A PROJECT II REPORT

ON

"E-RATION SYSTEM"

Submitted to

UNIVERSITY OF MUMBAI

In Partial Fulfilment of the Requirement for the Award of

**BACHELOR'S DEGREE IN
COMPUTER ENGINEERING**

BY

Shaikh Mumtaz Bano Abdul Aziz Soni	17CO13
Dar Waseem Ghulam Hassan Dar Misra Bano	17CO54
Dev Shahid Ghulam Ahmad Dev Zoona bano	17CO43
Patel Zulekha Ibrahimsha Nafisa	17CO06

UNDER THE GUIDANCE OF

PROF. MUKHTAR QURESHI

DEPARTMENT OF COMPUTER ENGINEERING

Anjuman-I-Islam's Kalsekar Technical Campus

SCHOOL OF ENGINEERING & TECHNOLOGY

Plot No. 2 3, Sector - 16, Near Thana Naka,

Khandagaon, New Panvel - 410206

2020-2021

AFFILIATED TO

UNIVERSITY OF MUMBAI

CERTIFICATE

Department of Computer Engineering,

School of Engineering and Technology,
Anjuman-I-Islam's Kalsekar Technical Campus
Khanda Gaon,New Panvel, Navi Mumbai. 410206

This is to certify that the project entitled ***E-Ration Shop*** is a bonafide work of **Shaikh Mumtaz(Roll No.17CO13),Waseem ahmad dar(17CO54),Shahid ahmad dev(17CO43) and Patel Zulekha Ibrahimsha(Roll No.17CO06)** submitted to the University of Mumbai in partial fulfillment of the requirement for the award of the degree of **Bachelor of Engineering** in **Department of Computer Engineering**.

Name	**Prof. Kalpana R. Bodke**
Supervisor	Project Coordinator
Prof. Tabrez Khan	**Dr. Abdul Razak Honnutagi**
Head of Department	Director

Project Approval for Bachelor of Engineering

This project entitled ***Project Title*** by ***Student Name*** is approved for the degree of ***Bachelor of Engineering in Department of Computer Engineering.***

Examiners

1. ....

2. ....

Supervisors

1. ....

2. ....

Chairman

....

Declaration

I declare that this written submission represents my ideas in my own words and where others ideas or words have been included, I have adequately cited and referenced the original sources. I also declare that I have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in my submission. I understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

Shaikh Mumtaz Bano
17CO13

Waseem Ahmad Dar
17CO54

Shahid Ahmad Dev
17CO43

Patel Zulekha Ibrahimsha
17CO06

Abstract

Tittle:E-Ration Shop

The Indian government distributes food for the poor people who are valid ration card holders and this manual system is facing many problems such as, Cardholders don't know the opening and closing of Ration shop and there is no mechanism to find errors such as duplicate, ineligible beneficiaries, Human intervention in updating transactions, Maintenance of thousands of all records in the form of hard copy is time-taking and difficult. No complaint forum. To Overcome these Problems, The "e-Ration shop" android application avoids above-mentioned problems. Each and every RationShop Geolocation will be located on the map and the information regarding RationShop in respective to the ration card and region is available in the Application and Shop opening and closing details will be available and cardholders details are available and Stock provided based on The estimate consumption of FPS grains in India is only 1 kg per person per month and changes depending on the Commodity. The PDS has been criticised for its inefficiences to serve the poorer sections of the population effectively

Shaikh Mumtaz
Roll No: 17CO13
B.E. (Computer Engineering)
University of Mumbai.

Waseem Ahmad Dar
Roll No: 17CO54
B.E. (Computer Engineering)
University of Mumbai.

Shahid Ahmad Dev
Roll No: 17CO43
B.E. (Computer Engineering)

University of Mumbai.

Patel Zulekha Ibrahimsha
Roll No: 17CO06
B.E. (Computer Engineering)
University of Mumbai.

