1. Introduction

1.1 Introduction

The application effectively manages the mess halls. The system allows users to access and mark their attendance through this app. It also enables customers to select dishes from the available menu of the day. The menu will be decided on the choice of the Student. The Student will mark their choice of dishes for upcoming days and the mess manager or admin will provide the dish based on the student selection. The app has the functionality which allow user to mark attendance and select the menu prior of the opening time of mess. By taking the attendance there will be less wastage of food and the mess managers can manage the mess in a better way with the staff. The app also gives the notification to the user for attendance.

The app also has different features for both students and managers. The manager will be the admin of the app and have access to set today's menu and see today's attendance i.e. the number of students who are coming to mess. The managers also have the functionality to set future menu for upcoming days. Student can give feedback on menu and can also raise complaint. Admin can handle feedbacks and complaints. We believe that the technology can give this type of facility effectively and helps to reduce food wastage.

1.2 Existing System

The Existing system is not completely computerized. There is no facility to see the menu and then mark their attendance according to their choice. Student cannot give feedback and register complain regarding food. There is food wastage if student is not present. Current system is not user friendly and it is manual. Admin cannot keep track of expenses and number of students enrolled in mess. Student feedback can help mess manager to improve quality of food and service. In existing system student cannot select menu prior.

Limitations in Existing System:

- The current system is not completely complete computerized and manual system in entering students data and handling it.
- There is no centralized database maintenance.
- There is no easy access to the particular students record.
- Searching, sorting, updation and data recovery is difficult.
- There is food wastage if student is absent.

1.3 Need for Proposed System

The main aim of any mess is to provide clean and fresh food to the students of the college. In many messes, there is no facility to see the menu and then mark their attendance according to their choice. Here we are providing features of mark attendance and also to see today's menu. It would be possible to reduce the food wastage in mess by marking attendance and without using much efforts and manpower if, there existed a software for the same. Thus, there arises a need to create an app for the same. Such software would make the entire Mess related management an automated system. The app is not only restricted to food attendance and see future menu but also have feature to give feedback and raise complaints.

2. Proposed System

2.1 Problem Statement

Student Mess management system is an android application which supports the management of menu, student attendance. The app has the functionality which allow user to mark attendance and select the menu prior of the opening time of mess. By taking the attendance there will be less wastage of food. Student Mess management system is accessed by entering the username and password which is added to the database. Such software would make the entire Mess related management an automated system.

2.2 Objectives of system

- To reduce manual work
- To provide user-friendly interface
- To quick data retrieval and large amount of data storage
- To reduce food wastage
- To manage mess effectively
- To provide better service to students based on their feedbacks and complaints

2.3 FUNCTIONAL REQUIREMENTS

Application enable students to register for meal, give feedback and raise a issue related to mess.

Support food wastage management, attendance management, easy to use.

Helps to track the expenses of mess and view daily meal item.

Software Requirements : -

Operating System : Windows 7/8/10 ,Linux

Front End : Java

Platform : Android Studio

Database : Firebase

Hardware Requirements:-

Ram : 8 GB

Hard Disk : Minimum 512 MB (or more)

2.4 Scope of the System

An android Mess app helps automate the various tasks that are carried out by mess management and attendees. Each student can create his/her own profile by registering on the app using an email id and student registration number. The mess management provides the list of upcoming meals that are shown on student’s feeds.

This app has two different portals for management and students. Mess managers can upload the next day's menu for all three meals. Students can register themselves for whichever meals they want to have. Students can register themselves for whichever meals they want to have. Facility for both the managers and student to see there monthly expenses. Add a complaint or suggestion regarding any mess activity. Feedback can be given on any meal with ratings. The managers should be able to view each students activity.

2.5 Modules :

Meal

- Add Meal – Admin can add meal with cost and date.
- View Meal –Students can view the details of meal.
- Update Meal – Admin can update the details of meal.
- Delete Meal – If needed, admin can also delete the details of meal.
- Register for Meal – Student can register for meal.

Attendance

- Add attendance – Admin can mark attendance of student by scanning QR code.
- View attendance – Admin and student can view the attendance.

Expenses

- Add expenses – Admin can also add expense details.
- View expenses – Admin can see details of expenses.
- Delete expenses – Records of expenses can be deleted by admin.

Feedback

- Give feedback – Student can give feedback of meal .
- View feedback – Admin can view the feedback.

Complaint

- Add complaint – Student can register complaint related to mess.
- View complaint – Admin can view the complaint.

2.6 Brief overview of the technology

1. **Android**

   Android is a mobile operating system developed by Google, based on the Linuxkernel and designed primarily for touchscreen mobile devices such as smart-phones and tablets. Android software development is the process by which newapplications are created for the Android operating system. Applications areusually developed in Java programming language using the Android softwaredevelopment kit (SDK) integrated with Android Studio.Android Studio is theofficial integrated development environment (IDE) for the Android platform**.**

2. **XML**

   Extensible Markup Language (XML) is a markup language that defines a set of rules for encoding documents in a format that is both human-readable and machine-readable. The design goals of XML emphasize simplicity, generality,and usability across the Internet. It is a textual data format with strong support via Unicode for different human languages. Although the design of XML focuses on documents, the language is widely used for the representation of arbitrary data structures such as those used in web services.

3. **Firebase**

   The Firebase Realtime Database lets you build rich, collaborative applications by allowing secure access to the database directly from client-side code. Data is persisted locally, and even while offline, realtime events continue to fire, giving the end user a responsive experience. When the device regains connection, the Realtime Database synchronizes the local data changes with the remote updates that occurred while the client was offline, merging any conflicts automatically.

   The Realtime Database provides a flexible, expression-based rules language, called Firebase Realtime Database Security Rules, to define how your data should be structured and when data can be read from or written to. When integrated with Firebase Authentication, developers can define who has access to what data, and how they can access it.

   The Realtime Database is a NoSQL database and as such has different optimizations and functionality compared to a relational database. The Realtime Database API is designed to only allow operations that can be executed quickly. This enables you to build a great realtime experience that can serve millions of users without compromising on responsiveness.

3. Requirement Determination And Analysis :

For anyone who owns a smartphone or some other kinds of mobile devices, he probably uses mobile apps to play games, buy things online , to listen music to read books , to get directions etc.

He can easy easily install the app from the Play store at 0 cost. After installing the app from store, user can make his/her unique Id's to login and use the features of the app.

3.1 Fact Finding Methods :

1.Research and site visit:

Analyst has to research with data of the related items.The data could be collect from the app files or from computer.Analyst can get data and information of their existing from web site.

2. Observation :

This is special purpose document that allows the analyst to collect information and opinions from respondents.Questionaries become useful when a little information is required from a numer of people.

3.2 Feasibility Study :

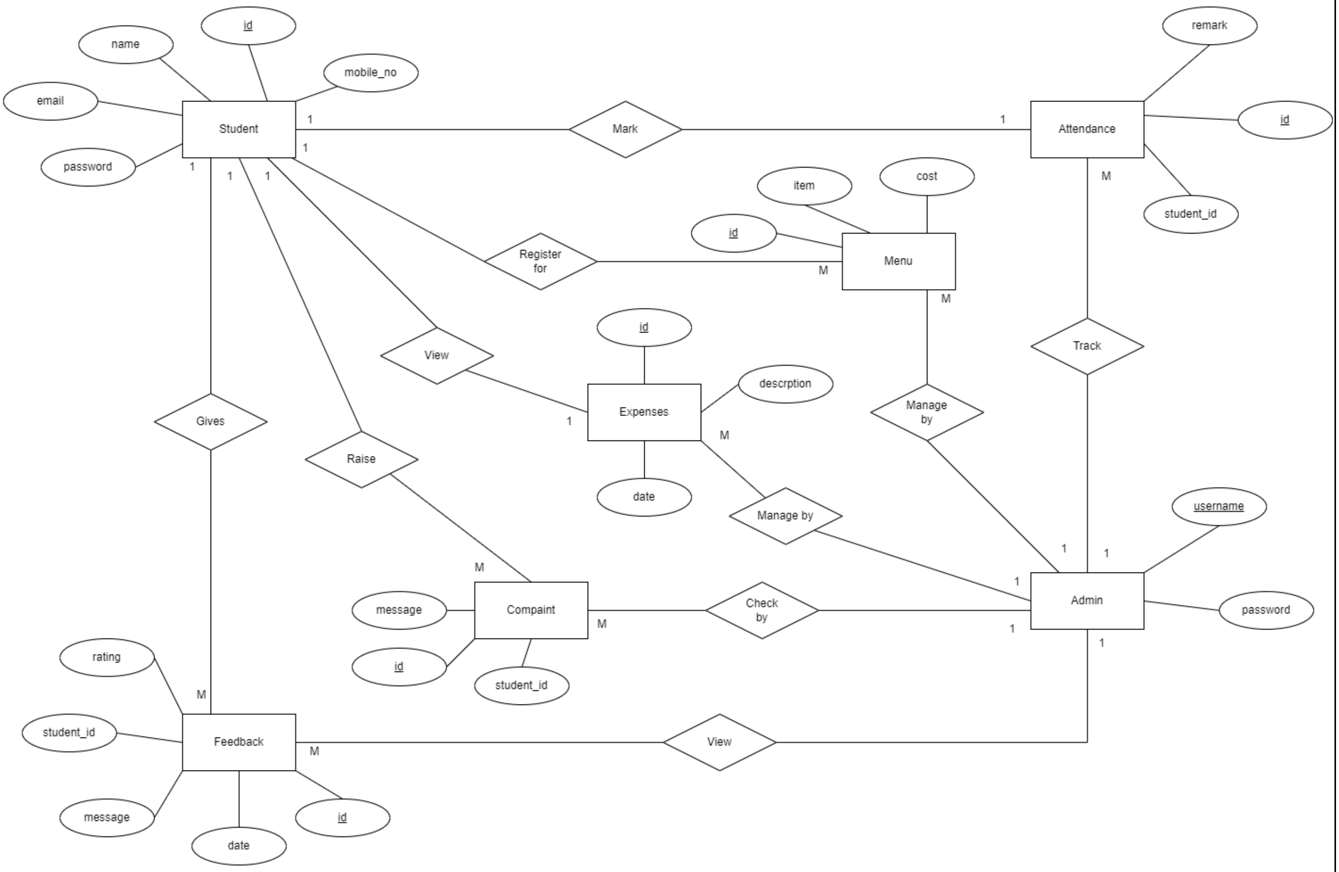
Reseach can be defined as an exploratory study based on elements of quantitative and quantitative data.The feasbility study main source of data colllected through customer requirements.The primary data consists of quantitative part of resident's background such as age,type of family and employment statu & quantitative part of their individual opinions.

Using open coding,categories are named and its properties.At this phase many categories can be generated with any data.Next axial coding categories subcategories round one category at a time ,linking it at the level of properties and dimension.

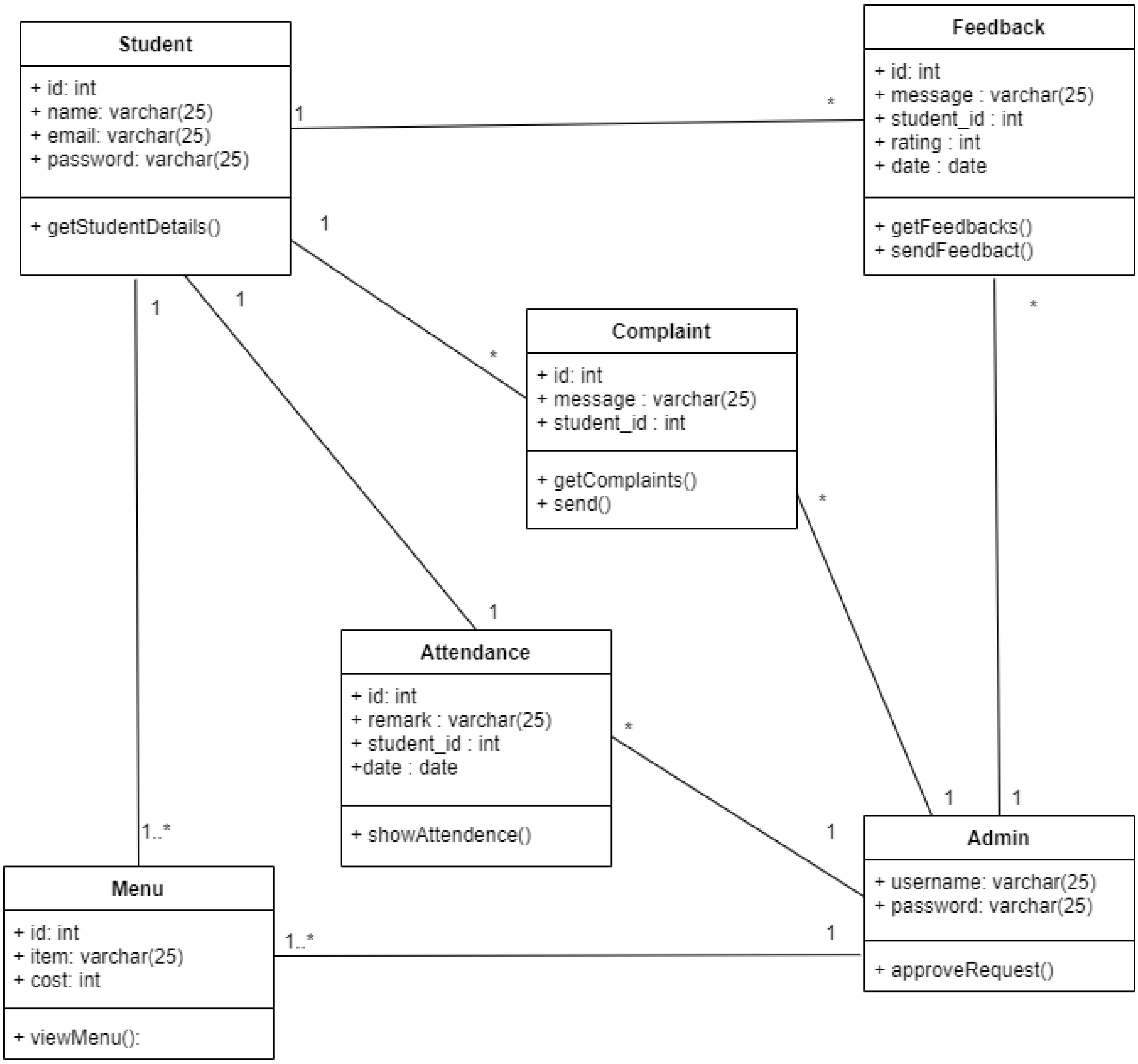
The research purpose is to study and analyse "Student Mess management sytem".

