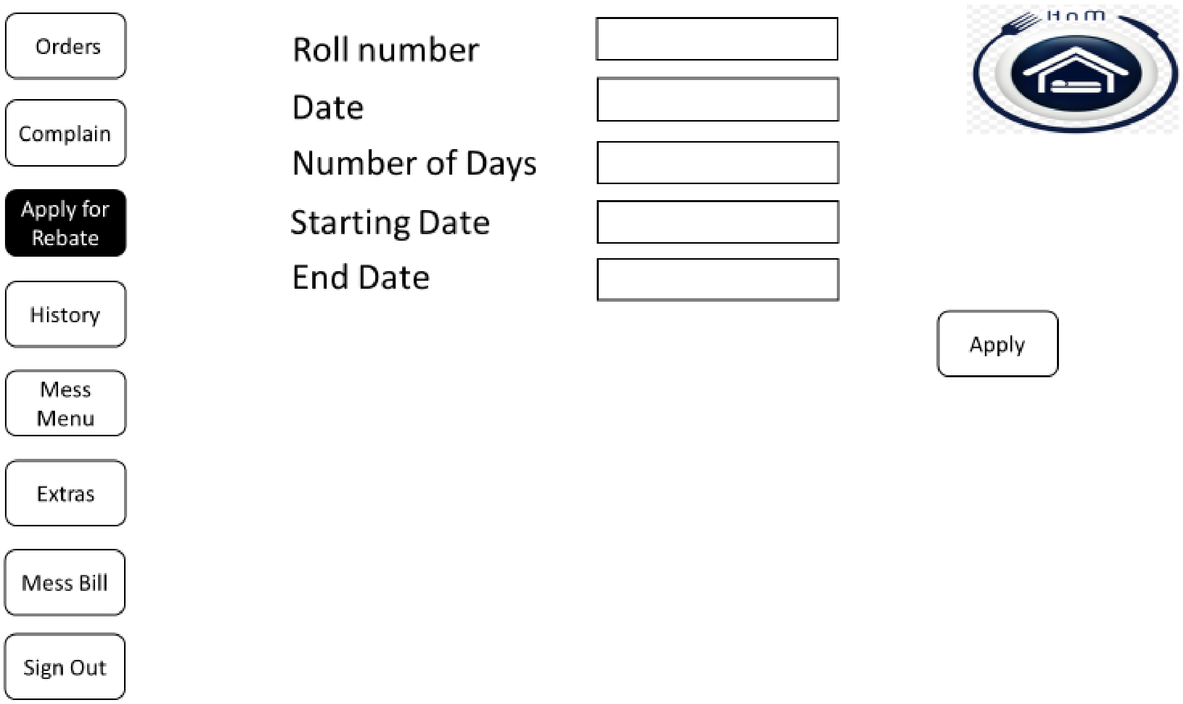


Serial No.	Name	Quantity	Price
1			
2			
3			

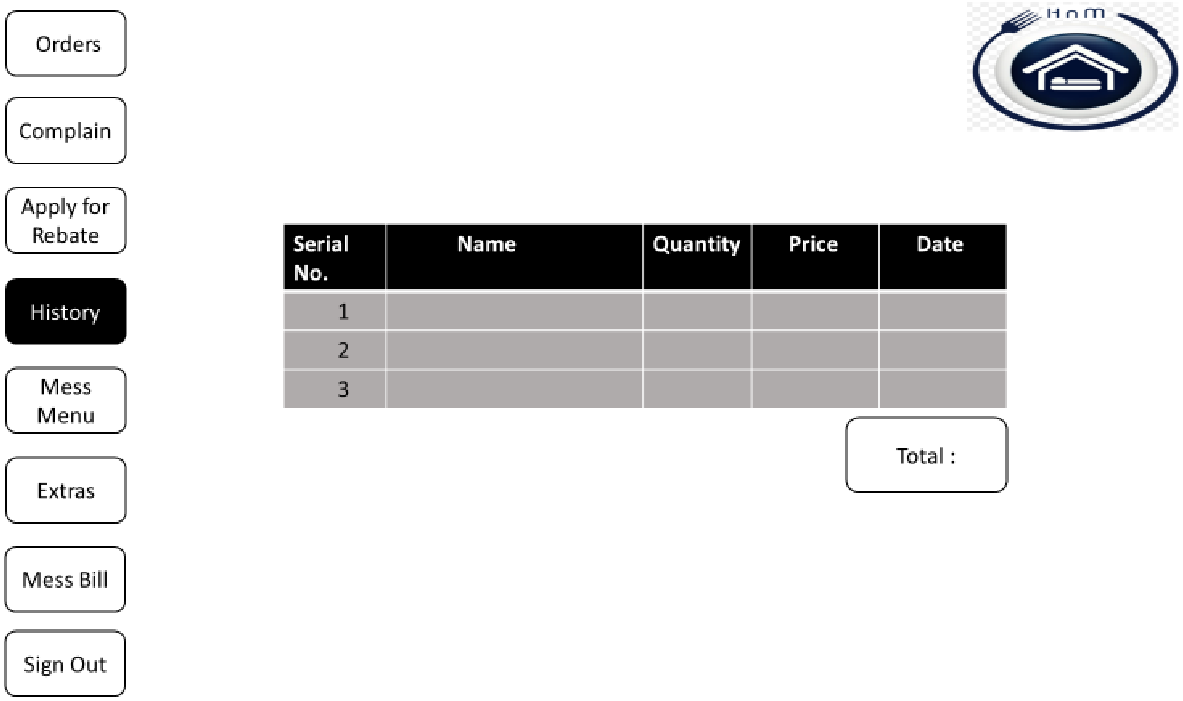
Complain(Resident's Dashboard)
Complain(complaint)



Apply for Rebate(Resident's Dashboard)
Mess_Rebate(Roll_no, Date, Start_Date, End_Date, No_of_days)



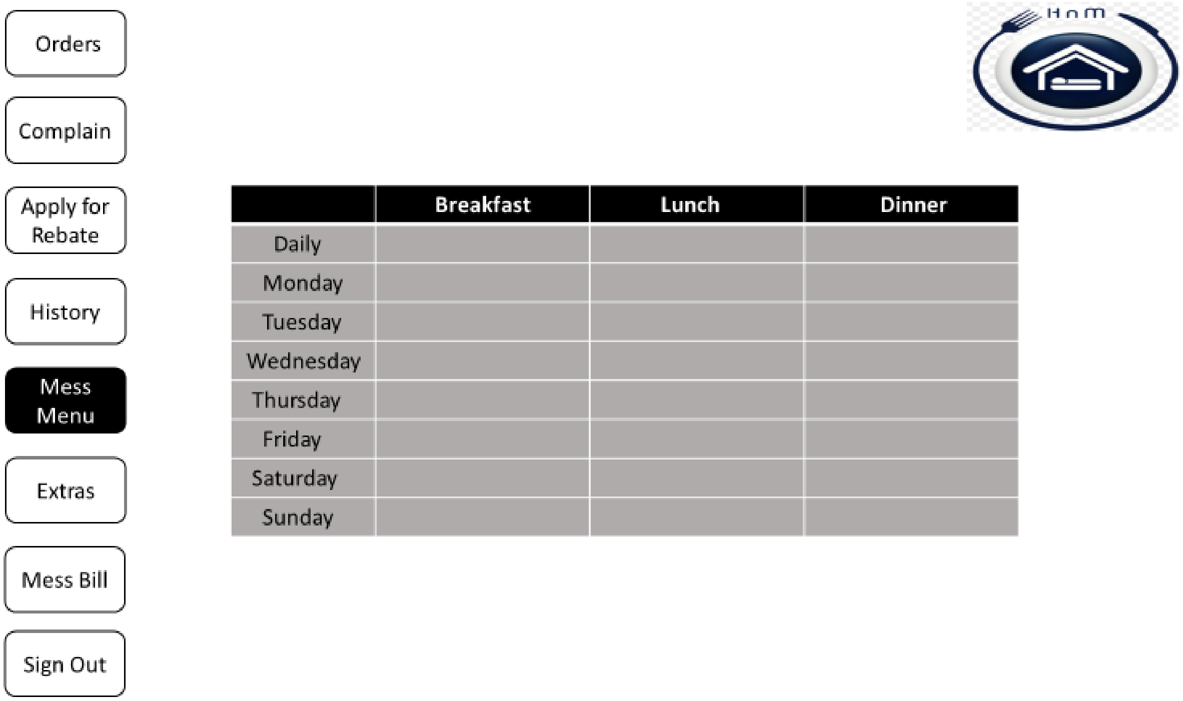
History(Resident's Dashboard)
History(User_id)



Serial No.	Name	Quantity	Price	Date
1				
2				
3				

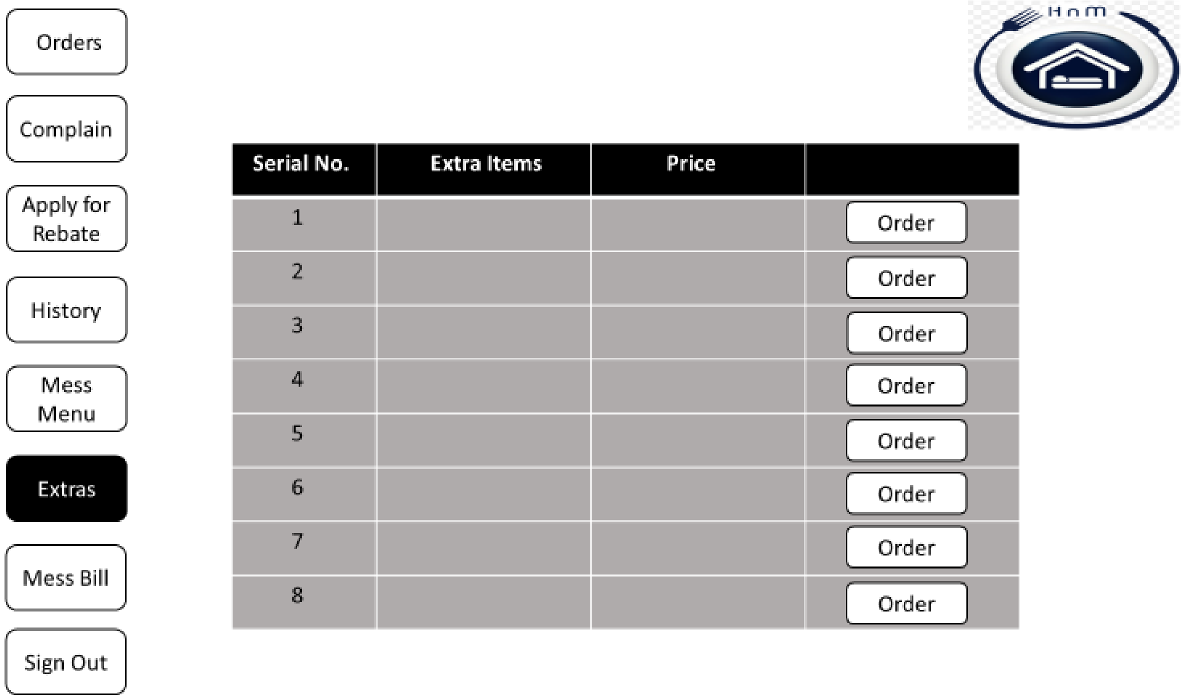
Mess Menu(Resident's and Guest's Dashboard)

get_menu(User_id)