ACKNOWLEDGMENT

I would like to take the opportunity to express my sincere thanks to my guide **Prof. Mukhtar Ansari**, Assistant Professor, Department of Computer Engineering, AIKTC, School of Engineering, Panvel for his invaluable support and guidance throughout my project research work. Without his kind guidance & support this was not possible.

I am grateful to him/her for his timely feedback which helped me track and schedule the process effectively. His/her time, ideas and encouragement that he gave is help me to complete my project efficiently.

I would also like to thank **Dr. Abdul Razak Honnutagi**, AIKTC, Panvel, for his encouragement and for providing an outstanding academic environment, also for providing the adequate facilities.

I am thankful to **Prof. Tabrez Khan**, HOD, Department of Computer Engineering, AIKTC, School of Engineering, Panvel and all my B.E. teachers for providing advice and valuable guidance.

I also extend my sincere thanks to all the faculty members and the non-teaching staff and friends for their cooperation.

Last but not the least, I am thankful to all my family members whose constant support and encouragement in every aspect helped me to complete my project.

Shaikh Mumtaz Bano Abdul Aziz
Roll No. 17CO13
(Department of Computer Engineering)
University of Mumbai.

Dar Waseem Ahmad Misra Bano

Roll No.17CO54

(Department of Computer Engineering)

University of Mumbai.

Dev Shahid Ahmad Zoona

Roll No.17CO43

(Department of Computer Engineering)

University of Mumbai.

Patel Zulekha Ibrahimsha

Roll No.17CO06

(Department of Computer Engineering)

University of Mumbai.

Contents

List of Figures

List of Tables

Keywords And Glossary

Keywords :

E Ration Shop, Complaints forum, Authentication System, Geolocation, Subsidized food.

Glossary :

A:

Application - a software program that runs on your computer.

Andriod Studio - Android Studio is the official Integrated Development Environment (IDE) for Android app development

Analysis - detailed examination of the elements or structure of something

Affirmative - supportive, hopeful, or encouraging.

D:

Database - a structured set of data held in a computer, especially one that is accessible in various ways.

Depicting - to represent or show something

F:

Firebase- Firebase is your server, your API and your datastore, all written so generically that you can modify it to suit most needs.

J:

java-Java is a high-level programming language developed by Sun Microsystems. It was originally designed for developing programs for set-top boxes and handheld devices, but later became a popular choice for creating web applications. L: Location - a particular place or position

N:

Notification - the action of notifying someone or something.

P:

Purpose-the reason for which something is done or created or for which something exists.

R:

Review - a formal assessment of something with the intention of instituting change if necessary.

S:

Strategy-a plan of action designed to achieve along-term orover all aim.

V:

Vocabulary - the body of words used in a particular language.

X:

XML-stands for Extensible Markup Language. It is a text-based markup language derived from Standard Generalized Markup Language (SGML).

Chapter 1

Introduction

1.1 Problem Statement and Objective

The PDS has been criticized for its urban bias and its failure to serve the poorer sections of the population effectively.Also many retail shopkeepers have large number of bogus cards to sell food grains in the open market. E-Ration Shop aims to ensure that only the entitled lot receives the subsidized food material and all other routine chores of inventory planning and reporting are done with least human intervention to ensure smooth operation of the PDS scheme , people can get all the benefits of the scheme easily and retailers can also some profit other than their salary.

1.2 Scope

• Every shop's geolocation will be located on the map.

• All the information regarding shop and shopkeeper respective to the ration card and region.

• Shop opening and closing details will be available .

• The Stock availability will be notified on regular basis. Product delivery feature.

- Easy payment system by PayTM , COD etc.

- Authentication system by aadhaar and OTP.
- The site will be available in HINDI and ENGLISH language both.