4. **System Design & Analysis**

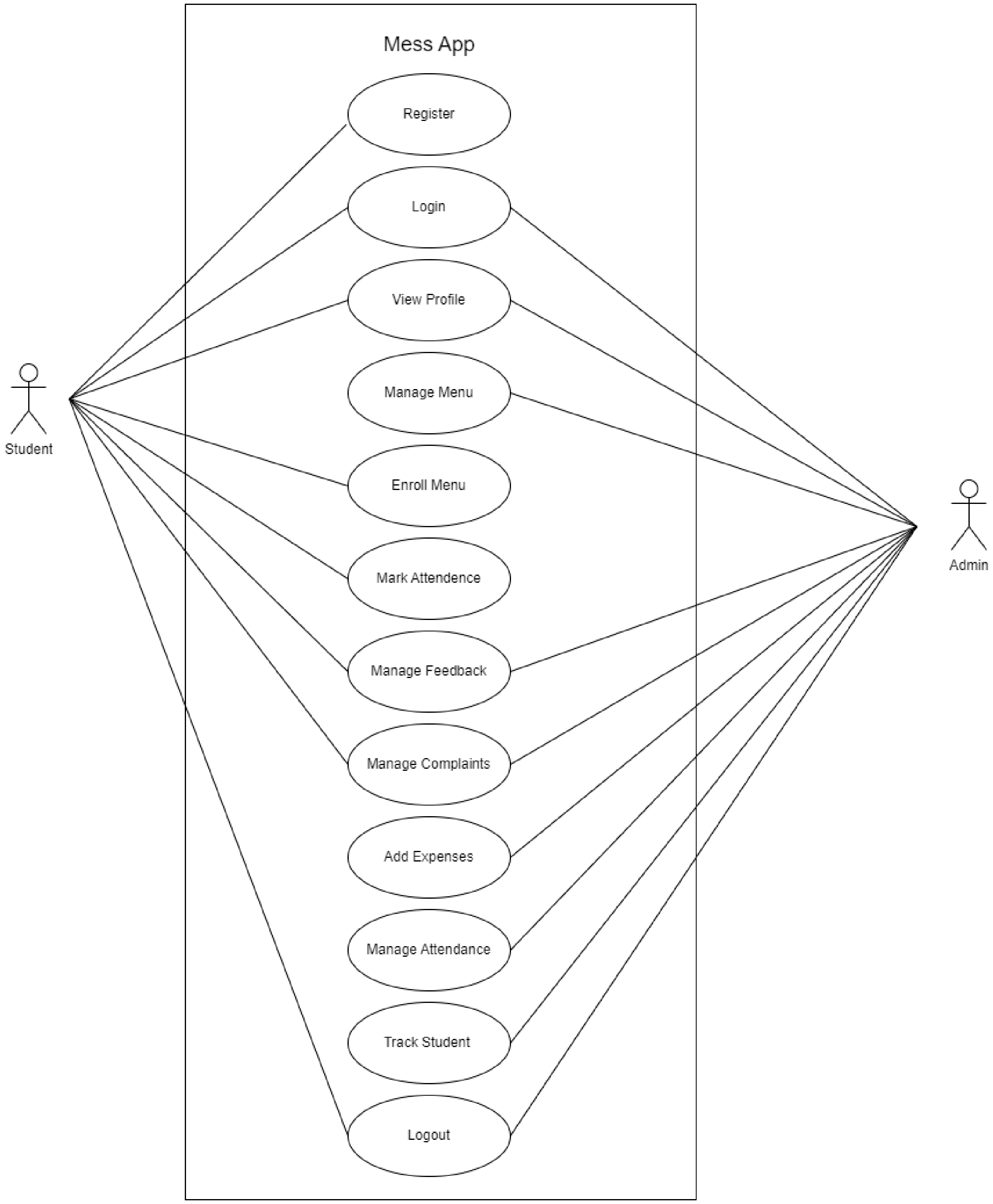
4.1 ER Diagram



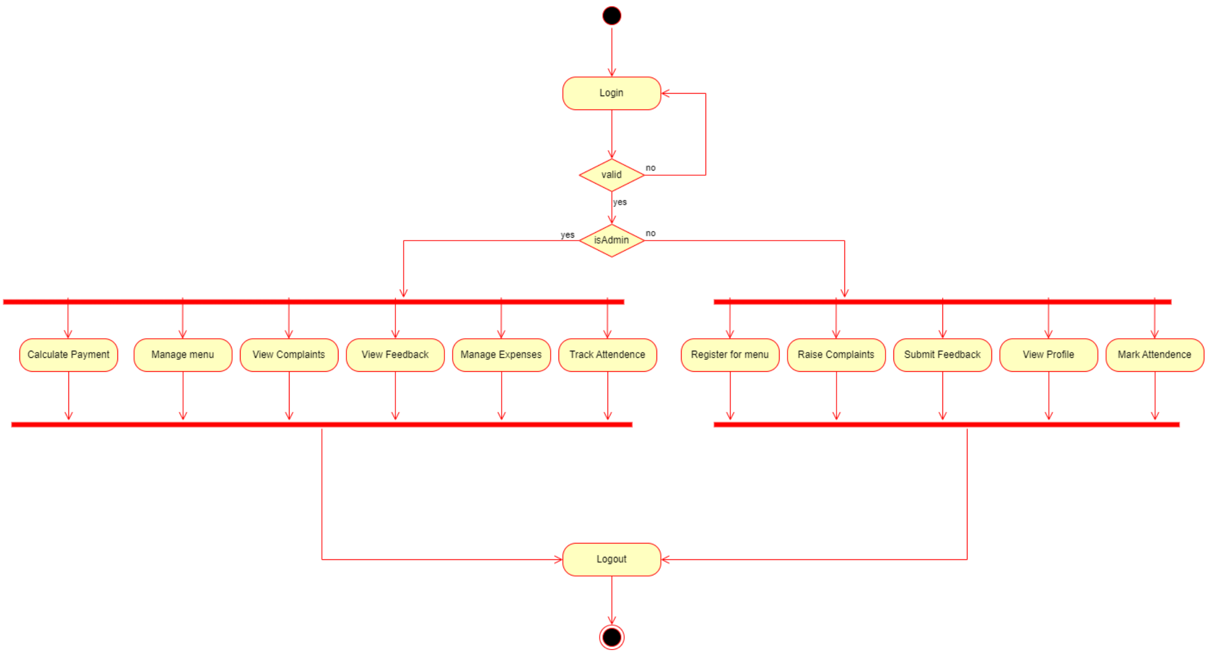
4.2 Class Diagram



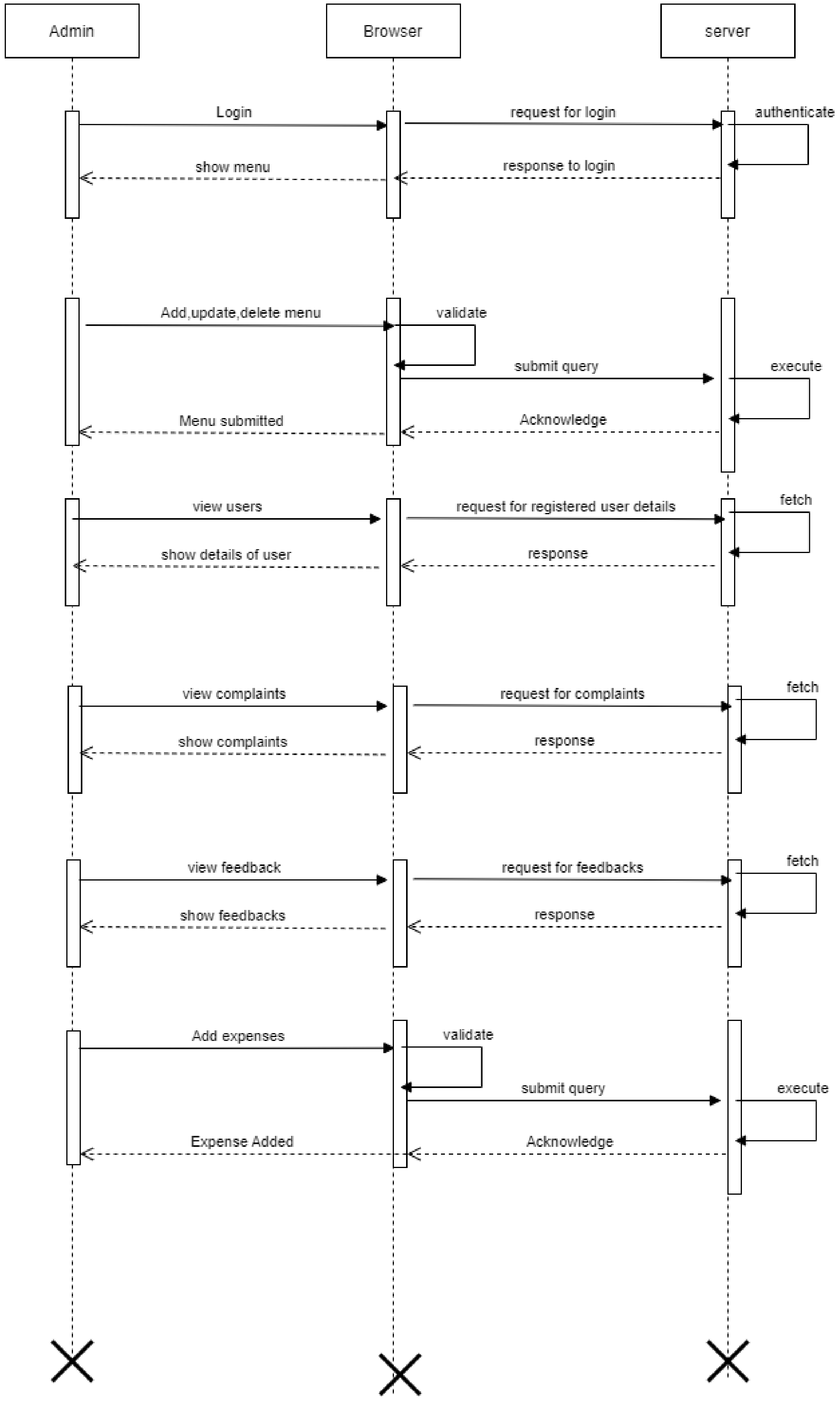
4.3 Use Case Diagram



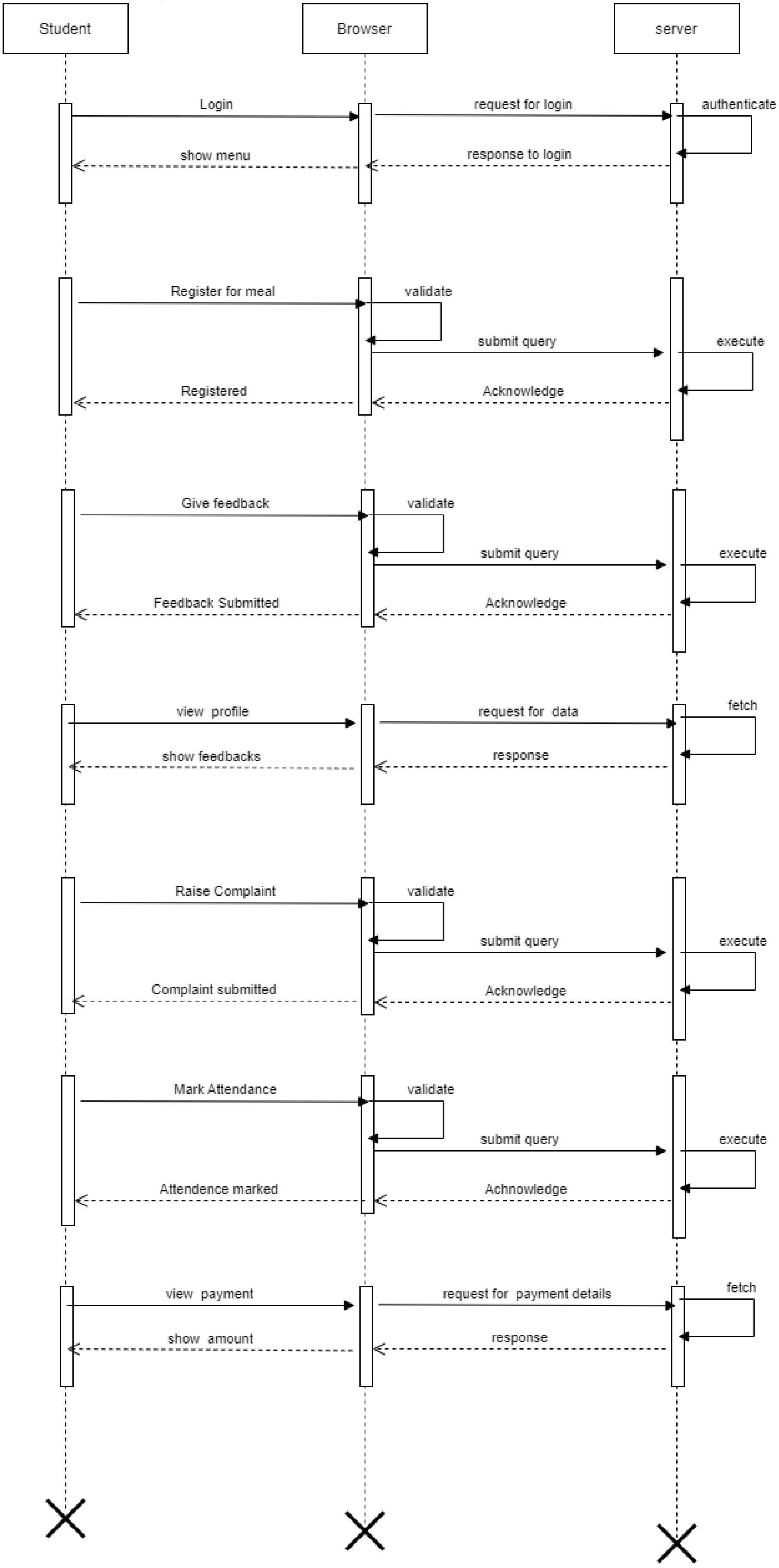
4.4 Activity Diagram



4.5 Sequence Diagram for Admin:



Sequence Diagram for User:



4.6 Data Dictionary

UserTable:

Column Name	Type	constraint	Description
user_id	Int	Primary key	User id
email	Varcha(20)	Not Null	User Email
hostel	Varchar(20)	Not Null	Hostel name
reg_num	Int	Not Null	Registration Number
room_num	Int	Not Null	Room Number

Meal Table:

Column Name	Type	constraint	Description
id	Int	Primary Key	MealId
title	Varchar(25)	Not Null	Meal name
description	Varchar(25)	Not Null	Meal desription
date	Date	Not Null	Addend Date
Reg_status	Boolean	Not Null	Registration Status

UserMeal Table:

Column Name	Type	constraint	Description
Id	Int	Primary Key	User meal Id
balance	Int	Not Null	Balance
User_id	Int	Not Null	User id
isBreakfast	Boolean	Not Null	Breakfast status
isLunch	Boolean	Not Null	Lunch status
isSnacks	Boolean	Not Null	Snacks status
isDinner	Boolean	Not Null	Dinner status

Complaints Table:

Column Name	Type	constraint	Description
id	Int	Primary Key	Complaint Id
email	Vachar(25)	Not Null	Email
message	Varchar(25)	Not Null	Message
name	Varchar(25)	Not Null	User name
User_id	int	Foreignn Key	User id