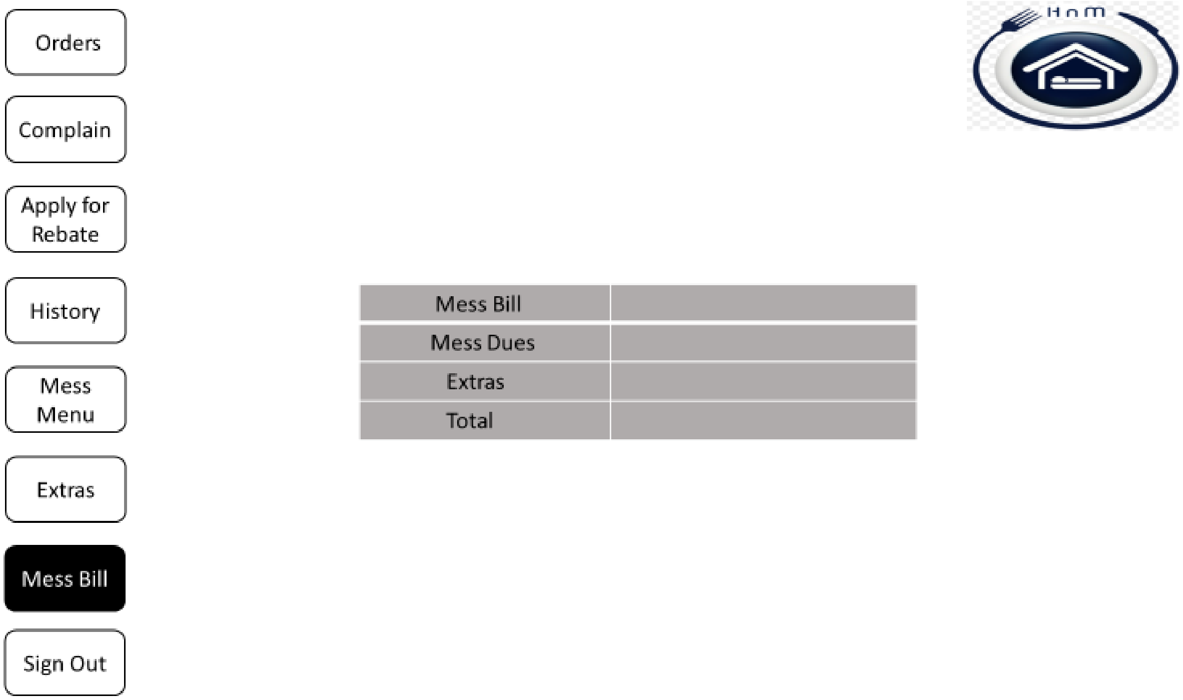
	Breakfast	Lunch	Dinner
Daily			
Monday			
Tuesday			
Wednesday			
Thursday			
Friday			
Saturday			
Sunday			

Extras(Resident's and Guest's Dashboard)
Book_extras(extra_item_name)



Serial No.	Extra Items	Price	
1			Order
2			Order
3			Order
4			Order
5			Order
6			Order
7			Order
8			Order

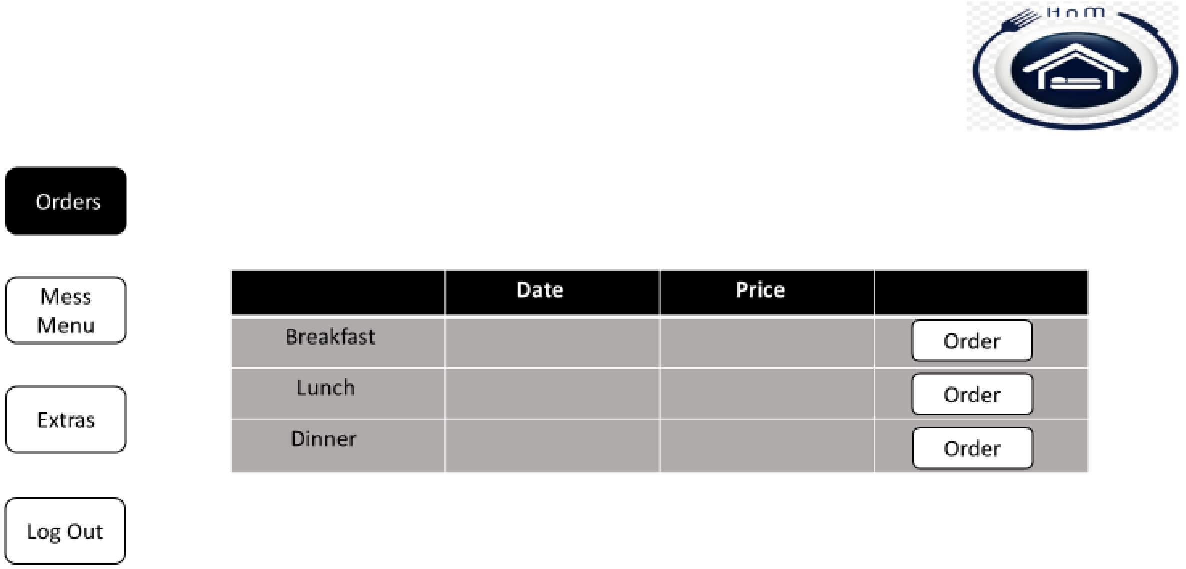
Mess Bill(Resident's Dashboard)
See_mess_menu(User_id)



Mess Bill	
Mess Dues	
Extras	
Total	

Orders(Guest's Dashboard)

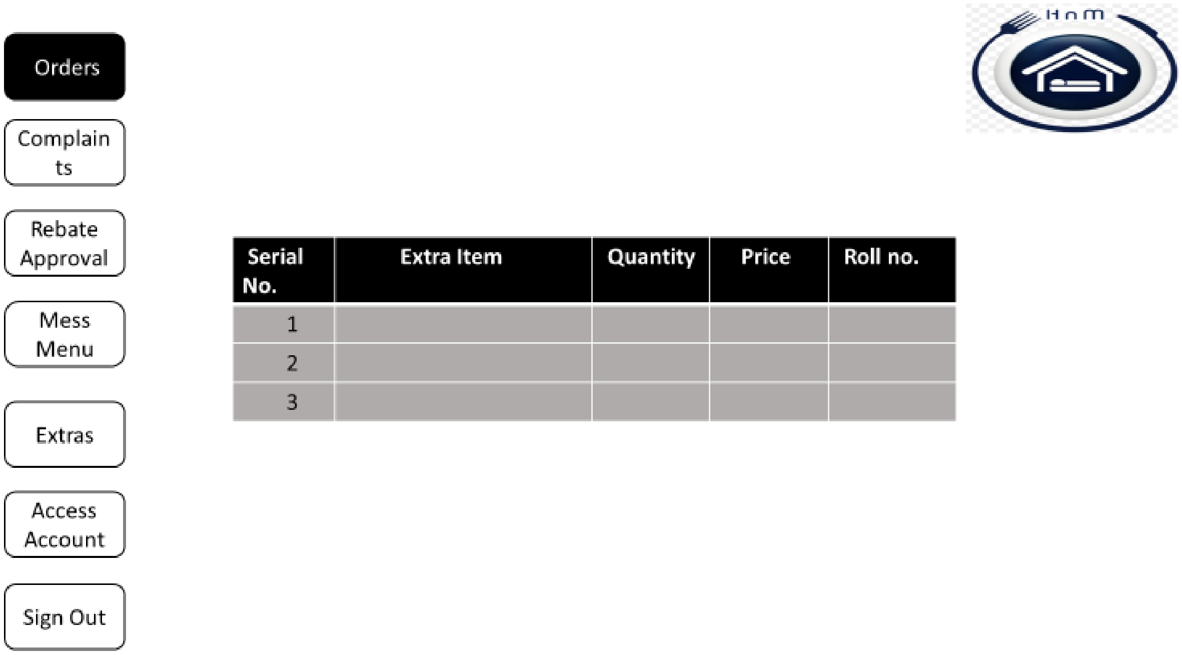
Book(meal_name)



	Date	Price	
Breakfast			Order
Lunch			Order
Dinner			Order

Orders(Manager's Dashboard)

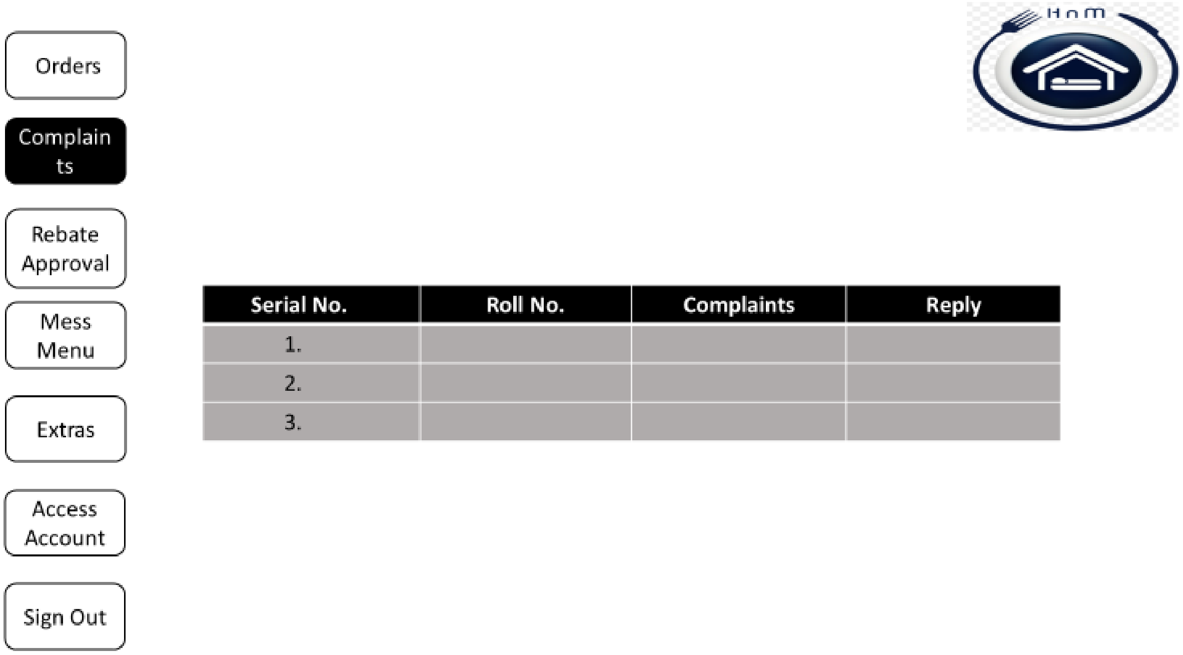
See_order(manager_id)