1.3 Technical Details

1.3.1 JAVA :

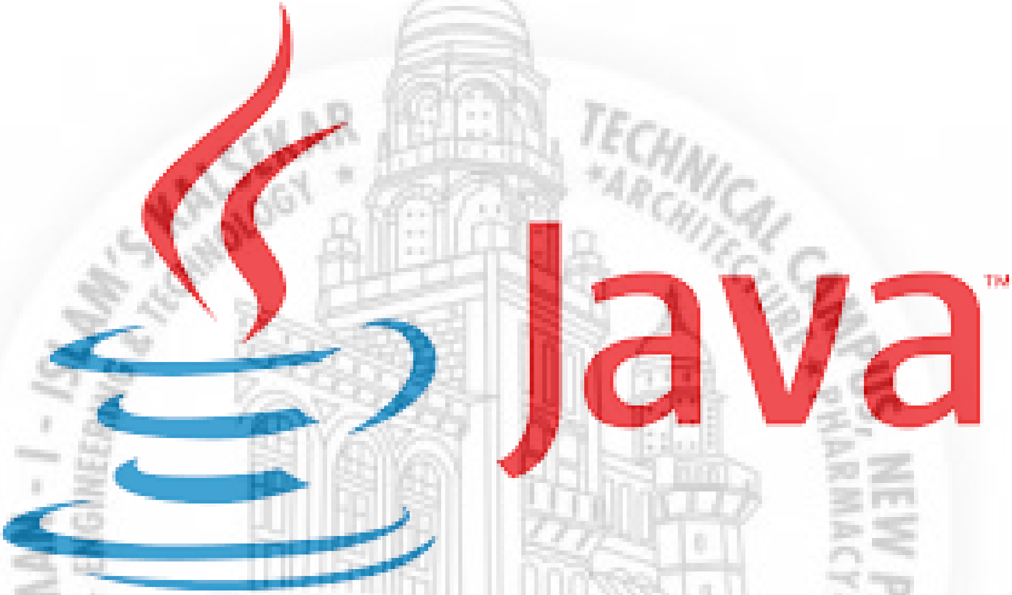


Figure 1.1: Java Logo

JAVA is a programming language which is used in Android App Development. It is class based and object oriented programming whose syntax is influenced by C++. The primary goals of JAVA is to be simple, object-oriented, robust, secure and high level.
JAVA application runs on JVM (JAVA Virtual Machine) but Android has it's own virtual machine called Dalvik Virtual Machine (DVM) optimized for mobile devices.
Java is a high-level programming language developed by Sun Microsystems. It was originally designed for developing programs for set-top boxes and handheld devices, but later became a popular choice for creating web applications.
Android applications are developed using the Java language. As of now, that's really your only option for native applications. Java is a very popular programming language developed by Sun Microsystems (now owned by Oracle). Developed long after C and C++, Java incorporates many of the powerful features of those powerful languages while addressing some of their drawbacks. Still, programming languages are only as powerful as their libraries. These

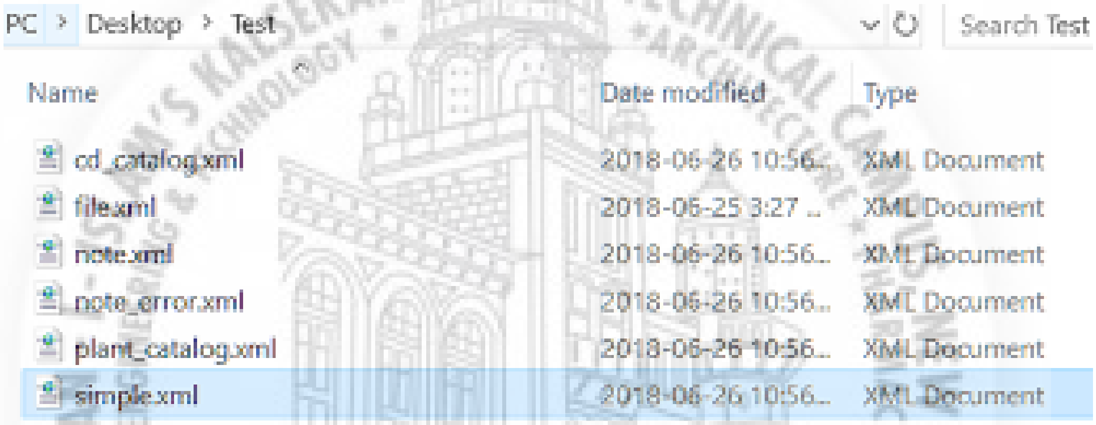
libraries exist to help developers build applications.