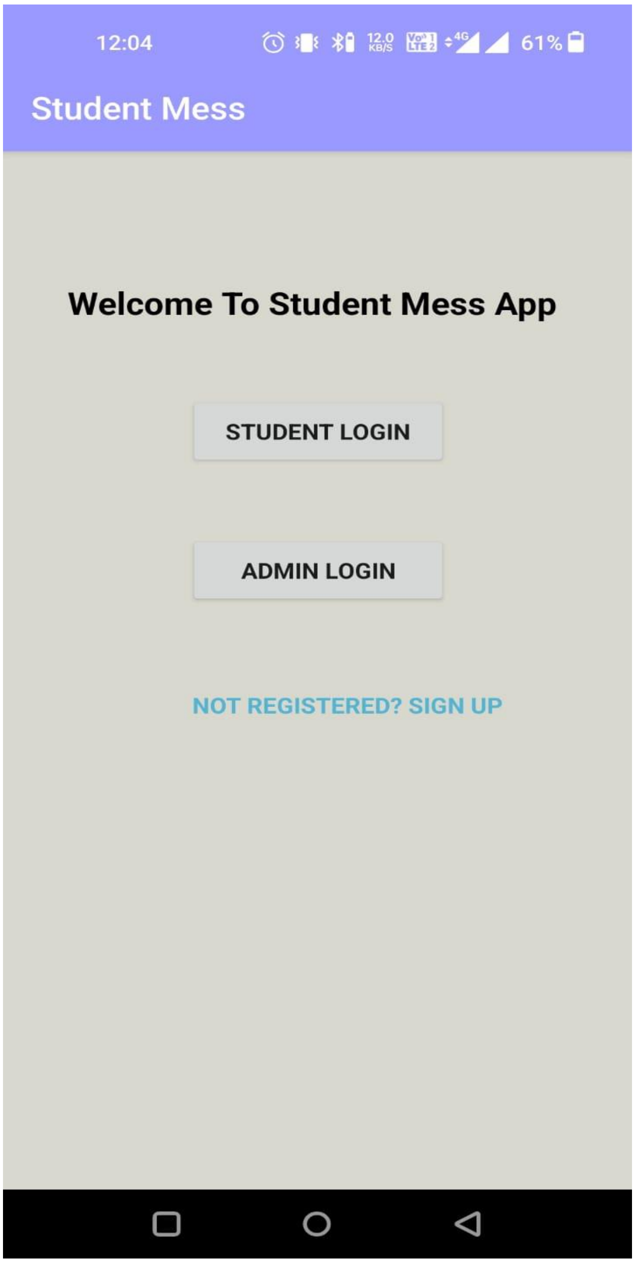
Expenses Table:

Column Name	Type	constraint	Description
Id	Int	Primary Key	Expense Id
cost	int	Not Null	cost
description	Varchar(25)	Not Null	Description
date	date	Not Null	date

Feedback Table:

Column Name	Type	constraint	Description
Id	Int	Primary Key	Complaint Id
meal	Vachar(25)	Not Null	Meal
Feedback	Varchar(25)	Not Null	feedback
rating	int	Not Null	Rating
User_id	int	Foreignn Key	User id
date	date	Not Null	date

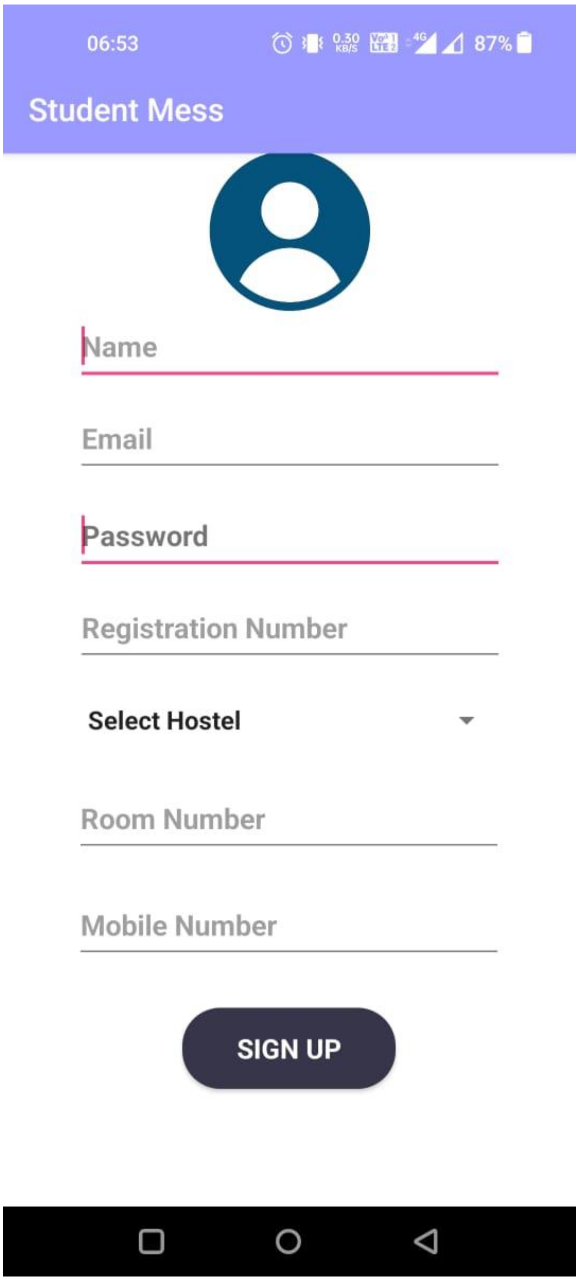
4.7 Output Screens:

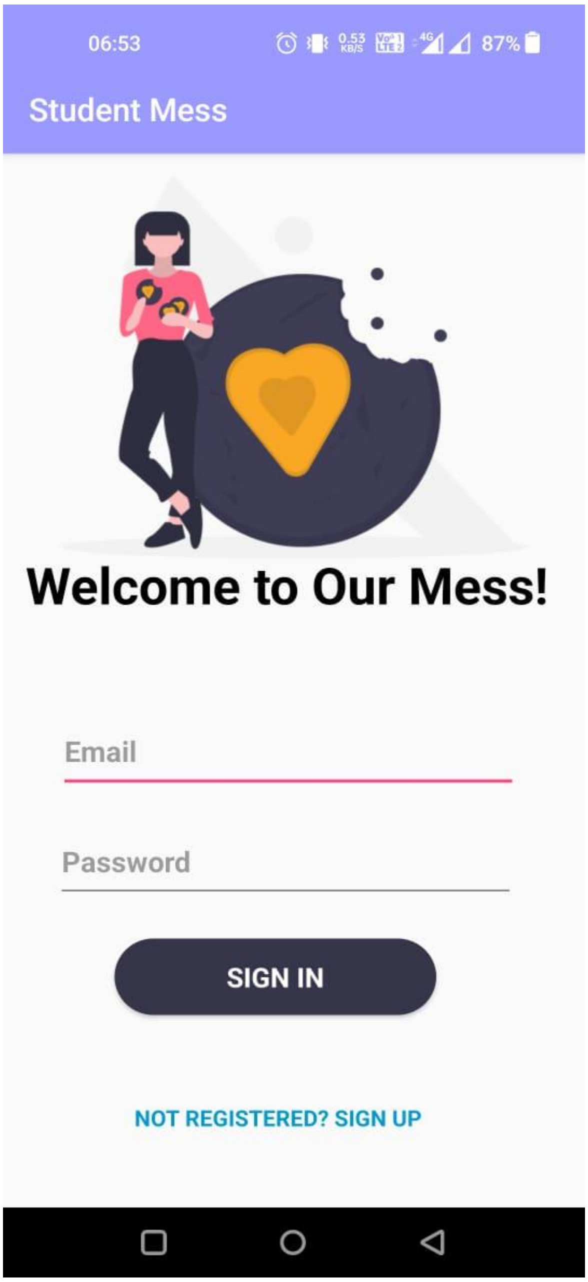


12:04 61%

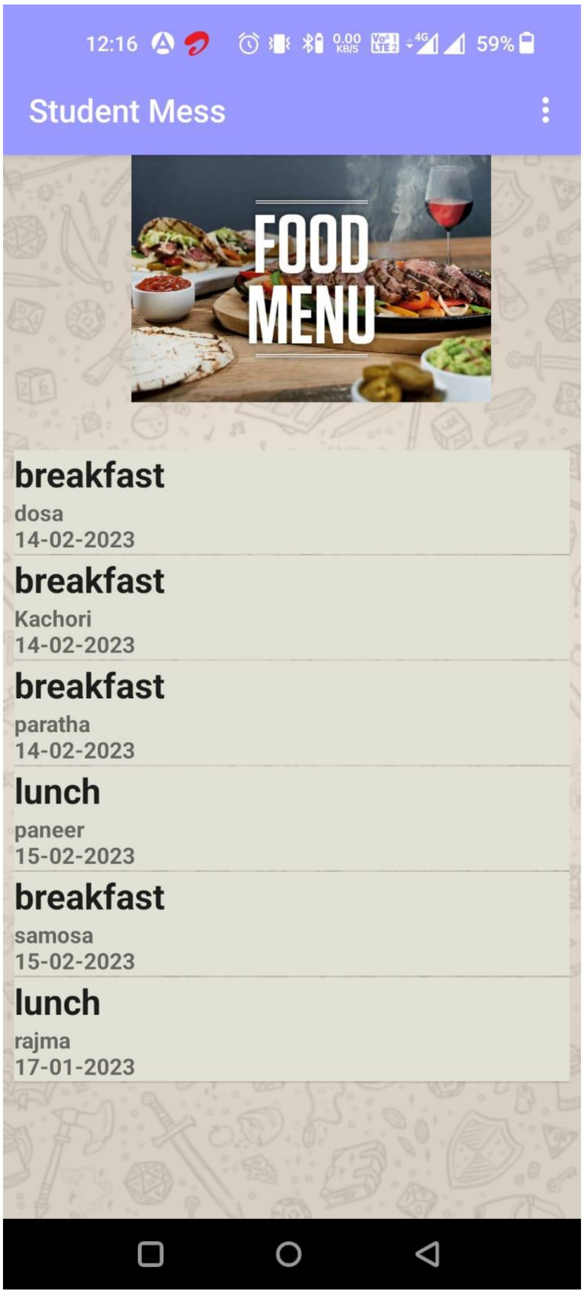
Student Mess

Welcome To Student Mess App





Welcome to Our Mess!