Serial No.	Extra Item	Quantity	Price	Roll no.
1				
2				
3				

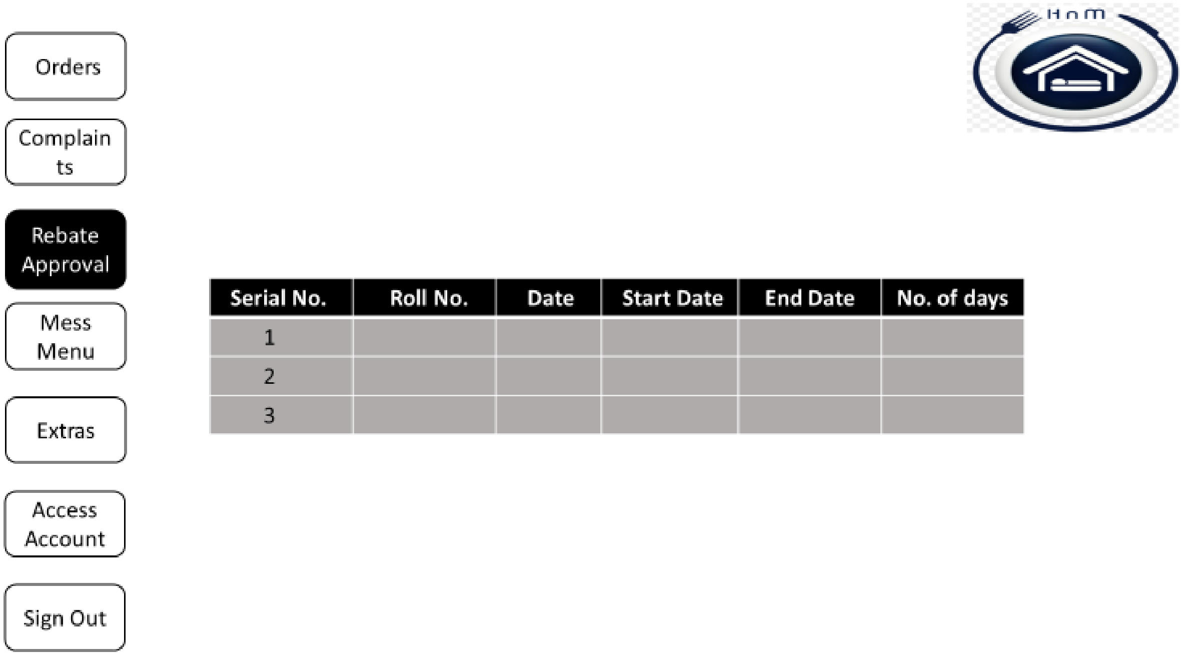
Complaints(Manager's Dashboard)

Reply_complaints(Reply)



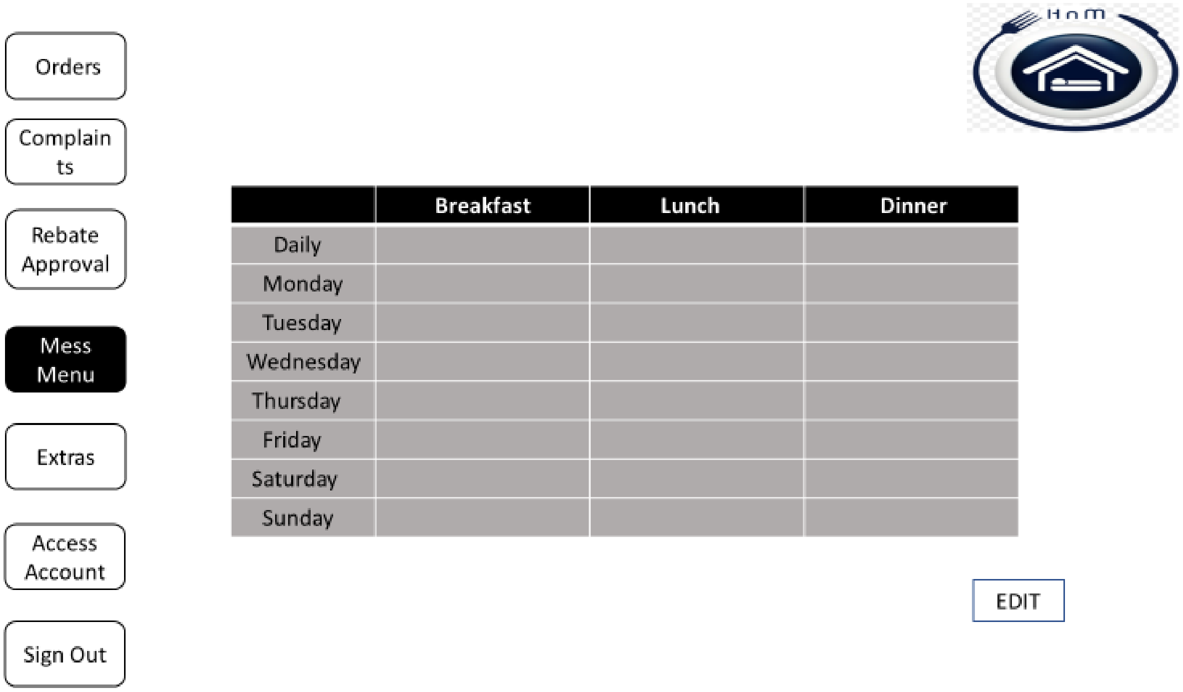
Serial No.	Roll No.	Complaints	Reply
1.			
2.			
3.			

Rebate Approval(Manager's Dashboard)
Upload_rebate_form (manager_id)



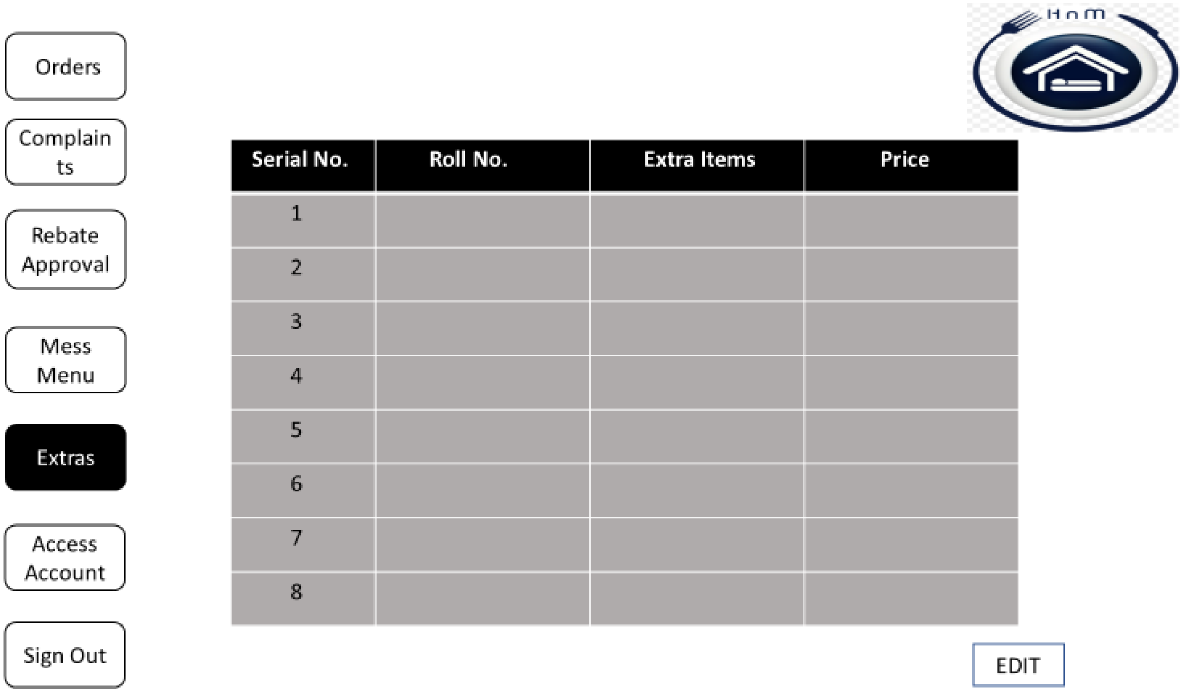
Serial No.	Roll No.	Date	Start Date	End Date	No. of days
1					
2					
3					

Mess_Menu(Manager's Dashboard)
Upload_mess_menu (mess_menu)



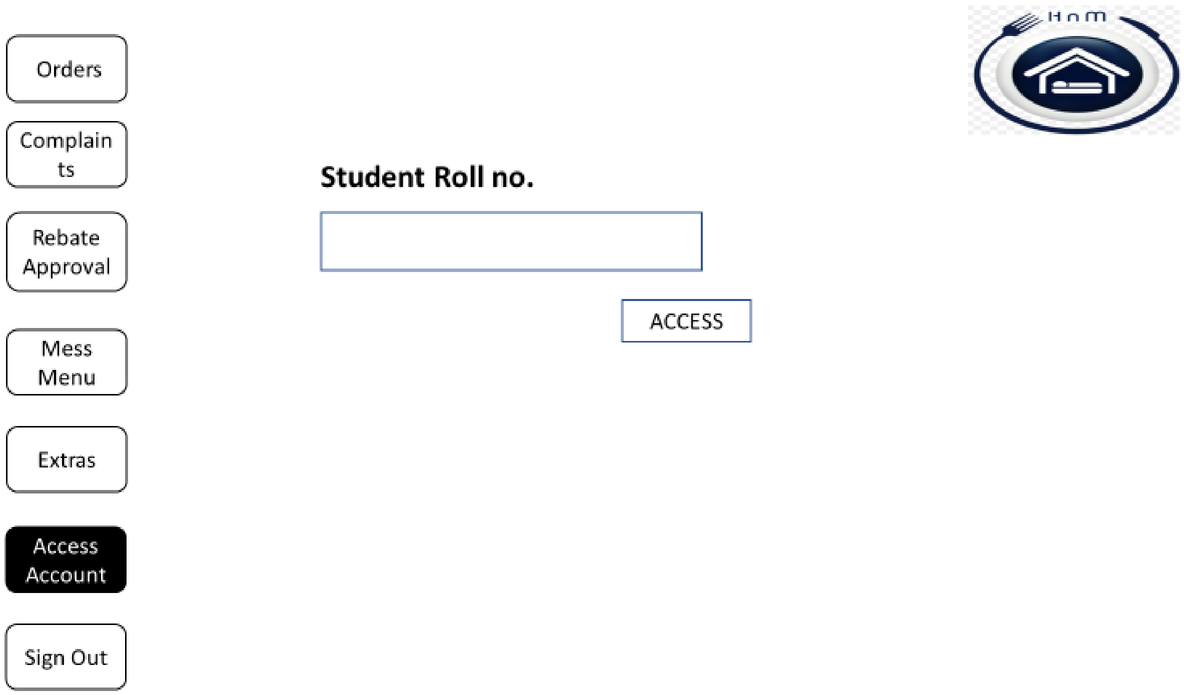
	Breakfast	Lunch	Dinner
Daily			
Monday			
Tuesday			
Wednesday			
Thursday			
Friday			
Saturday			
Sunday			