Some of the Java's important core features are:

- It's easy to learn and understand
- It's designed to be platform-independent and secure, using virtual machines
- It's object-oriented

Android relies heavily on these Java fundamentals. The Android SDK includes many standard Java libraries (data structure libraries, math libraries, graphics libraries, networking libraries and everything else you could want) as well as special Android libraries that will help you develop awesome Android applications.

1.3.2 XML:



Name	Date modified	Type
cd_catalog.xml	2018-06-26 10:56...	XML Document
file.xml	2018-06-25 3:27 ...	XML Document
note.xml	2018-06-26 10:56...	XML Document
note_error.xml	2018-06-26 10:56...	XML Document
plant_catalog.xml	2018-06-26 10:56...	XML Document
simple.xml	2018-06-26 10:56...	XML Document

Figure 1.2: Xml in Android

XML stands for Extensible Markup Language. It is a text-based markup language derived from Standard Generalized Markup Language (SGML). XML tags identify the data and are used to store and organize the data, rather than specifying how to display it like HTML tags, which are used to display the data. XML is not going to replace HTML in the near future, but it introduces new possibilities by adopting many successful features of HTML.

Different XML Files Used in Android:

In Android there are several xml files used for several different purposes. Below we define each and every one.

1. Layout XML Files:

Layout xml files are used to define the actual UI(User interface) of our application. It holds all the elements(views) or the tools that we want to use in our application. Like the TextView's, Button's and other UI elements.

Location in Android Studio: You will find out this file inside the res folder and inside it there is another folder named layout where you will get all the layout files for their respective activities or fragments.

2. Manifest xml File(Mainfest.xml):

This xml is used to define all the components of our application. It includes the names of our application packages, our Activities, receivers, services and the permissions that our application needs. For Example – Suppose we need to use internet in our app then we need to define Internet permission in this file.

Location in Android Studio: It is located inside app > manifests folder

3. Strings xml File(strings.xml):

This xml file is used to replace the Hard-coded strings with a single string. We define all the strings in this xml file and then access them in our app(Activity or in Layout XML files) from this file. This file enhance the reusability of the code.

4. Styles xml File(styles.xml):

This xml is used to define different styles and looks for the UI(User Interface) of application. We define our custom themes and styles in this file.

5. Drawable xml Files:

These are those xml files that are used to provide various graphics to the elements or views of application. When we need to create a custom UI we use drawable xml files. Suppose if we need to define a gradient color in the background of Button or any custom shape for a view then we create a Drawable xml file and set it in the background of View.

6. Color xml File (colors.xml):

This file is used to define the color codes that we used in our app. We simply define the color's in this file and used them in our app from this file.

7. Dimension xml File(dimens.xml):

This xml file is used to define the dimensions of the View's. Suppose we need a Button with 50dp(density pixel) height then we define the value 50dp in dimens.xml file and then use it in our app from this file.

1.3.3 FIREBASE :

Firebase is a Backend-as-a-Service — BaaS — that started as a YC11 startup and grew up into a next-generation app-development platform on Google Cloud Platform.

Figure 1.3: Firebase in Android

There are various sevices offered online such as storage, online processing, realtime database, authorisation of user etc. Google developed a platform called Firebase that provide all these online services. It also gives a daily analysis of usage of these services along with the details of user using it.

To simplify, it can be said that Firebase is a mobile and web application development platform. It provides services that a web application or mobile application might require. Anyone can easily include firebase to there application and it will make their online work way easier than it was used to be.