breakfast
dosa
14-02-2023

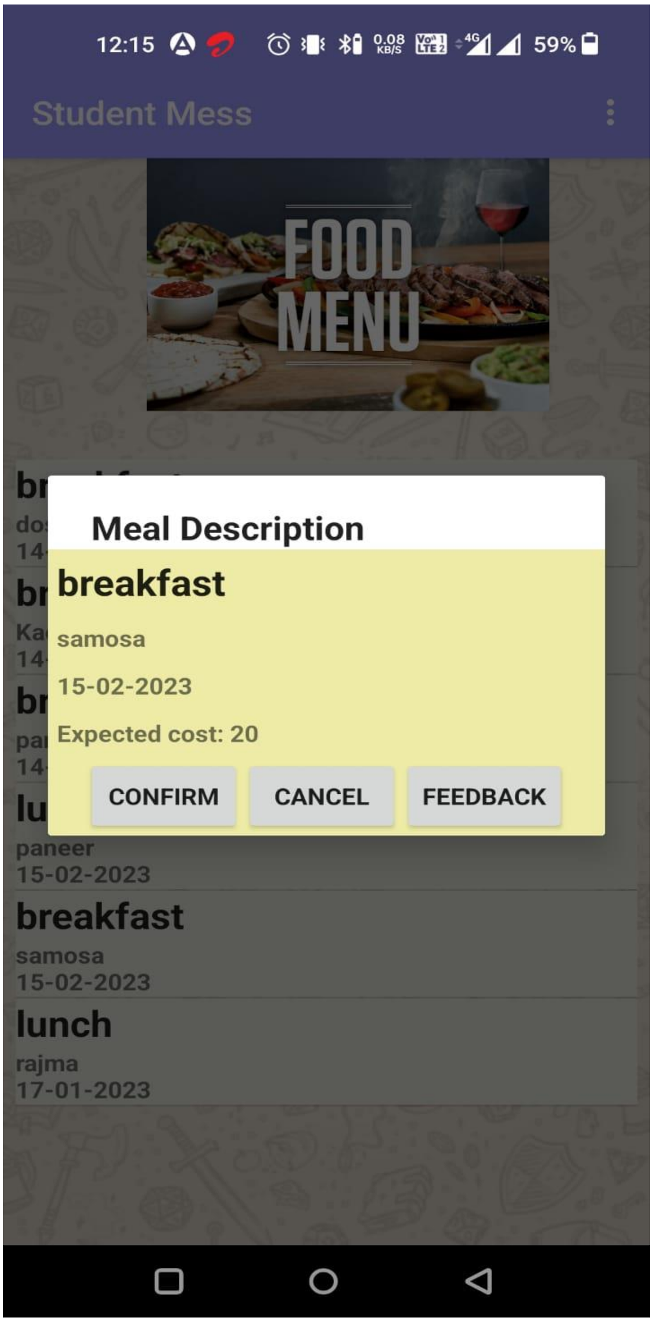
breakfast
Kachori
14-02-2023

breakfast
paratha
14-02-2023

lunch
paneer
15-02-2023

breakfast
samosa
15-02-2023

lunch
rajma
17-01-2023



Meal Description

breakfast

samosa

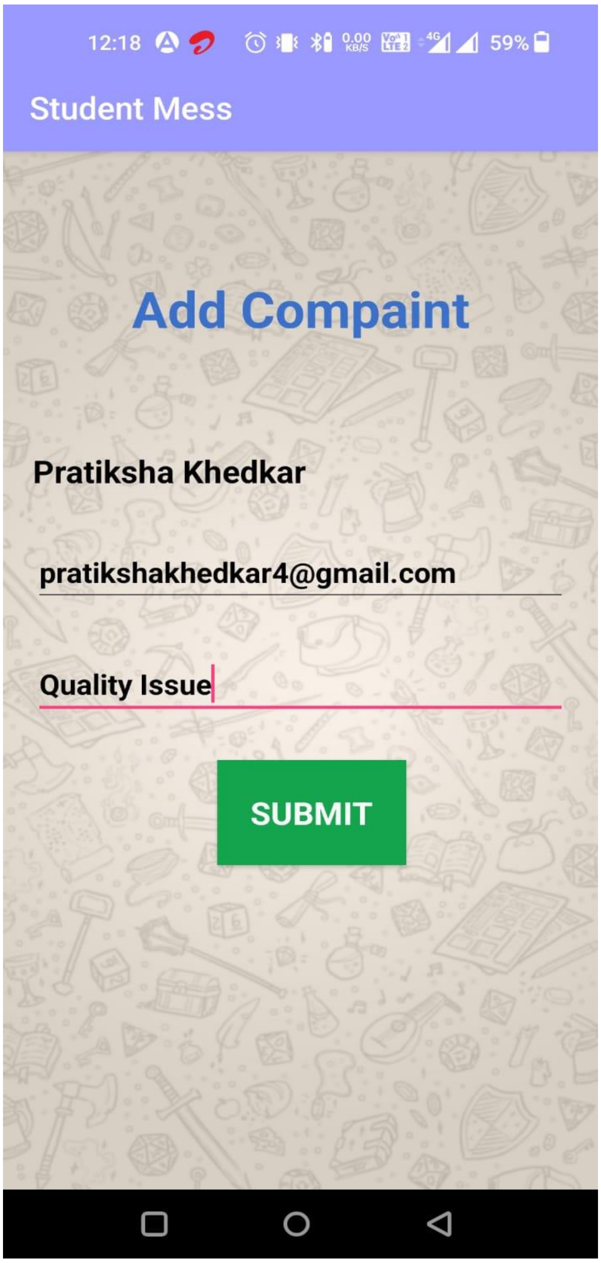
15-02-2023

Expected cost: 20

lu
paneer
15-02-2023

breakfast
samosa
15-02-2023

lunch
rajma
17-01-2023



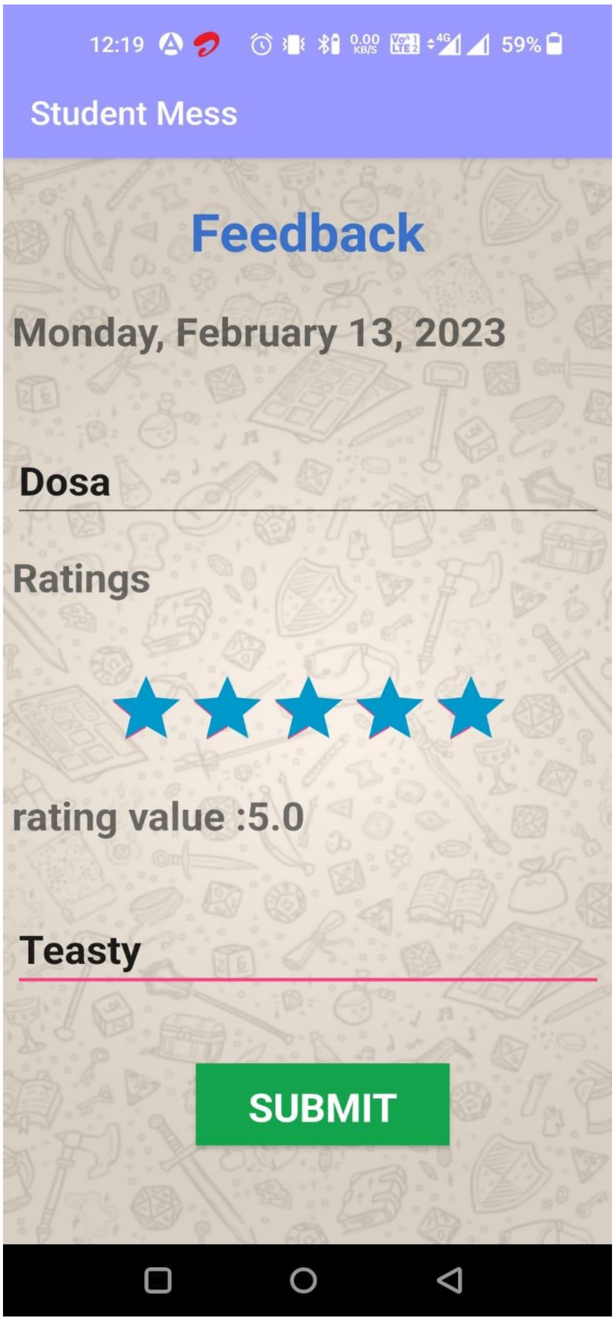
Student Mess

Add Compaint

Pratiksha Khedkar

pratikshakhedkar4@gmail.com

Quality Issue



Feedback

Monday, February 13, 2023

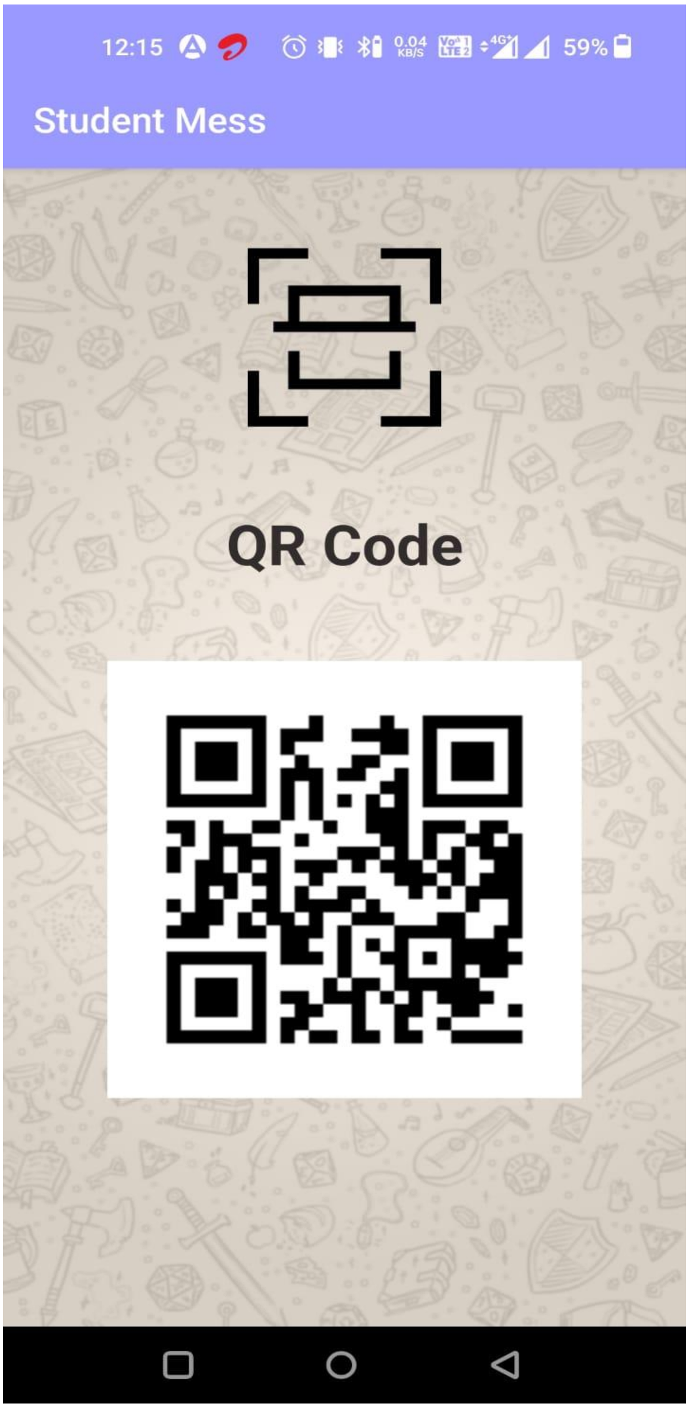
Dosa

Ratings

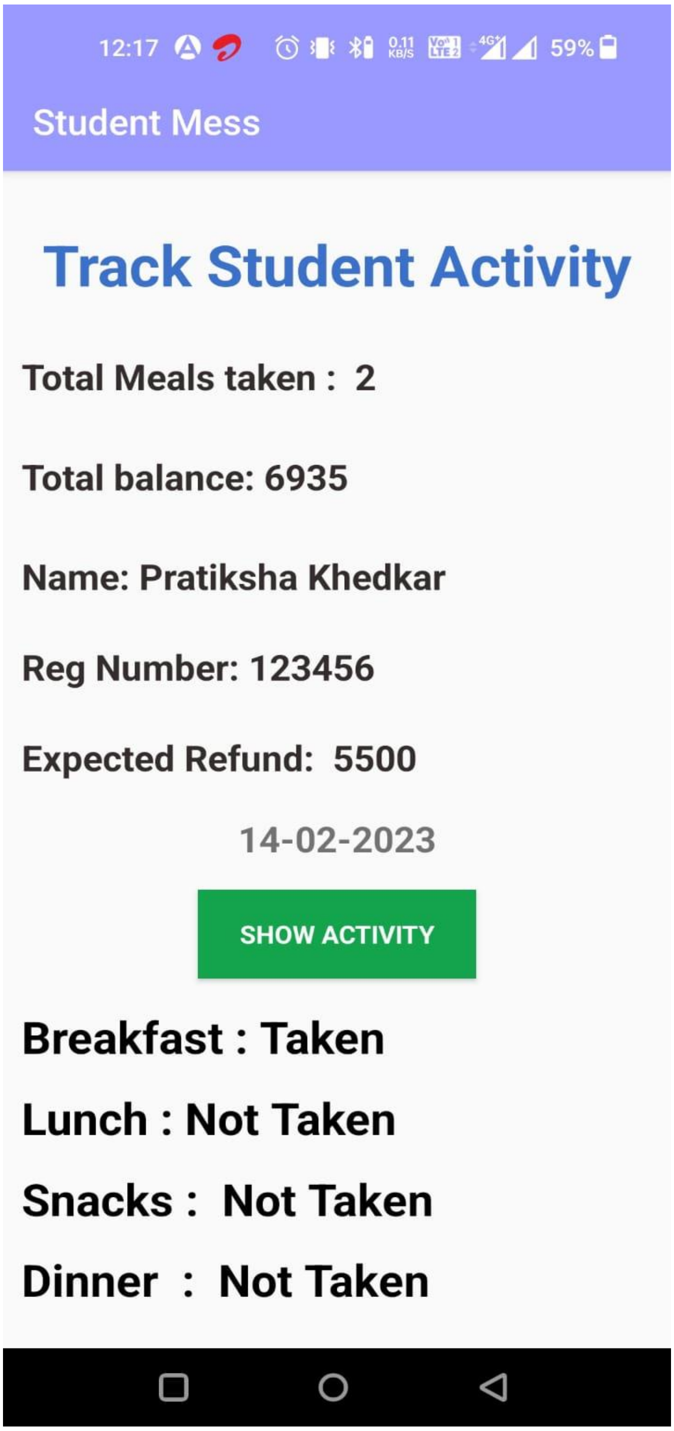
★★★★★

rating value :5.0

Teasty



12:15
59%
Student Mess
QR Code



Student Mess

Track Student Activity

Total Meals taken : 2

Total balance: 6935

Name: Pratiksha Khedkar

Reg Number: 123456

Expected Refund: 5500

14-02-2023

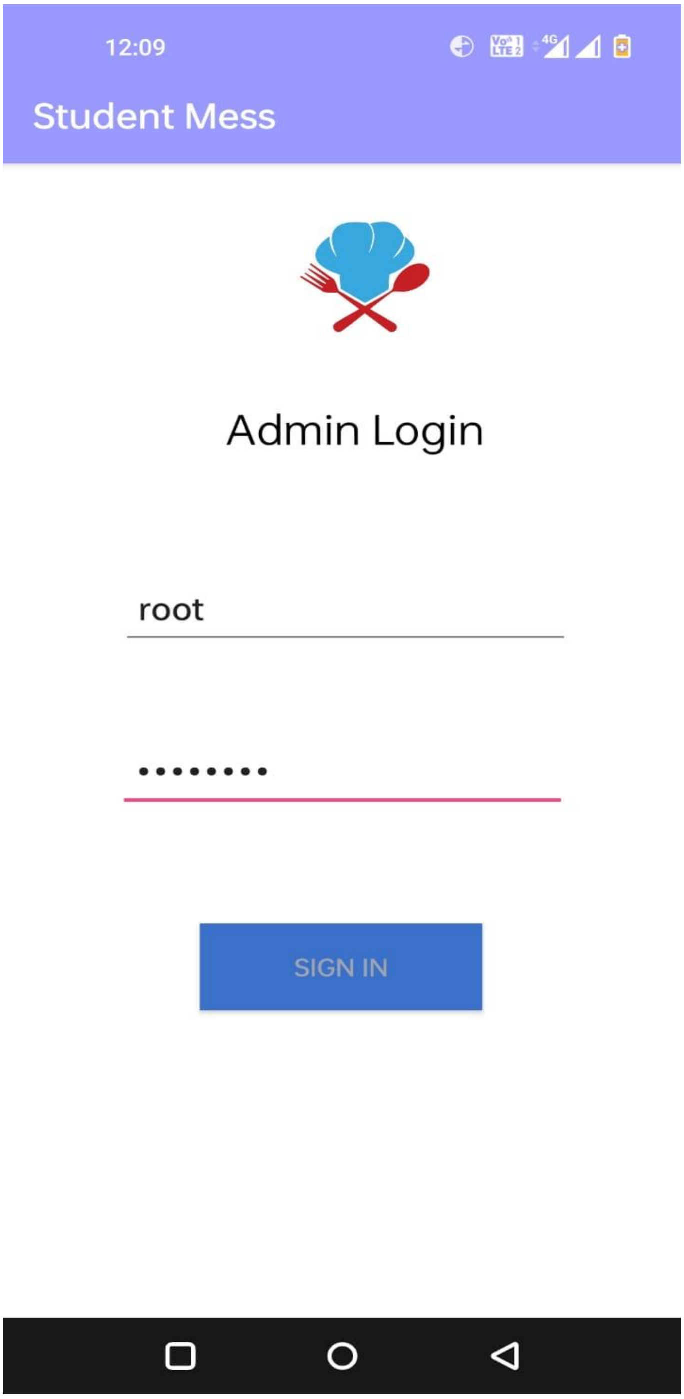
Breakfast : Taken

Lunch : Not Taken

Snacks : Not Taken

Dinner : Not Taken

Admin



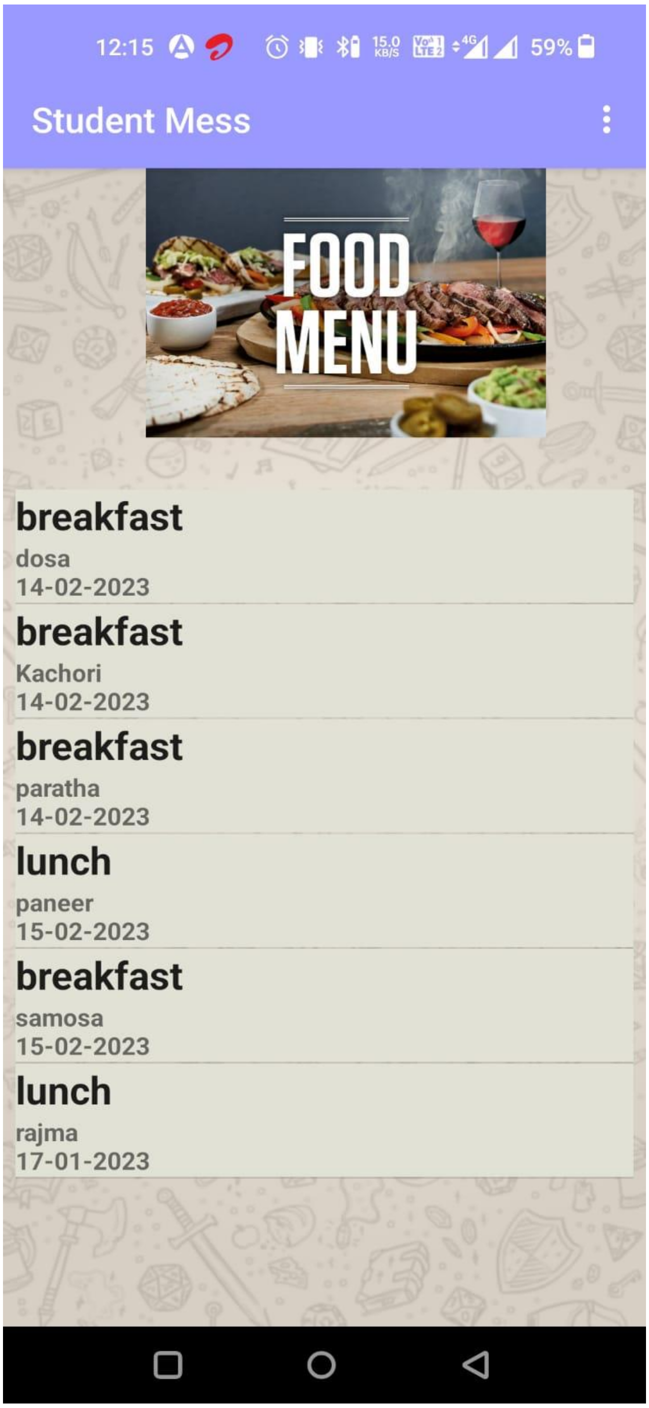
12:09

Student Mess

Admin Login

root

••••••••



breakfast
dosa
14-02-2023

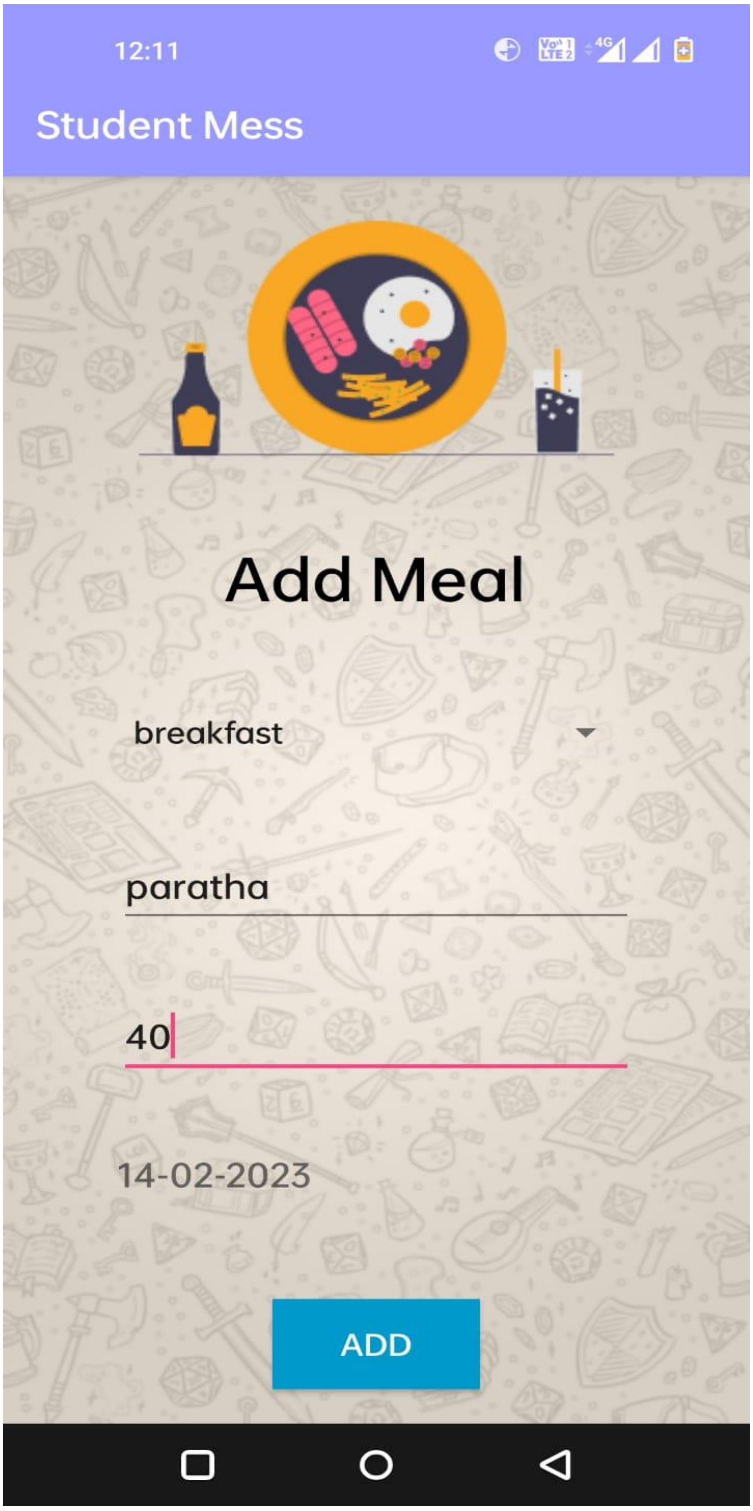
breakfast
Kachori
14-02-2023

breakfast
paratha
14-02-2023

lunch
paneer
15-02-2023

breakfast
samosa
15-02-2023

lunch
rajma
17-01-2023



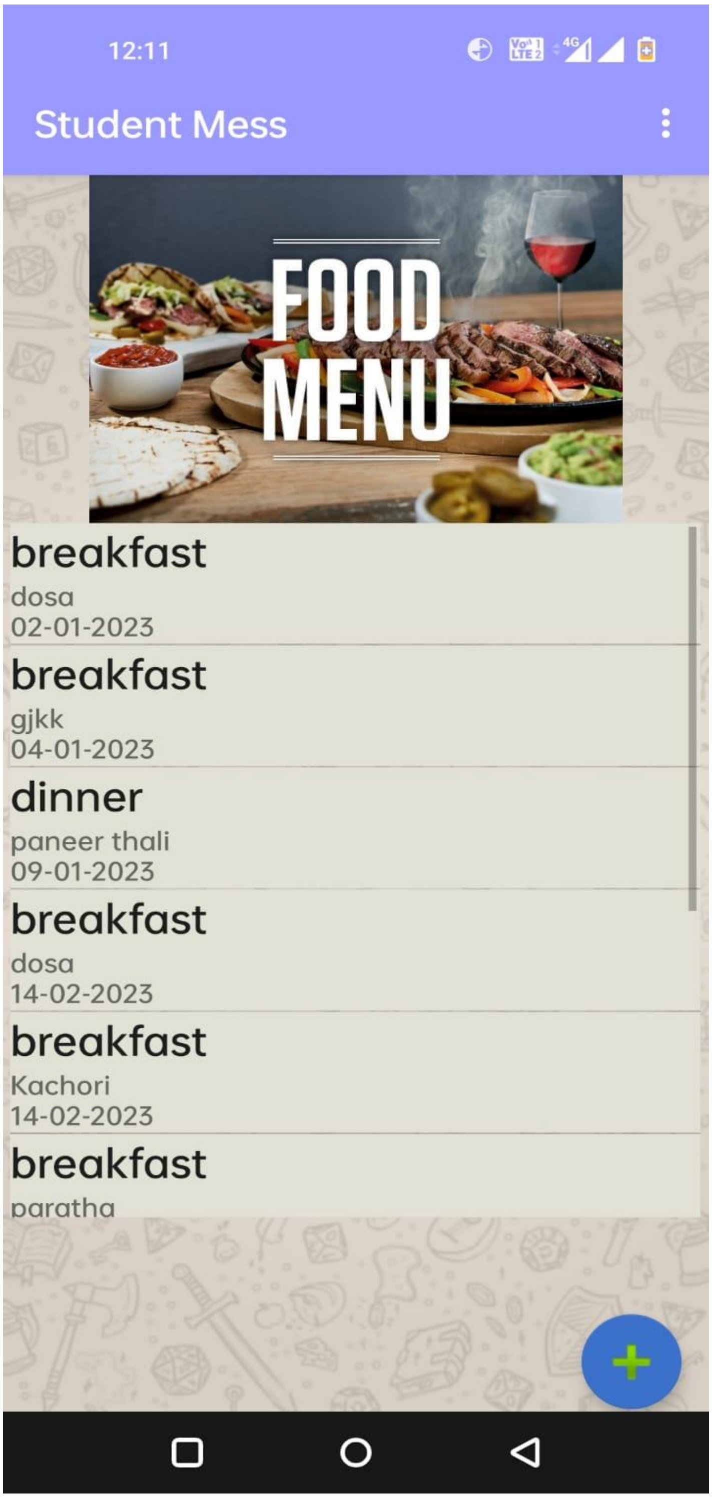
Add Meal

breakfast

paratha

40

14-02-2023



breakfast
dosa
02-01-2023

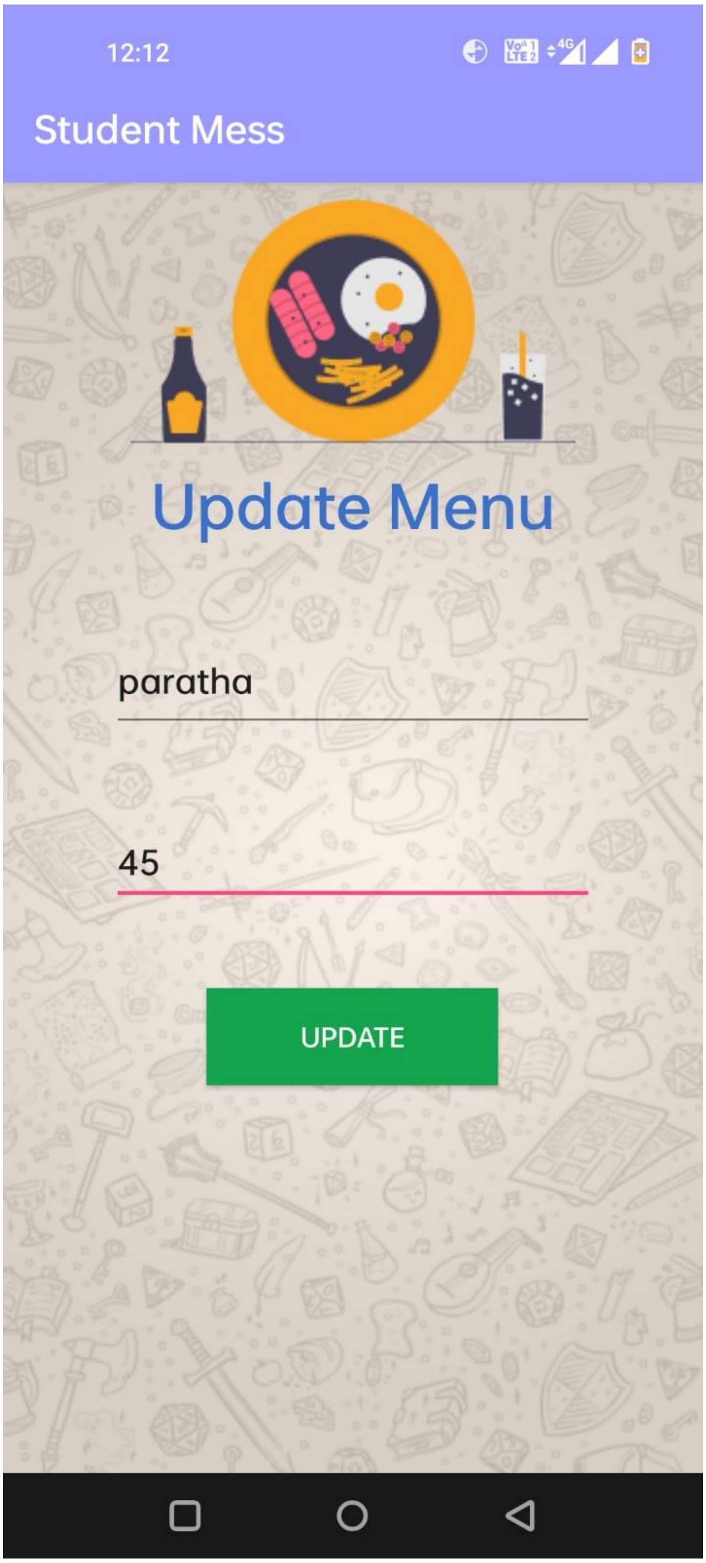