Extras(Manager’s Dashboard)
Upload_extras(extra_name,price)

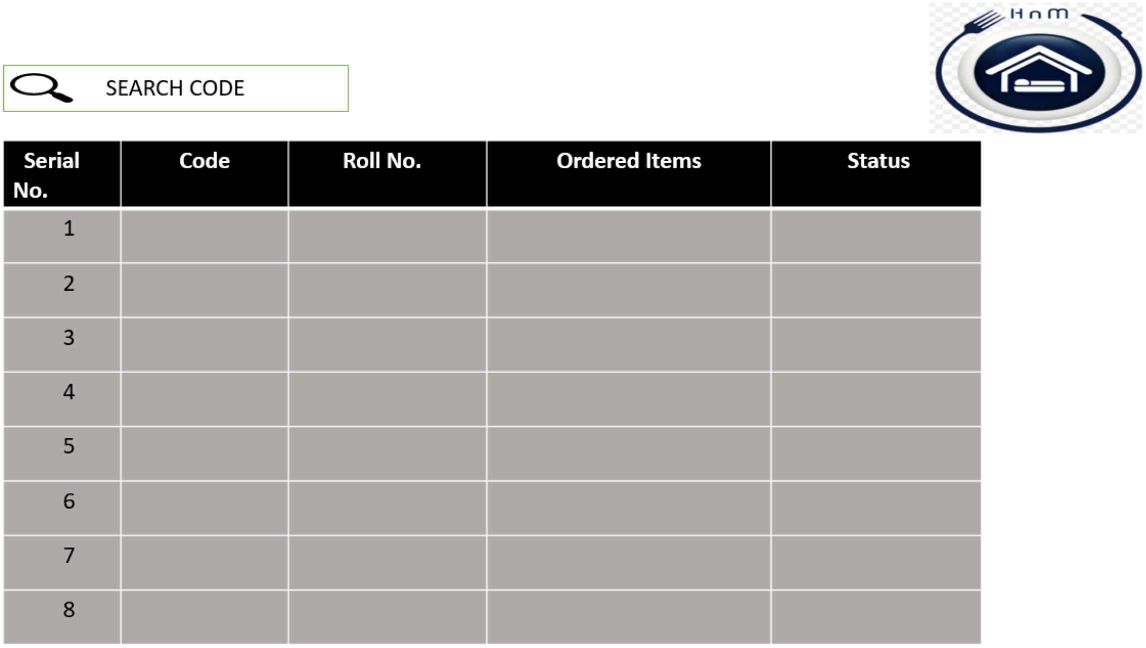


Serial No.	Roll No.	Extra Items	Price
1			
2			
3			
4			
5			
6			
7			
8			

Access Account(Manager's Dashboard)
Access(Roll_No)



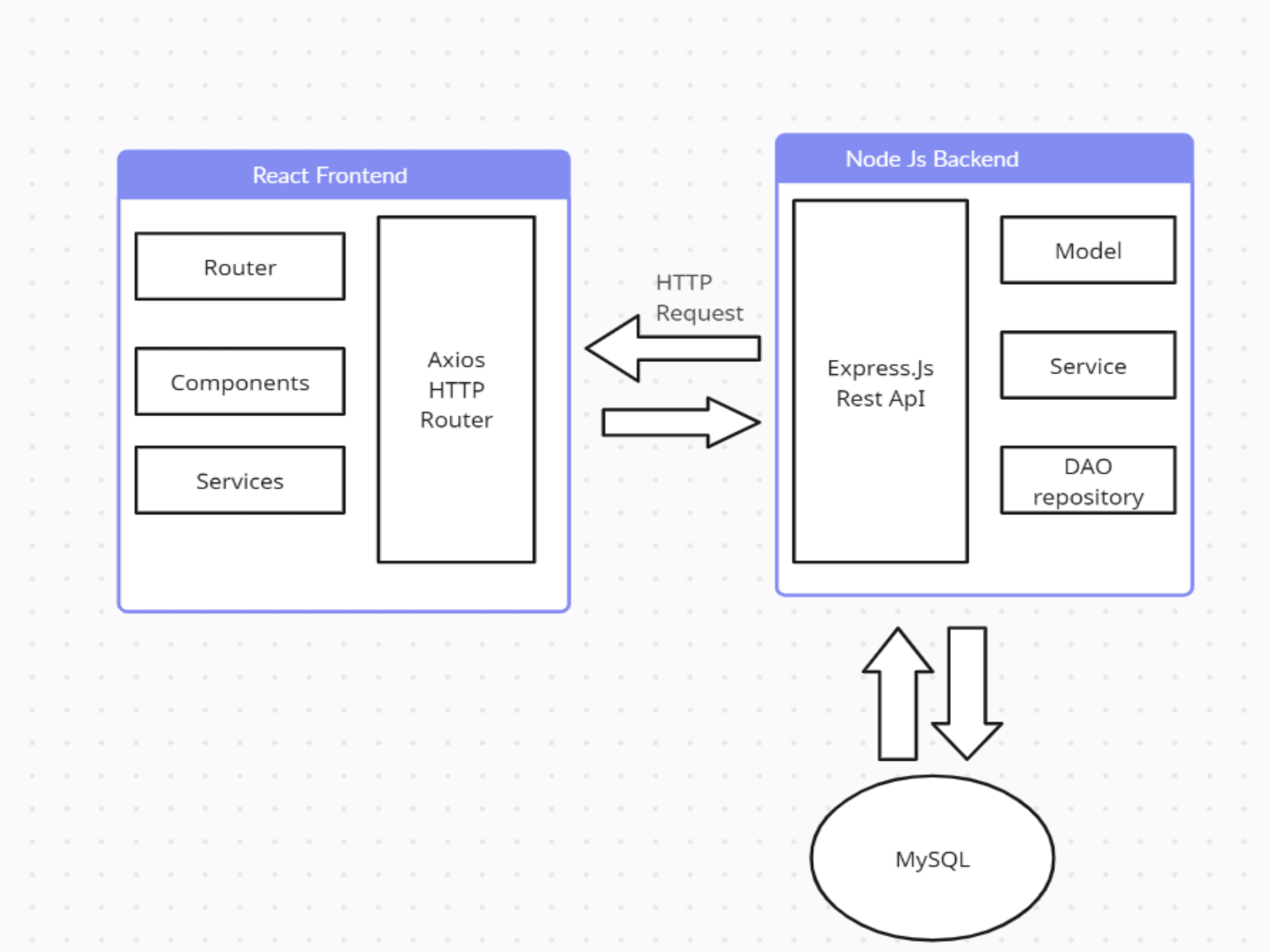
Verify Orders (worker's dashboard)
Edit status



Serial No.	Code	Roll No.	Ordered Items	Status
1				
2				
3				
4				
5				
6				
7				
8				

2 Architecture Design

Mess Automation System is a web-based application implementing a Model-View-Controller model.



Why are we using this: We are using this model since we would be interacting with the data in multiple ways, like checking and receiving requests, managing orders,complaints and rebates , and storing transactions. Also, in the near future, one might think of adding some functionalities to it, which makes the Model-View-Controller model the best fit.

Advantages: One great advantage of this model is that the data can be represented in various manners. Which will prove helpful to us while sorting students based on dues, or from the same academic year or in some other fashion. Also, it allows the data to change independently, it would be helpful to add or remove students from the database.

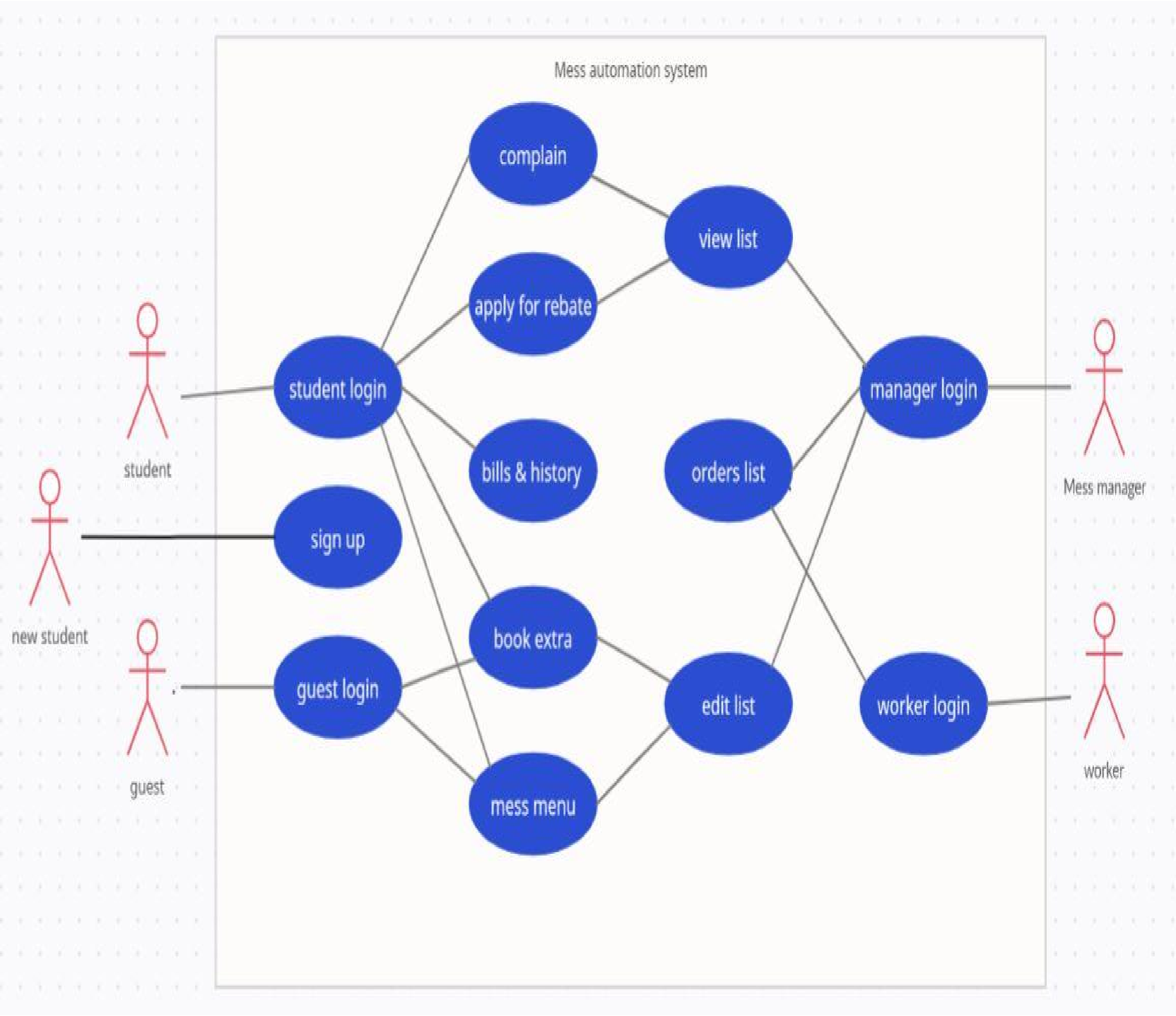
Disadvantages: One disadvantage is that we need to write some additional code for full functionality, even if we know that our implementation is simple in the starting stages.

3 Object Oriented Design

3.1 Use Case Diagrams

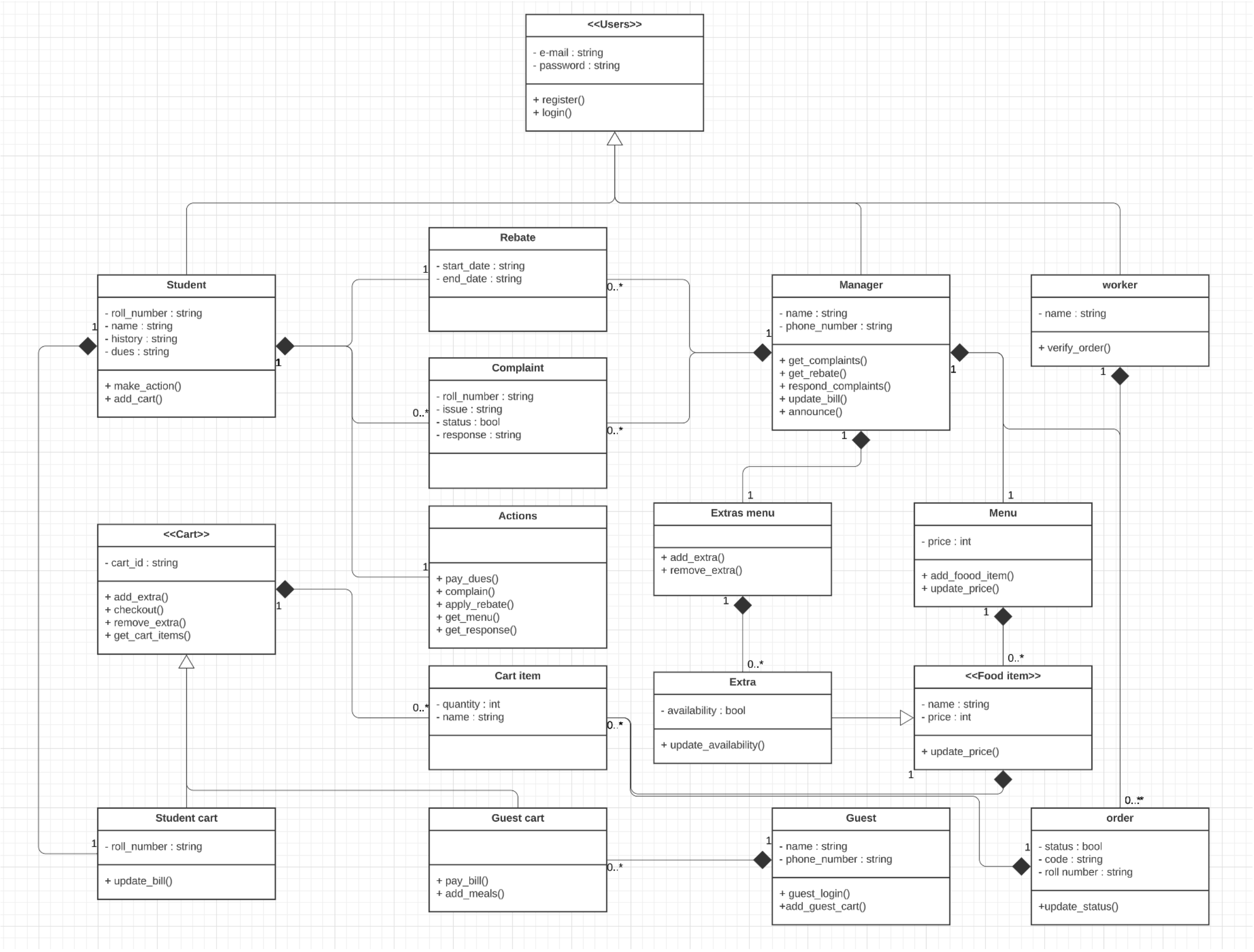
We have given a consolidated diagram showing different use cases provided:

● Student registration and account creation.
● Bills and history of transaction linked to a particular student.
● Applying for a rebate.
● Register as a guest using email and OTP verification.
● Checking out and booking extras.
● Manager overseeing the transaction and addressing complaints and rebates.