breakfast
gjkk
04-01-2023

dinner
paneer thali
09-01-2023

breakfast
dosa
14-02-2023

breakfast
Kachori
14-02-2023

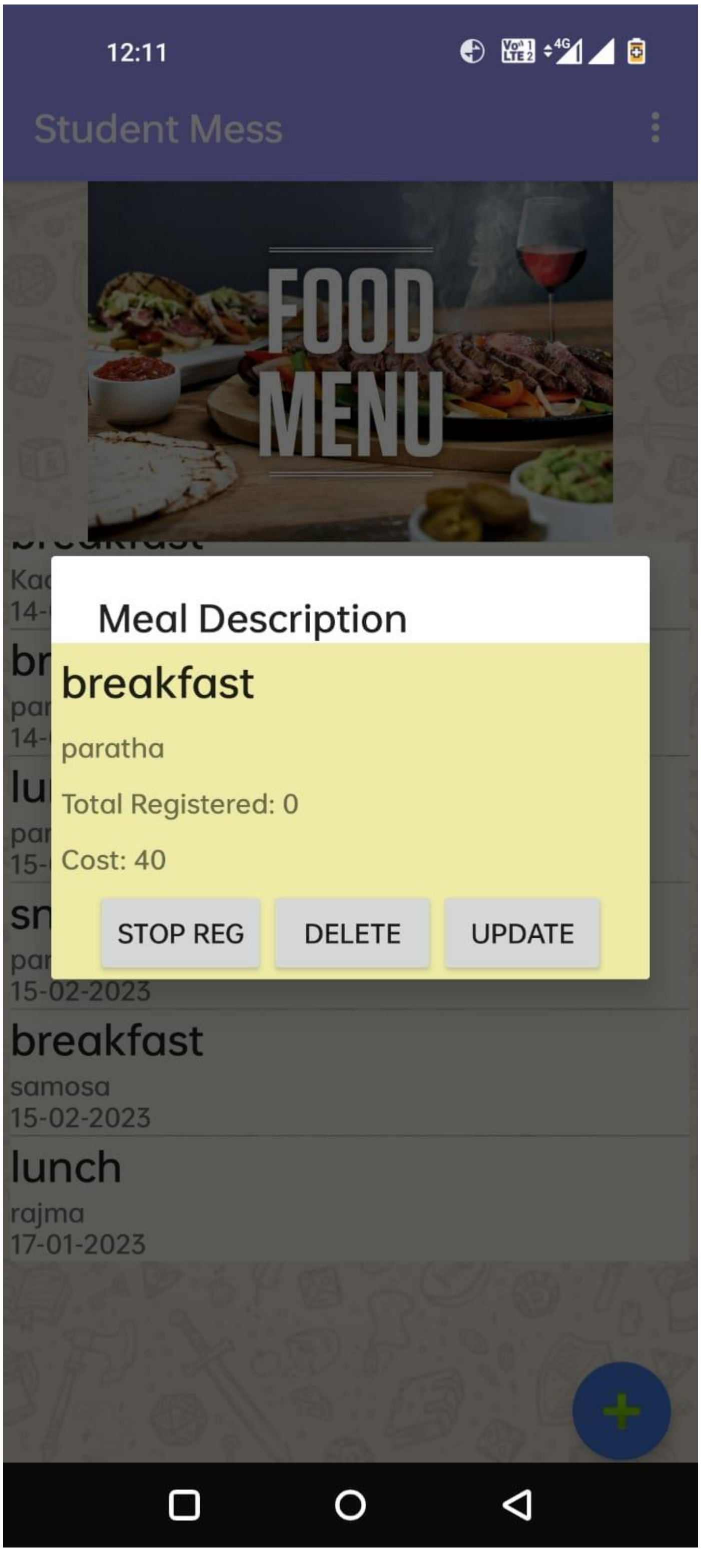
breakfast
paratha



Update Menu

paratha

45



Meal Description

breakfast

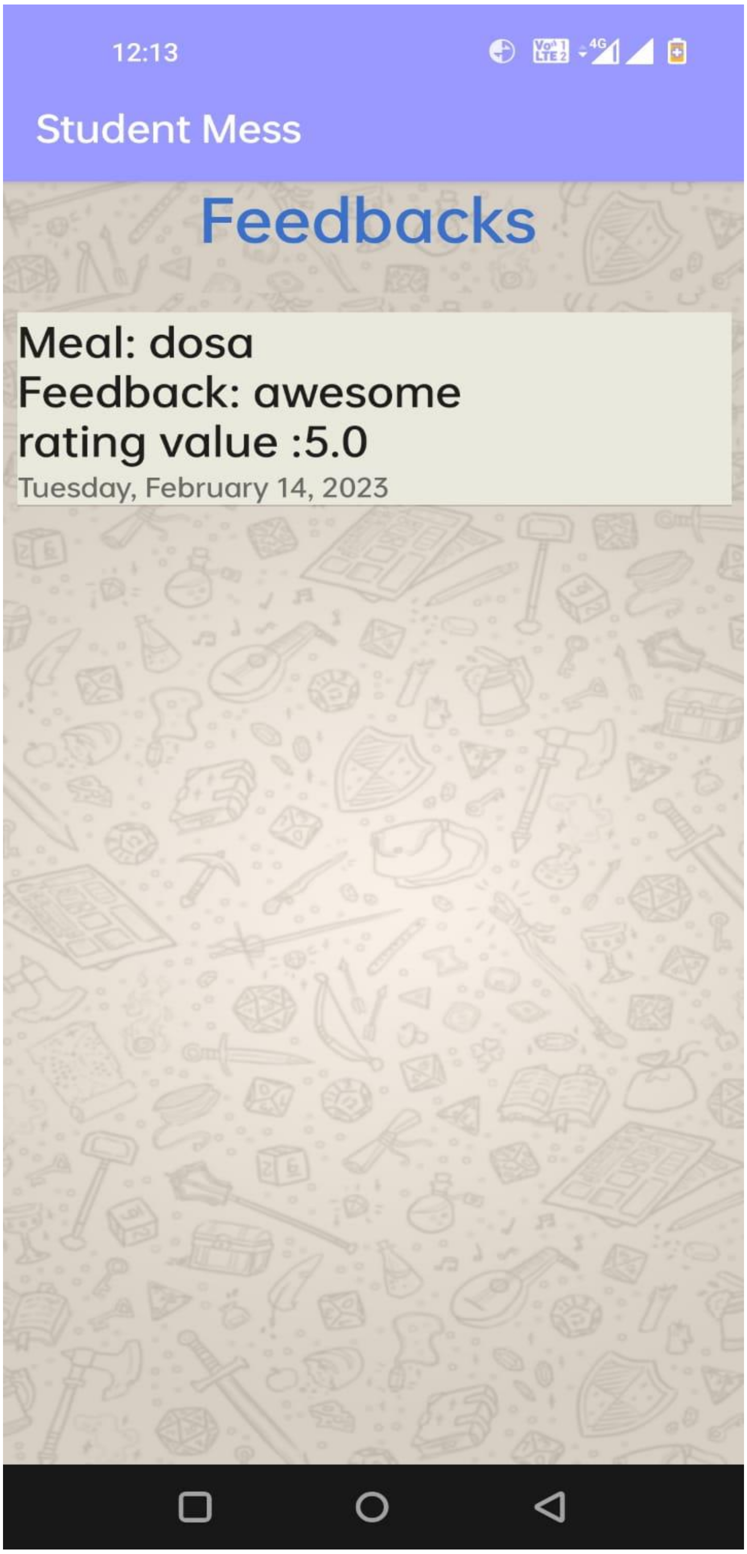
paratha

Total Registered: 0

Cost: 40

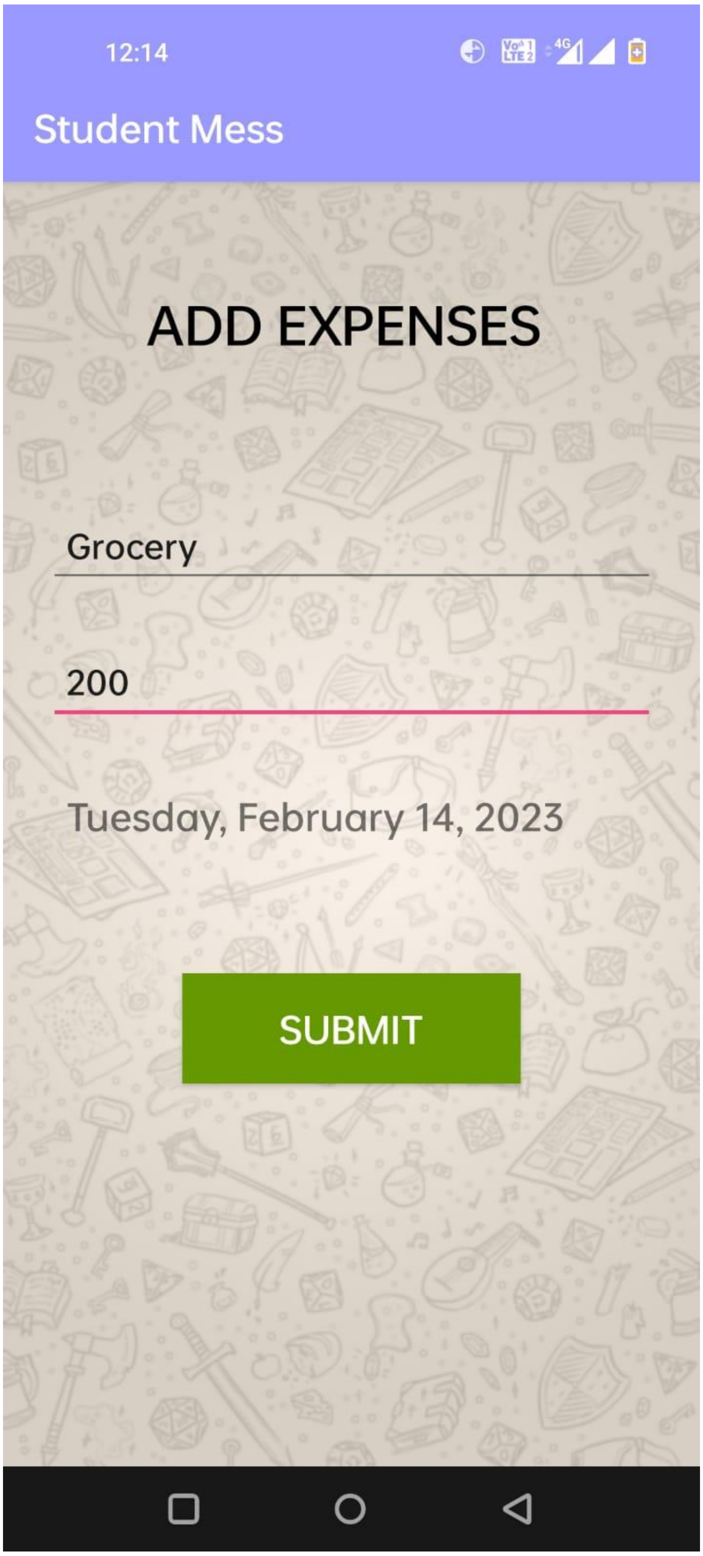
breakfast
samosa
15-02-2023

lunch
rajma
17-01-2023



Feedbacks

Meal: dosa
Feedback: awesome
rating value :5.0
Tuesday, February 14, 2023

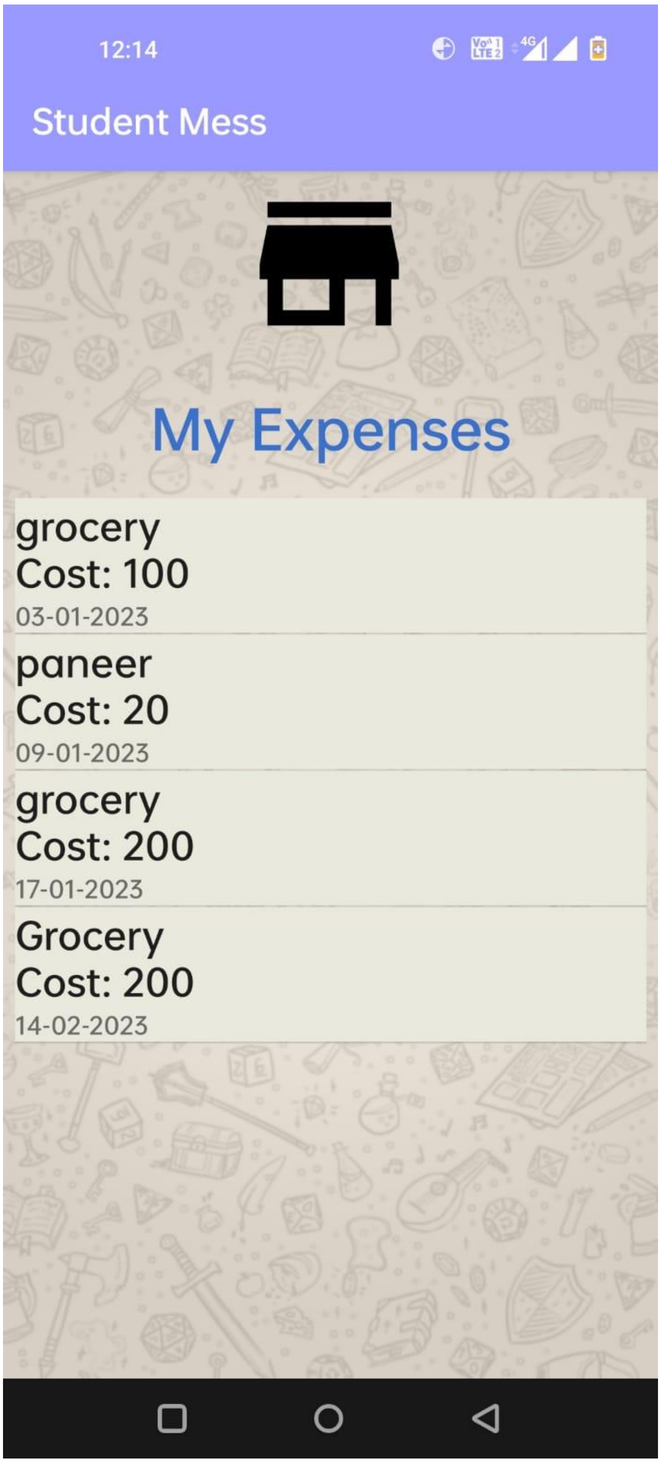


ADD EXPENSES

Grocery

200

Tuesday, February 14, 2023



Student Mess

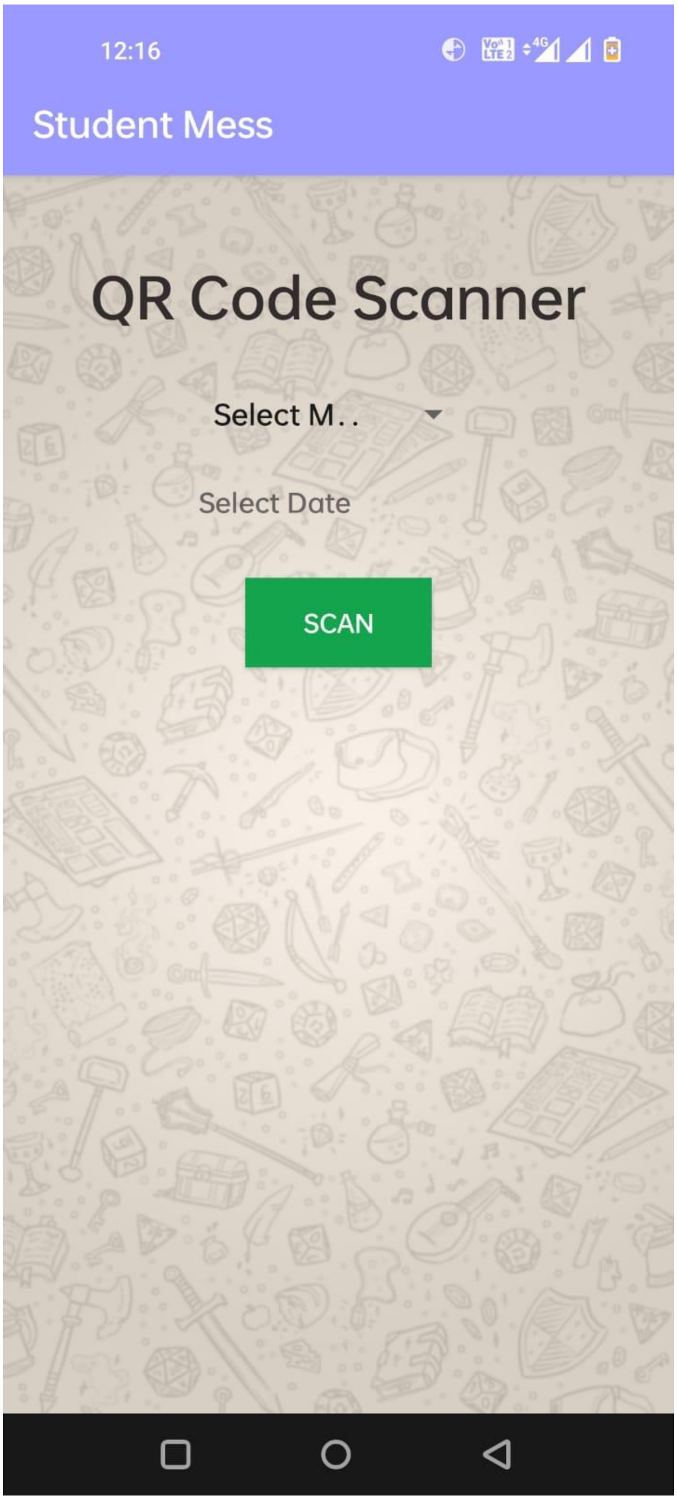
My Expenses

grocery
Cost: 100
03-01-2023

paneer
Cost: 20
09-01-2023

grocery
Cost: 200
17-01-2023

Grocery
Cost: 200
14-02-2023

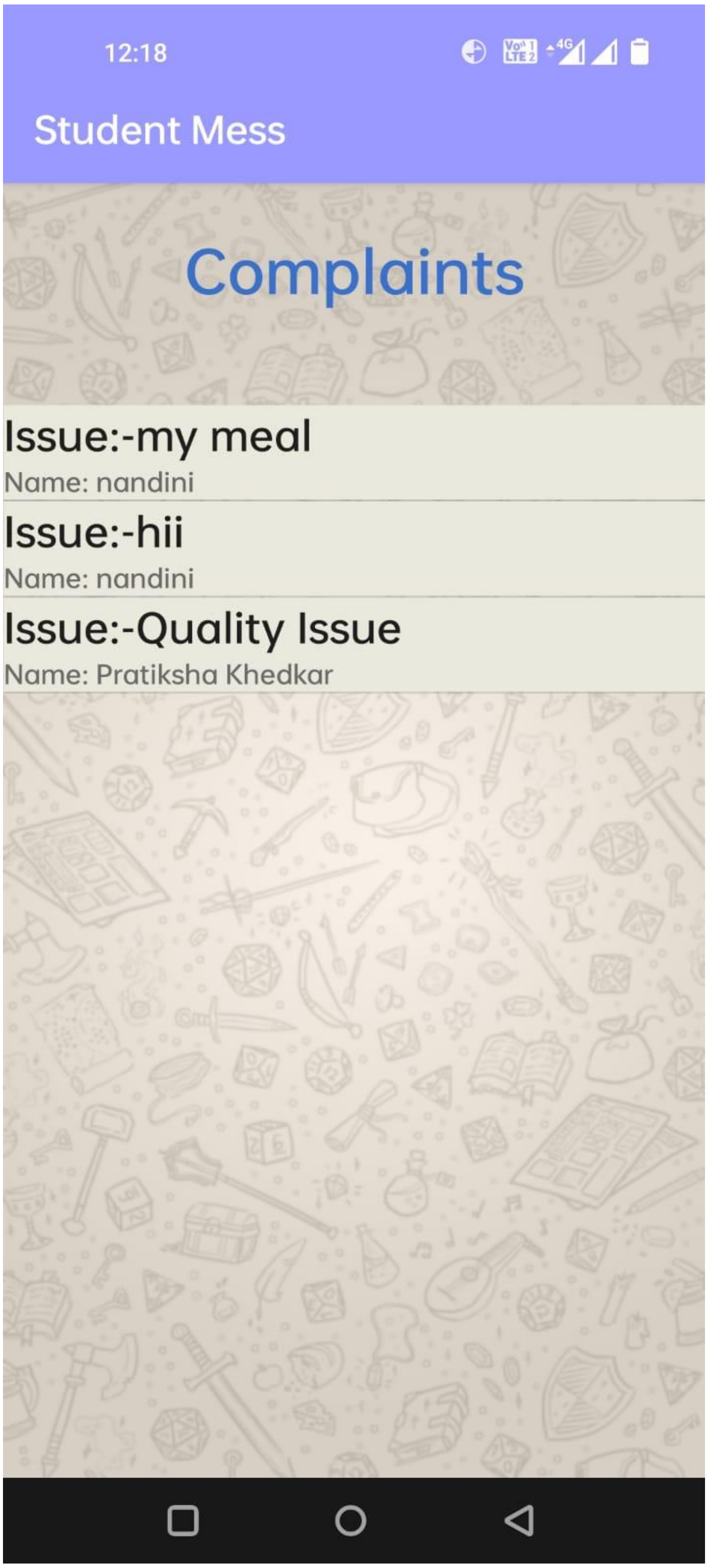


QR Code Scanner

Select M . .

Select Date

SCAN



Complaints

Issue:-my meal
Name: nandini

Issue:-hii
Name: nandini

Issue:-Quality Issue
Name: Pratiksha Khedkar

5. Test Cases

Test Unit: Add Meal

ID	Condition To be Tested	Test Data	Expected Output	Remarks
1.	Check if meal type is not selected	Type=”Breakfast”	Kindly select meal type	SUCCESSFUL
2.	Check if meal title field is empty	Title=”Idli”	Meal title should not be empty	SUCCESSFUL
3.	Check if cost field is empty	Cost=40	Cost should not be empty	SUCCESSFUL
4.	Check if cost fields contains other than digits	Cost=40	Cost fields should contain digits	SUCCESSFUL