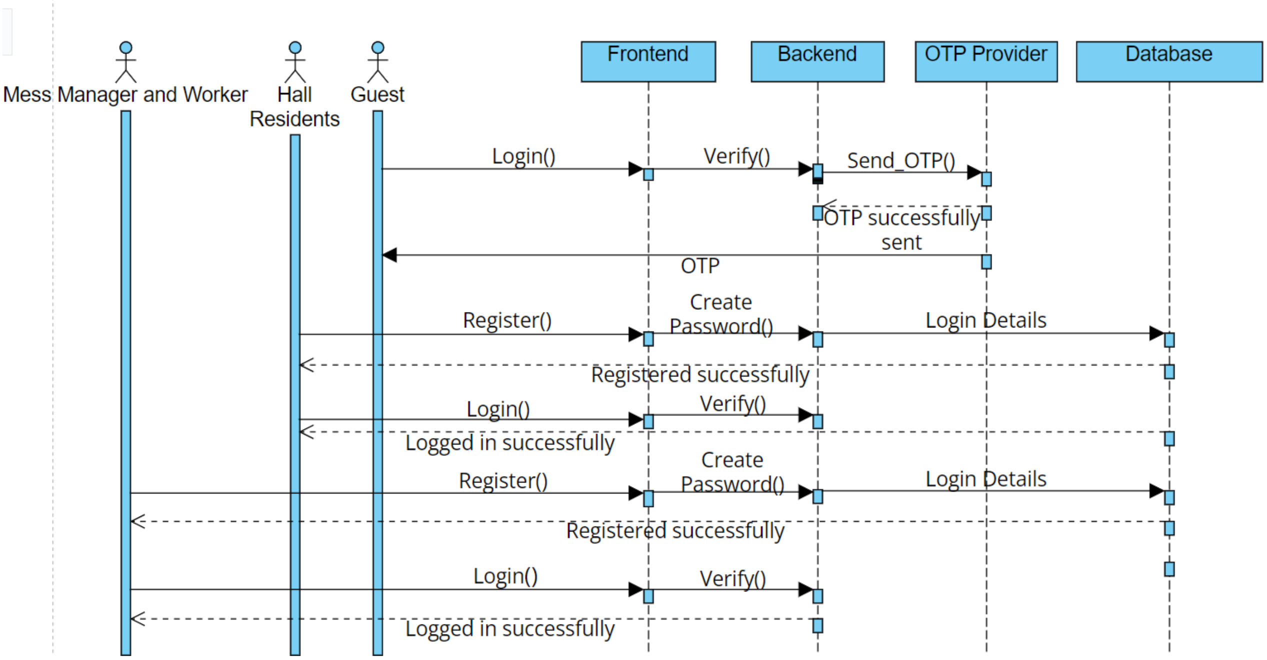


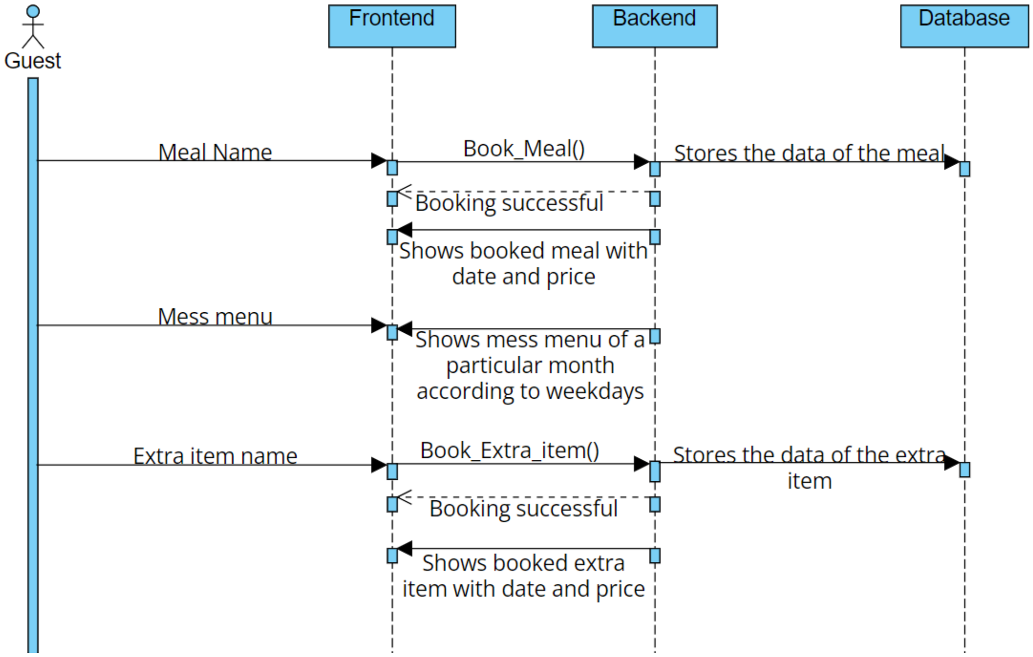
3.2 Class Diagrams

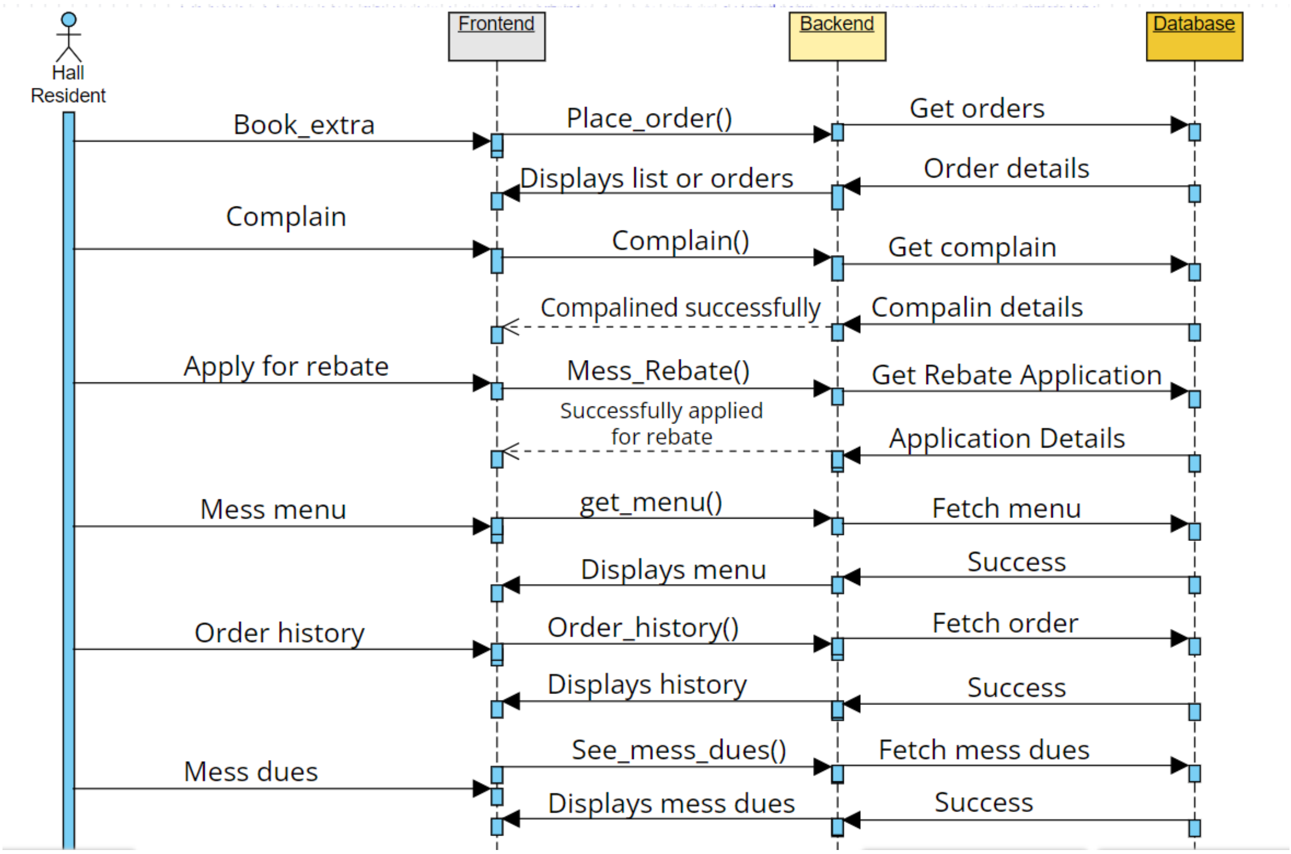
Mess Automation System

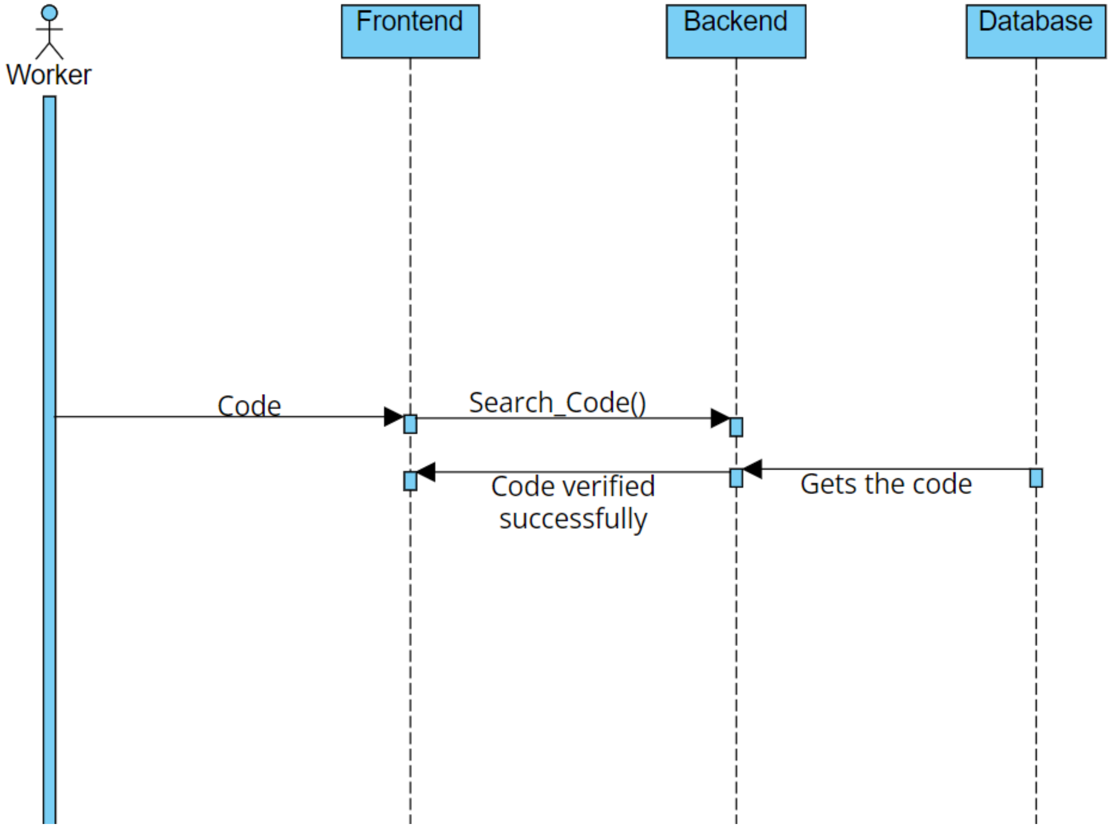


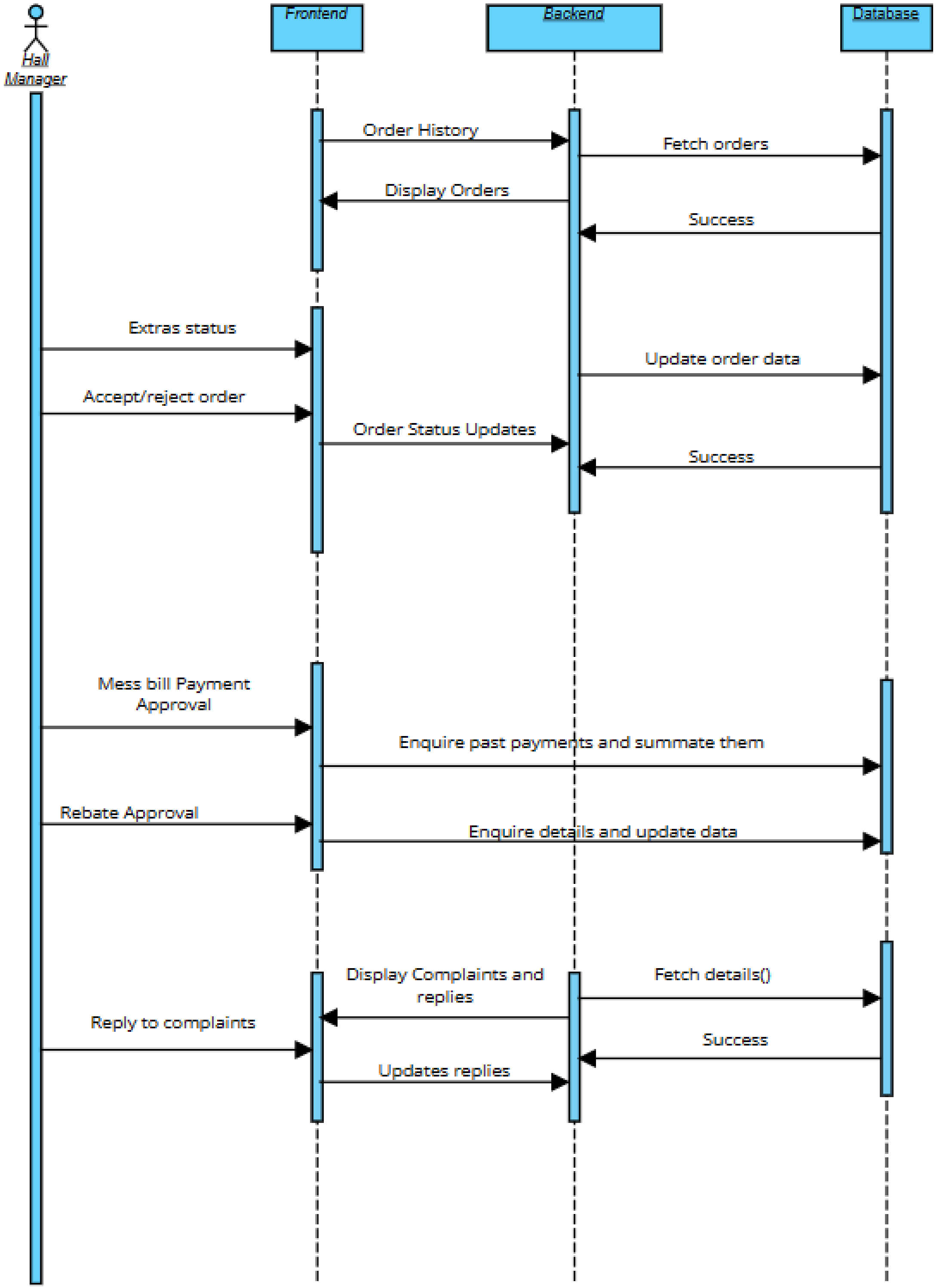
3.3 Sequence Diagrams



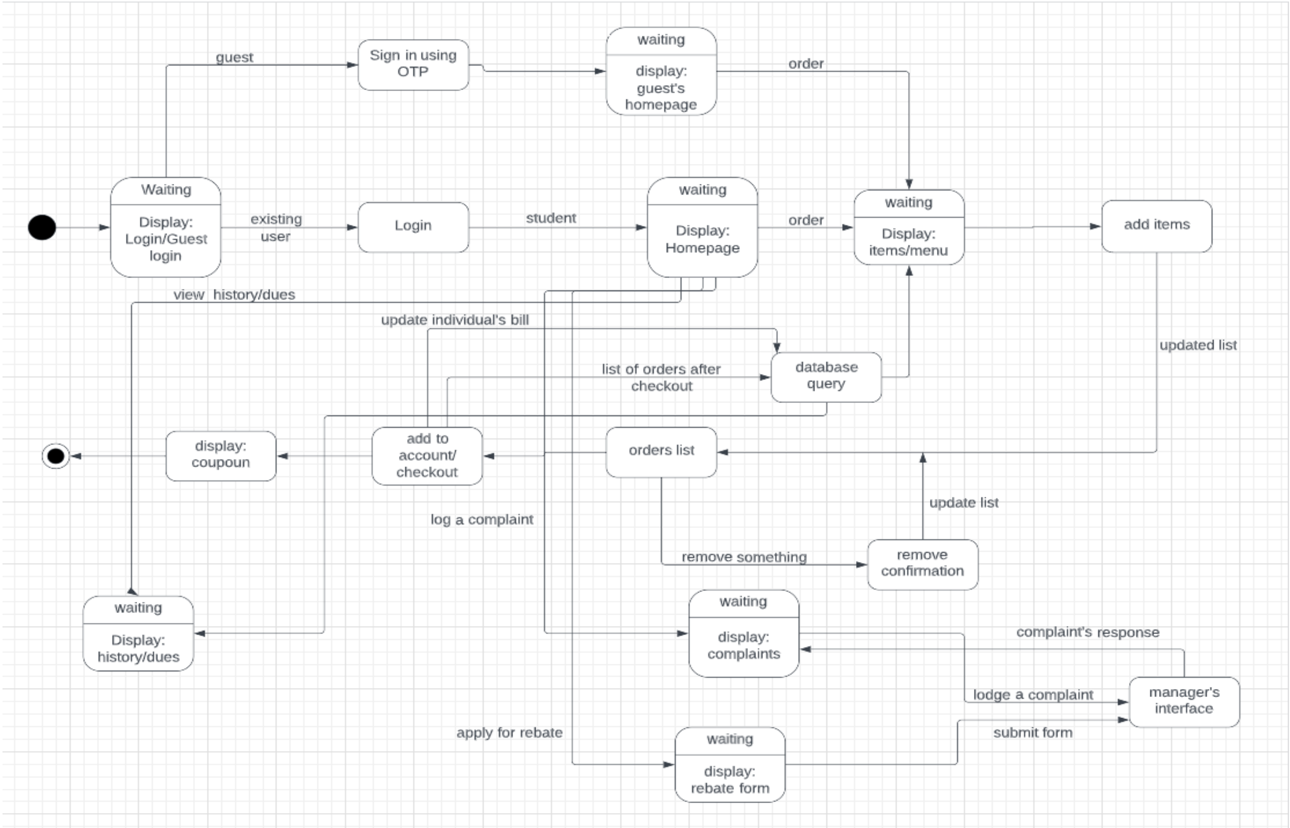

Guest
Frontend
Backend
Database
Meal Name
Book_Meal()
Stores the data of the meal
Booking successful
Shows booked meal with date and price
Mess menu
Shows mess menu of a particular month according to weekdays
Extra item name
Book_Extra_item()
Stores the data of the extra item
Booking successful
Shows booked extra item with date and price

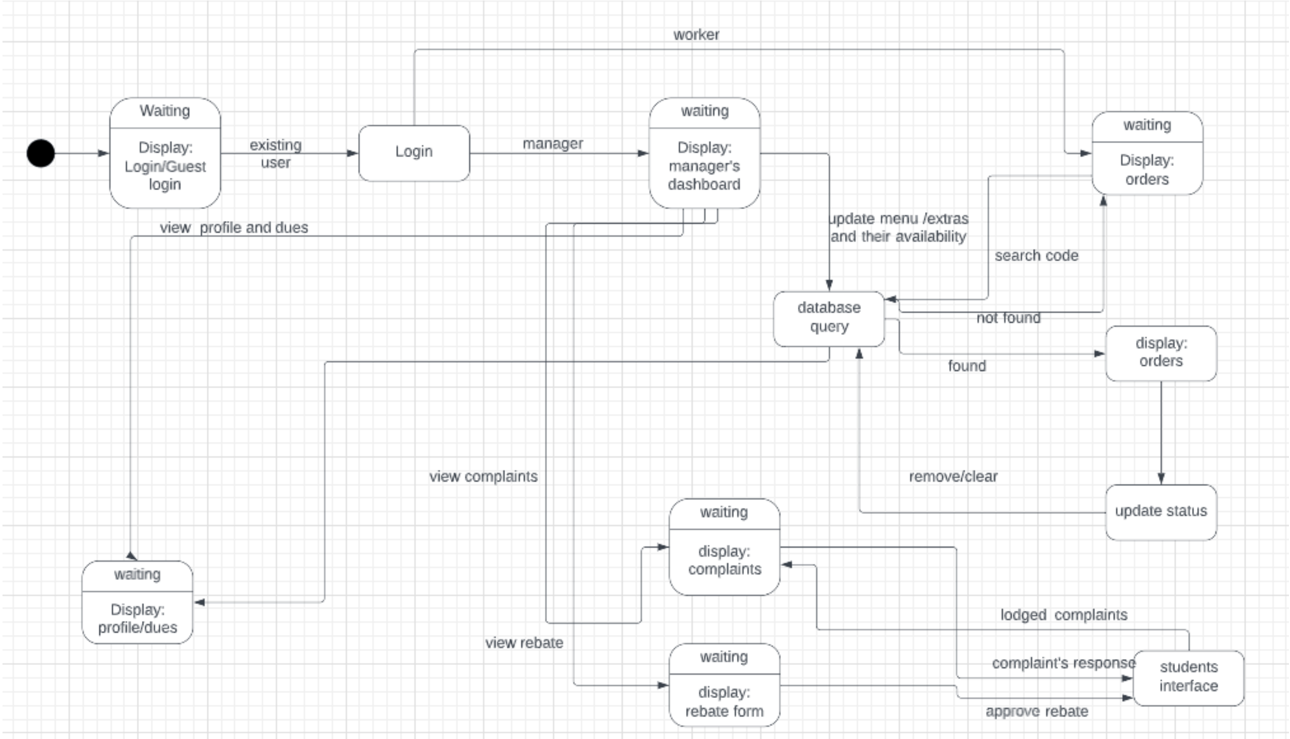

Hall Resident
Frontend
Backend
Database
Book_extra
Place_order()
Get orders
Displays list or orders
Order details
Complain
Complain()
Get complain
Compalined successfully
Compalin details
Apply for rebate
Mess_Rebate()
Get Rebate Application
Successfully applied for rebate
Application Details
Mess menu
get_menu()
Fetch menu
Displays menu
Success
Order history
Order_history()
Fetch order
Displays history
Success
Mess dues
See_mess_dues()
Fetch mess dues
Displays mess dues
Success


Worker
Frontend
Backend
Database
Code
Search_Code()
Code verified successfully
Gets the code


Hall Manager
Frontend
Backend
Database
Order History
Fetch orders
Display Orders
Success
Extras status
Update order data
Accept/reject order
Order Status Updates
Success
Mess bill Payment Approval
Enquire past payments and summate them
Rebate Approval
Enquire details and update data
Display Complaints and replies
Fetch details()
Reply to complaints
Success
Updates replies

3.4 State Diagrams





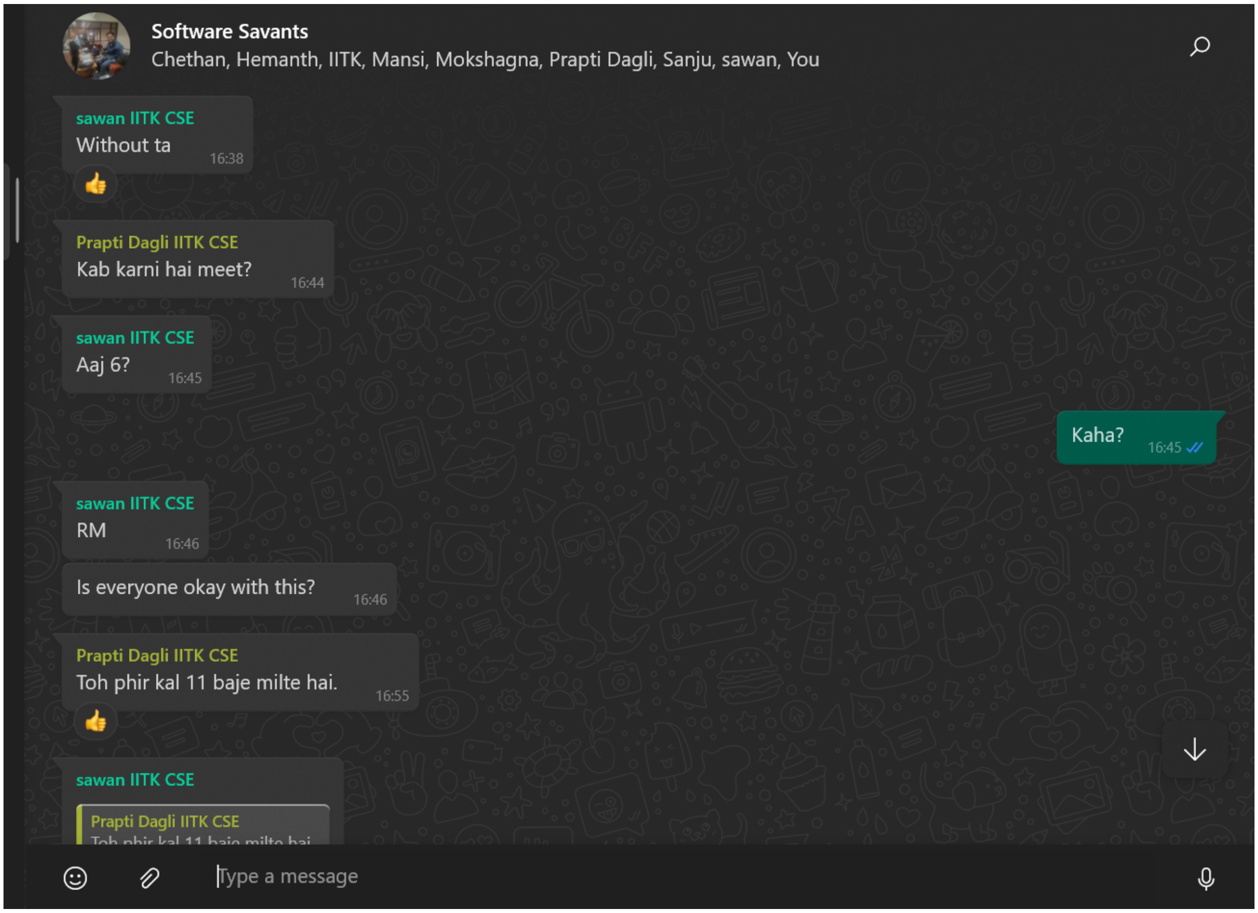
4 Project Plan

4.1 Communication

● The team communicates internally via a Whatsapp group both via messaging and voice calls.

● The team meets regularly to discuss the ongoings of the project and future plans.

● Smooth communication between the teammates is ensured and regular updates are provided on the work completed.