5.	Check if date is not selected	Date=”28/02/2023”	Kindly select date	SUCCESSFUL

Test Unit: Add Expense

ID	Condition To be Tested	Test Data	Expected Output	Remarks
1.	Check if title field is empty	Title=”Grocery”	Expense title field should not be empty	SUCCESSFUL
2.	Check if amount field is empty	Amount=1000	Amount should not be empty	SUCCESSFUL
3.	Check if cost fields contains other than digits	Amount=1000	Amount should contain digits	SUCCESSFUL
4.	Check if expense date is not selected	Date=”28/02/2023”	Kindly select date	SUCCESSFUL

Test Unit: Feedback

ID	Condition To be Tested	Test Data	Expected Output	Remarks
1	If feedback title in the form is empty	Title=”quality”	Feedback title should not be empty	SUCCESSFUL
2	If feedback message in the form is empty	Message=”Good Food”	Feedback message should not be empty	SUCCESSFUL
3	If rating is not selected	Rating= 2.5	Kindly select rating	SUCCESSFUL

Test Unit: Complaint

ID	Condition To be Tested	Test Data	Expected Output	Remarks
1	If email field is empty	Email=”pratikshakhedkar4@gmail.com”	Email Id should not be empty	SUCCESSFUL
2	If email id does not contains proper format	Email=”pratikshakhedkar4@gmail.com”	Entered email id is not valid	SUCCESSFUL
3	If complaint message field is empty	Complaint=”Staff is rude”	Message field should not be empty	SUCCESSFUL

6. **Drawbacks And Limitations**

- Admin have to add every meal for everyday.
- Student cannot keep track of payment history