Software Savants
Chethan, Hemanth, IITK, Mansi, Mokshagna, Prapti Dagli, Sanju, sawan, You
Type a message

sawan IITK CSE
Without ta 16:38
👍

Prapti Dagli IITK CSE
Kab karni hai meet? 16:44

sawan IITK CSE
Aaj 6? 16:45

Kaha? 16:45 ✓✓

sawan IITK CSE
RM 16:46

Is everyone okay with this? 16:46

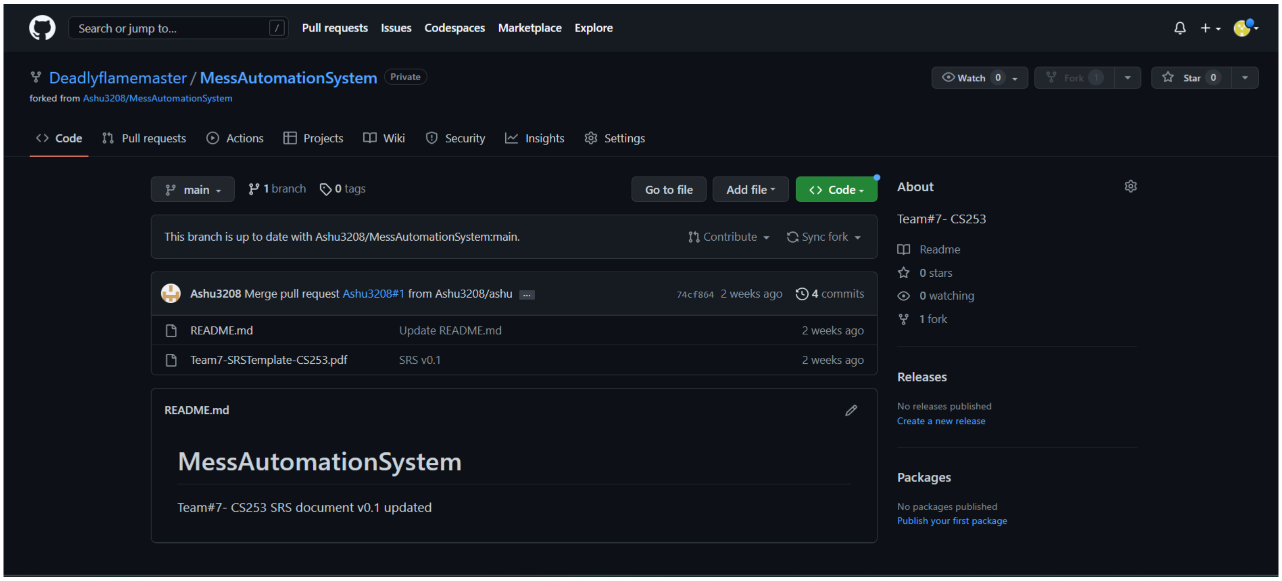
Prapti Dagli IITK CSE
Toh phir kal 11 baje milte hai. 16:55
👍

sawan IITK CSE
Prapti Dagli IITK CSE
Toh phir kal 11 baje milte hai

4.2 Code Collaboration

- A repository has been set up on GitHub with all the teammates as members for storing and collaborating on the source code of this project. .

- We aim to keep high code coverage of the codebase via unit tests.



Deadlyflamemaster / MessAutomationSystem Private
forked from Ashu3208/MessAutomationSystem

1 branch 0 tags

This branch is up to date with Ashu3208/MessAutomationSystem:main.

Ashu3208 Merge pull request Ashu3208#1 from Ashu3208/ashu ... 74cf864 2 weeks ago 4 commits

README.md	Update README.md	2 weeks ago
Team7-SRSTemplate-CS253.pdf	SRS v0.1	2 weeks ago

README.md

MessAutomationSystem

Team#7- CS253 SRS document v0.1 updated

About

Team#7- CS253

Readme
0 stars
0 watching
1 fork

Releases

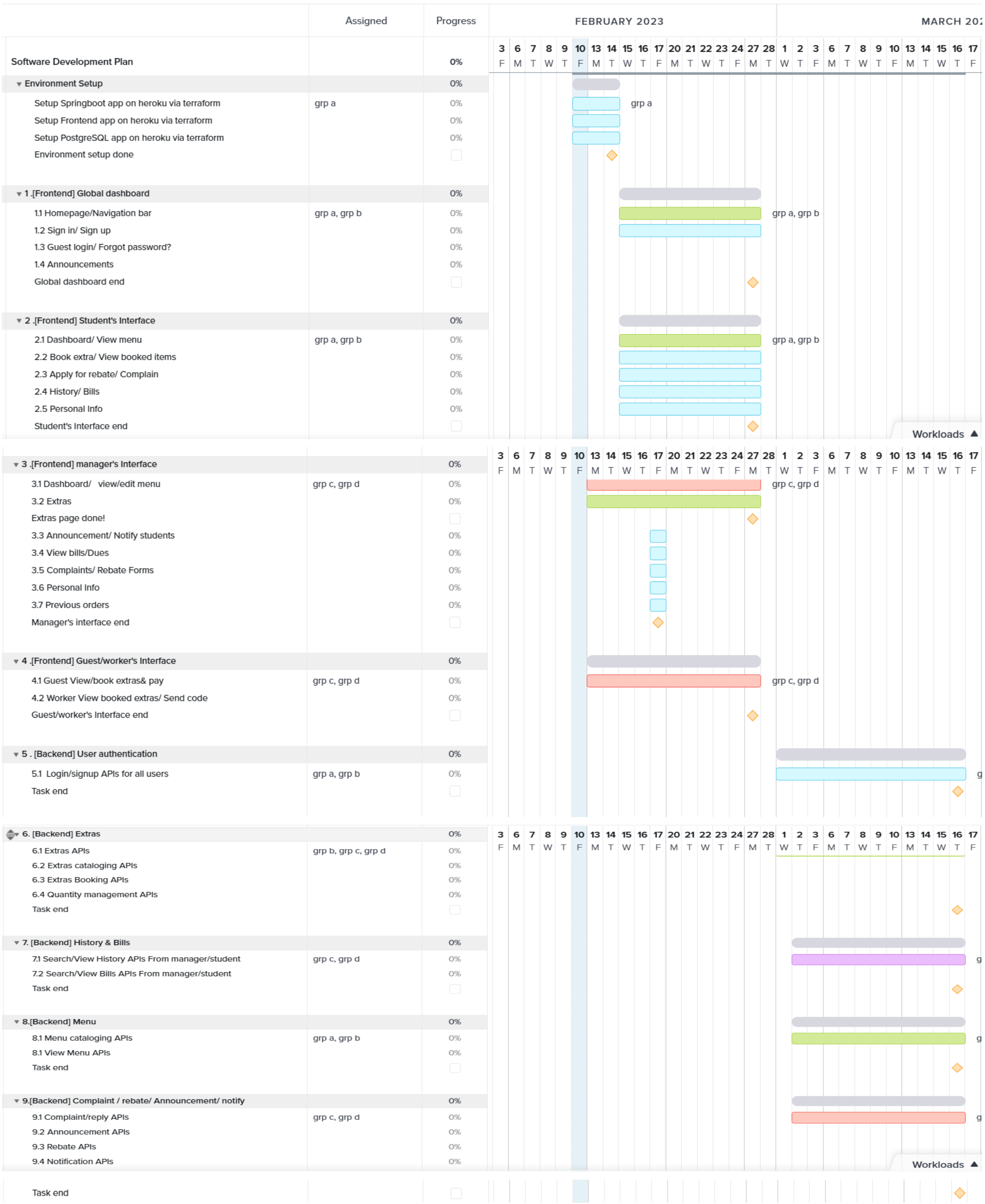
No releases published
Create a new release

Packages

No packages published
Publish your first package

4.3 Project Planning

- We have set up a JIRA board for the project planning and developing a Gantt Chart.
- Work has been divided into Epics, which is further divided into Stories.
- We will have weekly sprints where each team member will have clearly defined work.
- Every team member has internally taken the responsibility to focus majorly on either frontend or the backend.
- Our current Gantt chart looks like this:



	Assigned	Progress
Software Development Plan		0%
▾ Environment Setup		0%
Setup Springboot app on heroku via terraform	grp a	0%
Setup Frontend app on heroku via terraform		0%
Setup PostgreSQL app on heroku via terraform		0%
Environment setup done		□
▾ 1 .[Frontend] Global dashboard		0%
1.1 Homepage/Navigation bar	grp a, grp b	0%
1.2 Sign in/ Sign up		0%
1.3 Guest login/ Forgot password?		0%
1.4 Announcements		0%
Global dashboard end		□
▾ 2 .[Frontend] Student's Interface		0%
2.1 Dashboard/ View menu	grp a, grp b	0%
2.2 Book extra/ View booked items		0%
2.3 Apply for rebate/ Complain		0%
2.4 History/ Bills		0%
2.5 Personal Info		0%
Student's Interface end		□
▾ 3 .[Frontend] manager's Interface		0%
3.1 Dashboard/ view/edit menu	grp c, grp d	0%
3.2 Extras		0%
Extras page done!		□
3.3 Announcement/ Notify students		0%
3.4 View bills/Dues		0%
3.5 Complaints/ Rebate Forms		0%
3.6 Personal Info		0%
3.7 Previous orders		0%
Manager's interface end		□
▾ 4 .[Frontend] Guest/worker's Interface		0%
4.1 Guest View/book extras& pay	grp c, grp d	0%
4.2 Worker View booked extras/ Send code		0%
Guest/worker's Interface end		□
▾ 5 . [Backend] User authentication		0%
5.1 Login/signup APIs for all users	grp a, grp b	0%
Task end		□
▾ 6. [Backend] Extras		0%
6.1 Extras APIs	grp b, grp c, grp d	0%
6.2 Extras cataloging APIs		0%
6.3 Extras Booking APIs		0%
6.4 Quantity management APIs		0%
Task end		□
▾ 7. [Backend] History & Bills		0%
7.1 Search/View History APIs From manager/student	grp c, grp d	0%
7.2 Search/View Bills APIs From manager/student		0%
Task end		□
▾ 8.[Backend] Menu		0%
8.1 Menu cataloging APIs	grp a, grp b	0%
8.1 View Menu APIs		0%
Task end		□
▾ 9.[Backend] Complaint / rebate/ Announcement/ notify		0%
9.1 Complaint/reply APIs	grp c, grp d	0%
9.2 Announcement APIs		0%
9.3 Rebate APIs		0%
9.4 Notification APIs		0%
Task end		□

Appendix A - Group Log

Since the beginning of the project, our entire team has been very enthusiastic.We have formed a Whatsapp group for effective communication.We have recently made a private repo on Github where we do collaboration.

Meeting minutes	Agenda
29 Jan 2023 7:00-8:00 pm	Discuss the initial ideas on the design document and various aspects of it. Brainstorm on the architecture and frontend and backend framework.
5 Feb 2023 11:00 am-12:30 pm	Done a significant portion of the design document and distributed the remaining work among all the teammates.
9 Feb 2023 6:00-7:00 pm	Updated the TA with the current progress and also gave a short presentation of SDS to the TA.
10 Feb 2023 7:00-9:00 pm	Completed all the sections of the SDS and carefully reviewed the SDS.