7. Future Enhancement :

- Instead of adding meal everytime we can provide multiselect dropdown for food item and admin can add meal with cost by selecting item.
- Integrate payment gateway to make payment and keep track of previous payments.

8. Conclusion :

- The project is based on the requirement specification of the user and the analysis of the existing system, with flexibility for future enhancement.
- The expanded functionality of today's software requires an appropriate approach towards software development.
- Mess management software is designed for the customers who regularly visit outside for their lunch or dinner.
- The numbers of mess are increasing day by day and they all try to provide the best facility to a customer and hence all thing are becoming digital therefore we provide a facility for the customer such as:
- Easy choice of food dishes from the menu.
- It will also help manage the mess in a better way and help in decreasing

9. Bibliography :

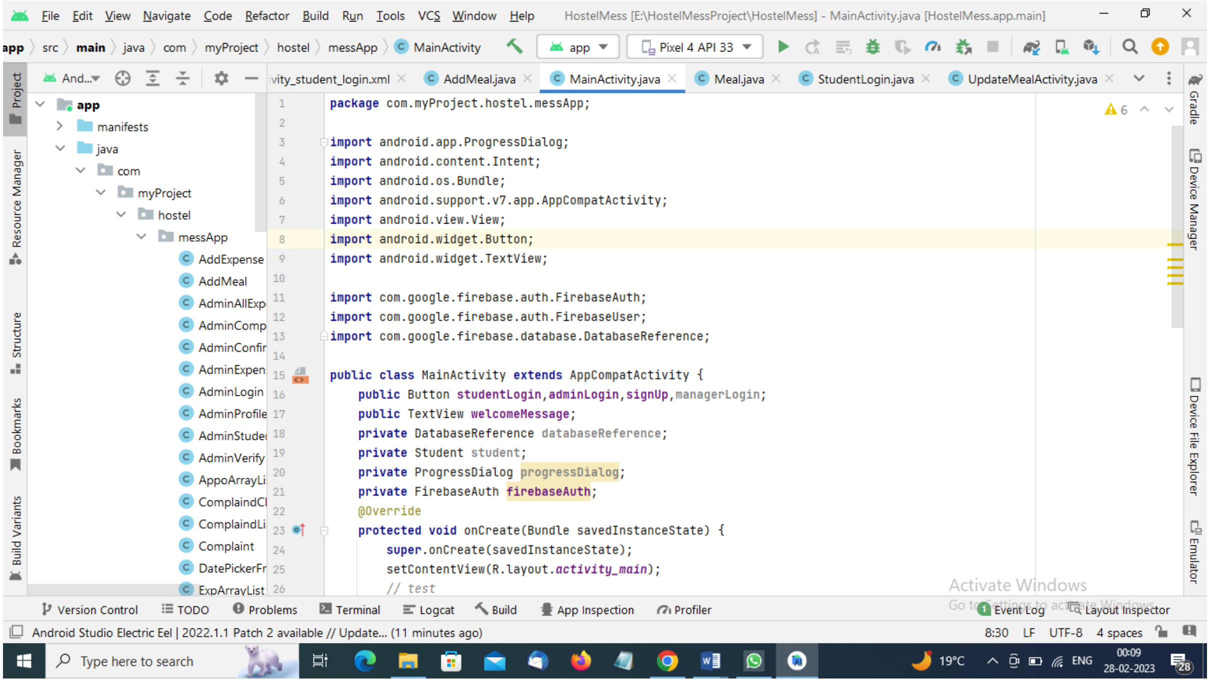
Reference books :

1. Headfirst Android Development
2. Android Development for Dummies
3. Android Introducing Googles Mobile Development Platform
4. Busy Coder's Guide to Android Development
5. Android Programming the Big Nerd

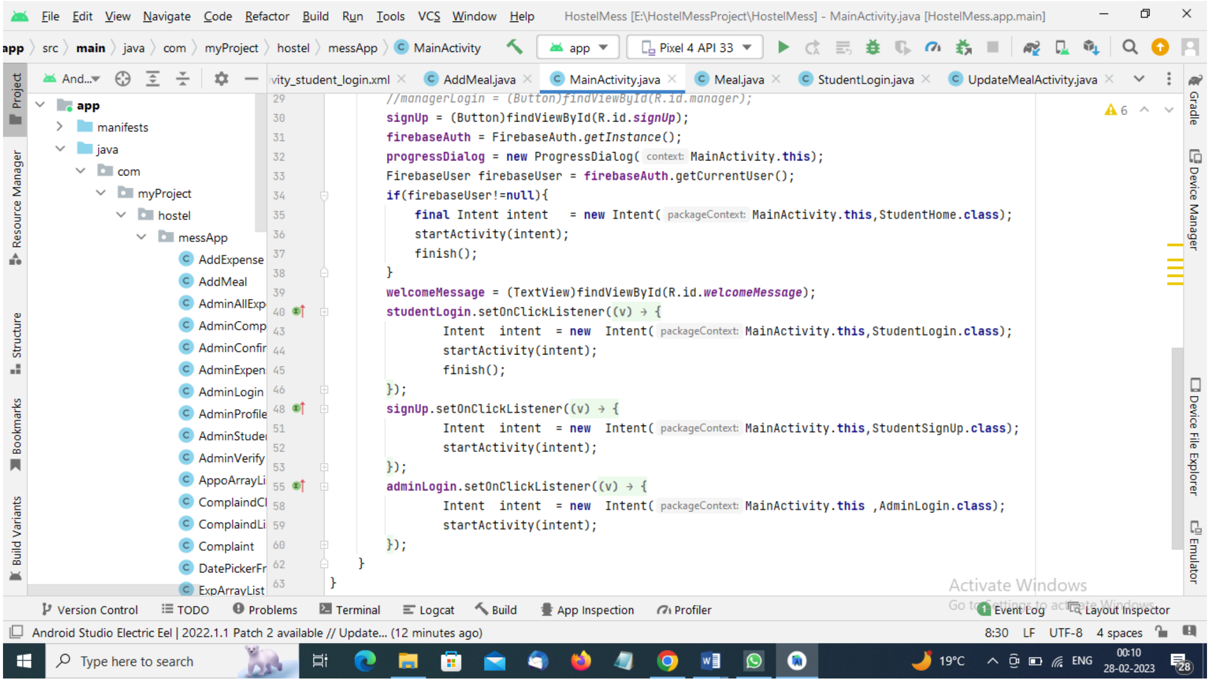
Web links:

- **www.developer.android.com**
- **www.javatpoint.com**
- **www.androidforums.com**

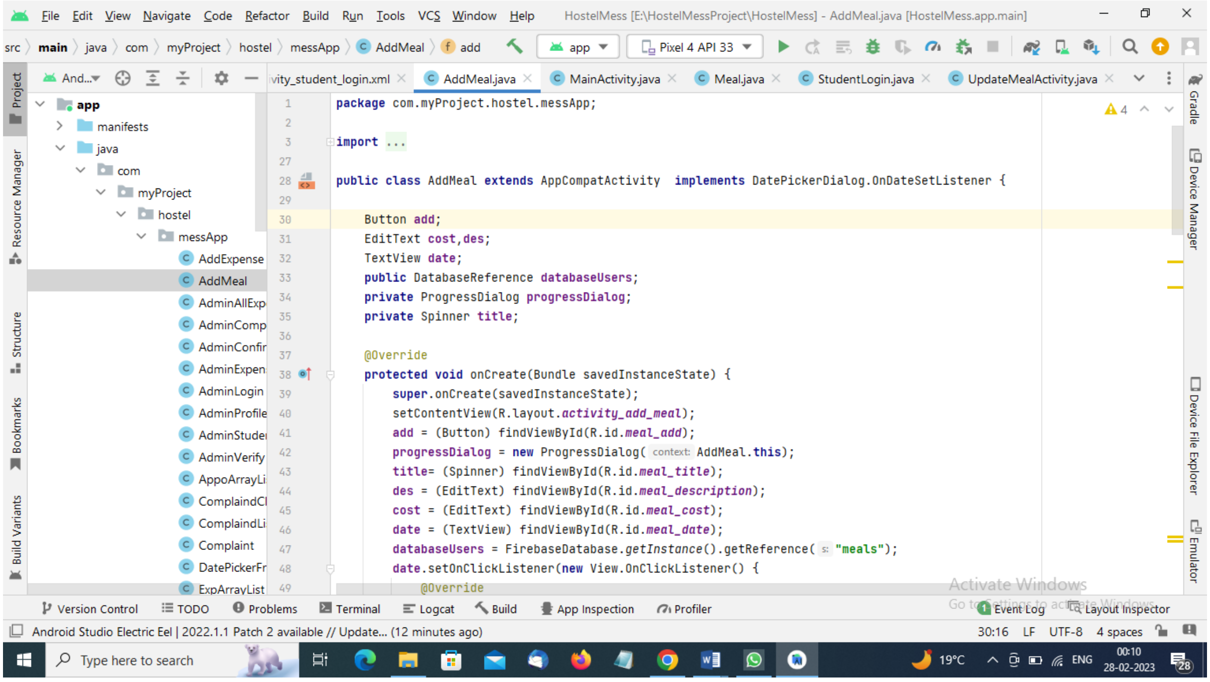
10. Sample Code :



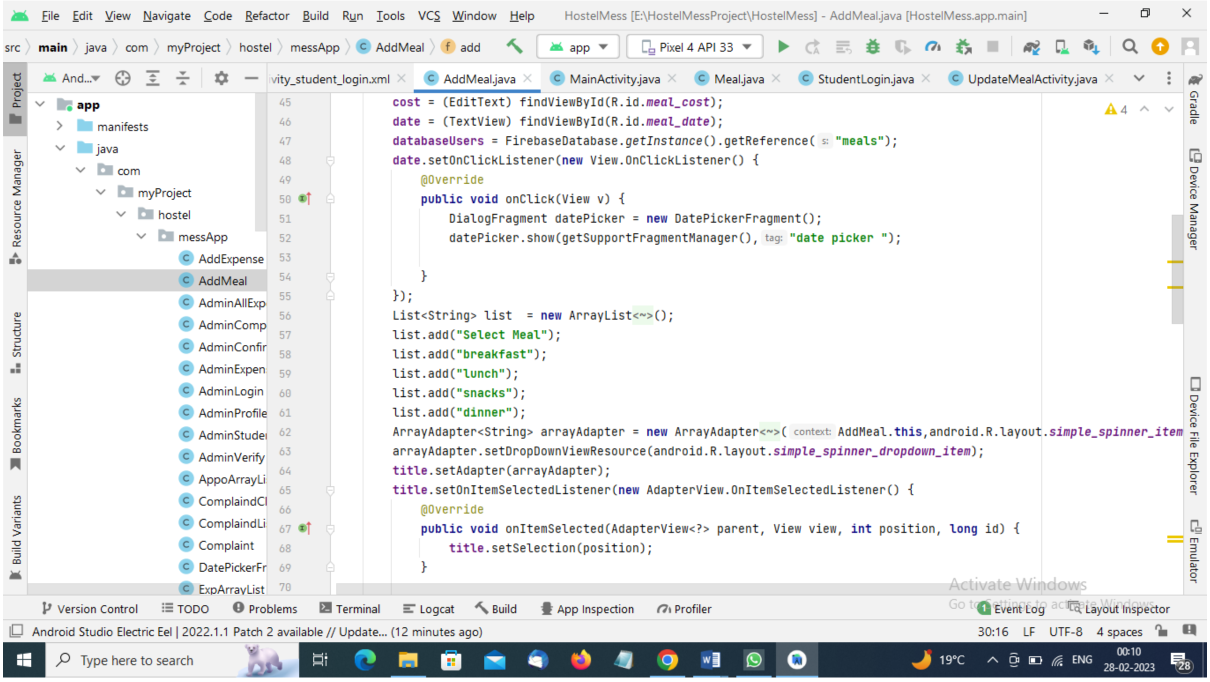
```java
package com.myProject.hostel.messApp;

import android.app.ProgressDialog;
import android.content.Intent;
import android.os.Bundle;
import android.support.v7.app.AppCompatActivity;
import android.view.View;
import android.widget.Button;
import android.widget.TextView;

import com.google.firebase.auth.FirebaseAuth;
import com.google.firebase.auth.FirebaseUser;
import com.google.firebase.database.DatabaseReference;

public class MainActivity extends AppCompatActivity {
    public Button studentLogin,adminLogin,signUp,managerLogin;
    public TextView welcomeMessage;
    private DatabaseReference databaseReference;
    private Student student;
    private ProgressDialog progressDialog;
    private FirebaseAuth firebaseAuth;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        // test
```



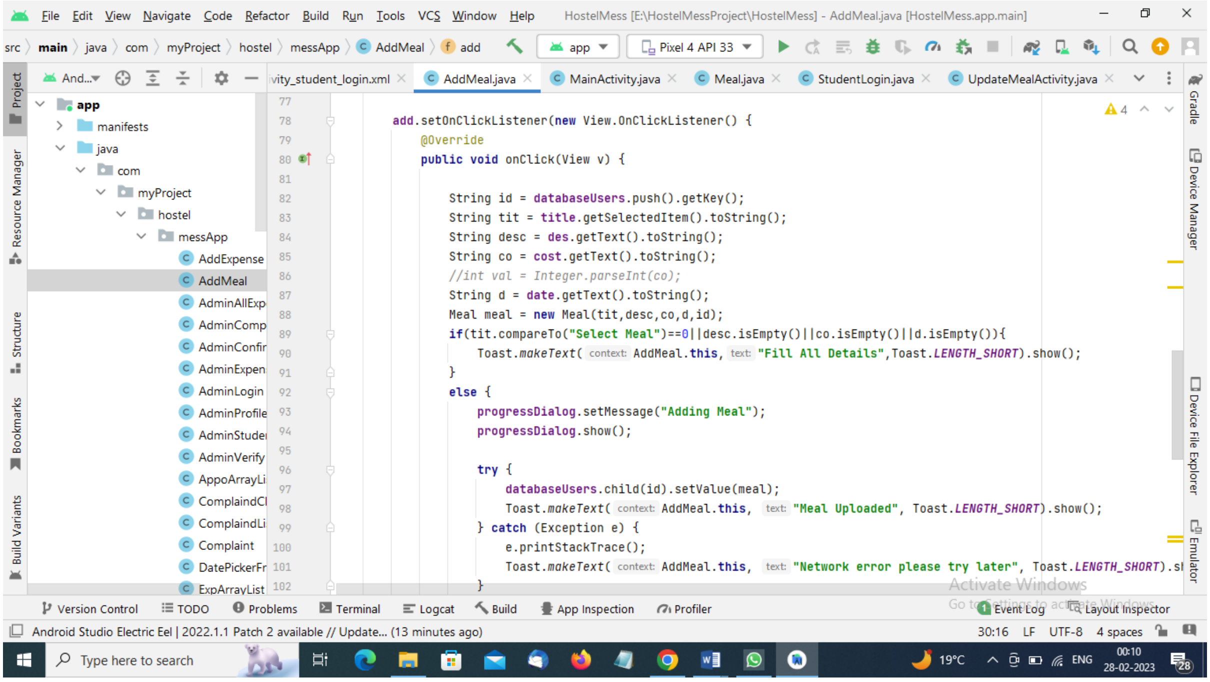
```java
        //managerLogin = (Button)findViewById(R.id.manager);
        signUp = (Button)findViewById(R.id.signUp);
        firebaseAuth = FirebaseAuth.getInstance();
        progressDialog = new ProgressDialog( context: MainActivity.this);
        FirebaseUser firebaseUser = firebaseAuth.getCurrentUser();
        if(firebaseUser!=null){
            final Intent intent   = new Intent( packageContext: MainActivity.this,StudentHome.class);
            startActivity(intent);
            finish();
        }
        welcomeMessage = (TextView)findViewById(R.id.welcomeMessage);
        studentLogin.setOnClickListener((v) → {
                Intent  intent  = new  Intent( packageContext: MainActivity.this,StudentLogin.class);
                startActivity(intent);
                finish();
        });
        signUp.setOnClickListener((v) → {
                Intent  intent  = new  Intent( packageContext: MainActivity.this,StudentSignUp.class);
                startActivity(intent);
        });
        adminLogin.setOnClickListener((v) → {
                Intent  intent  = new  Intent( packageContext: MainActivity.this ,AdminLogin.class);
                startActivity(intent);
        });
    }
}
```



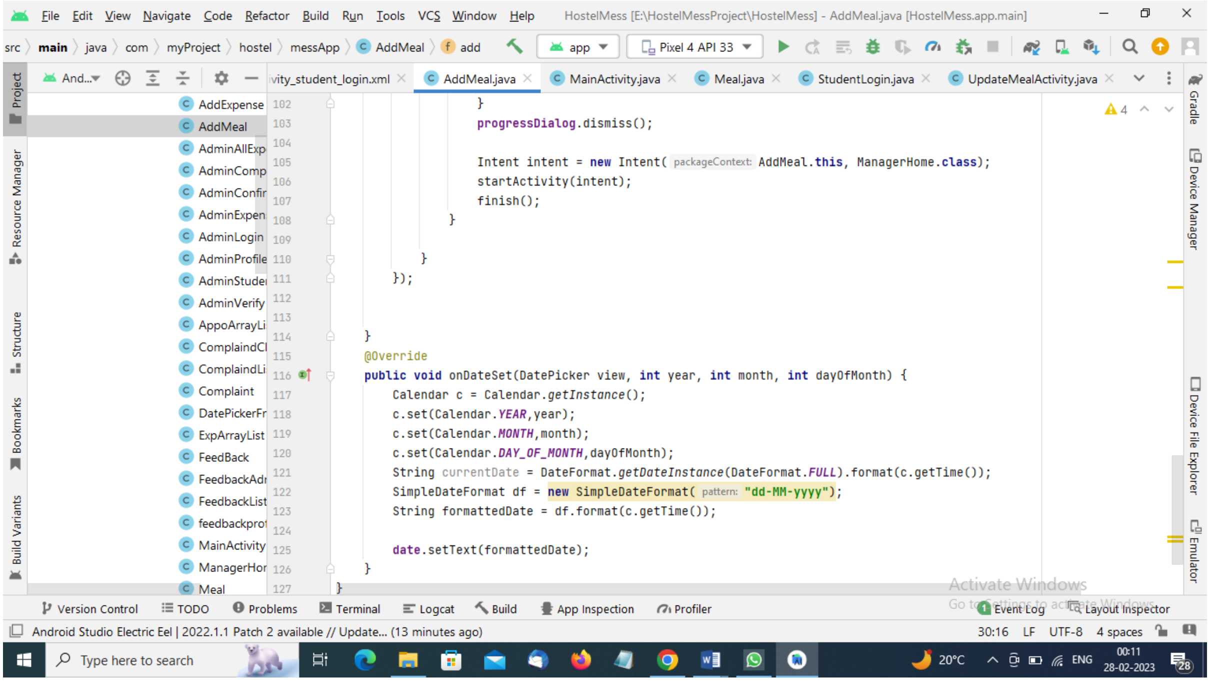
```java
package com.myProject.hostel.messApp;

import ...

public class AddMeal extends AppCompatActivity  implements DatePickerDialog.OnDateSetListener {

    Button add;
    EditText cost,des;
    TextView date;
    public DatabaseReference databaseUsers;
    private ProgressDialog progressDialog;
    private Spinner title;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_add_meal);
        add = (Button) findViewById(R.id.meal_add);
        progressDialog = new ProgressDialog( context: AddMeal.this);
        title= (Spinner) findViewById(R.id.meal_title);
        des = (EditText) findViewById(R.id.meal_description);
        cost = (EditText) findViewById(R.id.meal_cost);
        date = (TextView) findViewById(R.id.meal_date);
        databaseUsers = FirebaseDatabase.getInstance().getReference( s: "meals");
        date.setOnClickListener(new View.OnClickListener() {
            @Override
```


HostelMess [E:\HostelMessProject\HostelMess] - AddMeal.java [HostelMess.app.main]
src 〉 main 〉 java 〉 com 〉 myProject 〉 hostel 〉 messApp 〉 AddMeal 〉 add
app
Pixel 4 API 33
vity_student_login.xml
AddMeal.java
MainActivity.java
Meal.java
StudentLogin.java
UpdateMealActivity.java
Android Studio Electric Eel | 2022.1.1 Patch 2 available // Update... (12 minutes ago)
30:16 LF UTF-8 4 spaces
Type here to search
19°C ENG 00:10 28-02-2023

```java
        cost = (EditText) findViewById(R.id.meal_cost);
        date = (TextView) findViewById(R.id.meal_date);
        databaseUsers = FirebaseDatabase.getInstance().getReference( s: "meals");
        date.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                DialogFragment datePicker = new DatePickerFragment();
                datePicker.show(getSupportFragmentManager(), tag: "date picker ");

            }
        });
        List<String> list  = new ArrayList<~>();
        list.add("Select Meal");
        list.add("breakfast");
        list.add("lunch");
        list.add("snacks");
        list.add("dinner");
        ArrayAdapter<String> arrayAdapter = new ArrayAdapter<~>( context: AddMeal.this,android.R.layout.simple_spinner_item
        arrayAdapter.setDropDownViewResource(android.R.layout.simple_spinner_dropdown_item);
        title.setAdapter(arrayAdapter);
        title.setOnItemSelectedListener(new AdapterView.OnItemSelectedListener() {
            @Override
            public void onItemSelected(AdapterView<?> parent, View view, int position, long id) {
                title.setSelection(position);
            }
```



```java

        add.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {

                String id = databaseUsers.push().getKey();
                String tit = title.getSelectedItem().toString();
                String desc = des.getText().toString();
                String co = cost.getText().toString();
                //int val = Integer.parseInt(co);
                String d = date.getText().toString();
                Meal meal = new Meal(tit,desc,co,d,id);
                if(tit.compareTo("Select Meal")==0||desc.isEmpty()||co.isEmpty()||d.isEmpty()){
                    Toast.makeText( context: AddMeal.this, text: "Fill All Details",Toast.LENGTH_SHORT).show();
                }
                else {
                    progressDialog.setMessage("Adding Meal");
                    progressDialog.show();

                    try {
                        databaseUsers.child(id).setValue(meal);
                        Toast.makeText( context: AddMeal.this, text: "Meal Uploaded", Toast.LENGTH_SHORT).show();
                    } catch (Exception e) {
                        e.printStackTrace();
                        Toast.makeText( context: AddMeal.this, text: "Network error please try later", Toast.LENGTH_SHORT).sl
                    }
```



```java
                    }
                    progressDialog.dismiss();

                    Intent intent = new Intent( packageContext: AddMeal.this, ManagerHome.class);
                    startActivity(intent);
                    finish();
                }

            }
        });


    }
    @Override
    public void onDateSet(DatePicker view, int year, int month, int dayOfMonth) {
        Calendar c = Calendar.getInstance();
        c.set(Calendar.YEAR,year);
        c.set(Calendar.MONTH,month);
        c.set(Calendar.DAY_OF_MONTH,dayOfMonth);
        String currentDate = DateFormat.getDateInstance(DateFormat.FULL).format(c.getTime());
        SimpleDateFormat df = new SimpleDateFormat( pattern: "dd-MM-yyyy");
        String formattedDate = df.format(c.getTime());

        date.setText(formattedDate);
    }
}
```