1.3.4 ANDROID :

Android is a mobile operating system (OS) based on the Linux kernel and currently developed by Google. With a user interface based on direct manipulation, Android is designed primarily for touch screen mobile devices such as Smartphone and tablet computers, with specialized user interfaces for televisions (Android TV), cars (Android Auto), and wrist watches (Android Wear).

Figure 1.4: Android System

The OS uses touch inputs that loosely correspond to real-world actions, like swiping, tapping, pinching, and reverse pinching to manipulate on-screen objects, and a virtual keyboard. Despite being primarily designed for touch screen input, it also has been used in game consoles, digital cameras, regular PCs (e.g. the HP Slate 21) and other electronics.

Android is popular with technology companies which require a ready-made, low-cost and customizable operating system for high-tech devices.
Android's open nature has encouraged a large community of developers and enthusiasts to use the open-source code as a foundation for community-driven projects, which add new features for advanced users or bring Android to devices which were officially released running other operating systems. The operating system's success has made it a target for patent litigation as part of the so-called "Smartphone wars" between technology companies.

1.3.5 ANDROID STUDIO :

Figure 1.5: Android studio application

Android Studio is the official Integrated Development Environment (IDE) for Android app development, based on IntelliJ IDEA . On top of IntelliJ's powerful code editor and developer tools, Android Studio offers even more features that enhance your productivity when building Android apps, such as:

- A flexible Gradle-based build system
- A fast and feature-rich emulator
- A unified environment where you can develop for all Android devices Apply Changes to push code and resource changes to your running app without restarting your app

Code templates and GitHub integration to help you build common app features and import sample code

1.4 Innovativeness Usefulness

- Will have predefined list of dependents of the ration card holder.
- Will display the current rates.
- Will have generated automatic billing etc.

Chapter 2

Literature Review

2.1 An Automated Approach to Public Distribution System Using Internet of Things

The process starts with the user scanning their smart card at the reader, followed by the biometric authentication.This authentication is too costly. Upon successful authentication, the user is prompted to enter the desired quantity of food grains via a touch screen interface. This touch screen interface comprises of a TFT LCD Capacitive Touch display, this gives the user a sensitive and easy to use platform to enter the details. As family is having limited quota of food grains allotted to them, this step checks for the validity of the data entered by the user. After successful operation, the user is prompted to complete the payment process.

2.1.1 Weaknesses

This system is too costly as it implements biometric authentication and vainding machine.

2.1.2 How to Overcome

Here the cost is less because we are not using hardware devices.

2.2 SMART RATIONING SYSTEM

Ration card system is used for the distribution of essential commodities to India’s below poverty line population. Whereas use of a ration card to obtain the various goods (sugar, rice, oil, kerosene, etc) from the ration shops. . The present ration distribution framework has lot many irregularities such as Ration delivered to open market, ration is not available, ration provided is not of good quality or not in sufficient quantity etc. Because of lack of awareness toward ration allocation there is a long wait in time, people have to stand in Q for longer duration just to fulfill their basic needs. The proposed framework replaces these irregularities and provides efficient automated rationing system. The proposed automatic ration shop for public distribution framework is based on Smart Card innovation that replaces ordinary ration cards with Aadhar cards that provides unique barcode and QR code. Smart card based automatic ration distribution system involves Real time database updating system that consist of information of goods allocated or pending in one’s account including details about dispensing of goods using particular Aadhar card.

2.2.1 Weaknesses

Because of lack of awareness toward ration allocation there is a long wait in time, people have to stand in Q for longer duration just to fulfill their basic needs.

2.2.2 How to Overcome

Ration shops do not open every day. Nor do they keep regular hours. So to avoid discomfort to the customers a system generated message will be delivered to their mobile when the stock is available and the shop is opened so that it does not cause any trouble to the customer.

2.3 AUTOMATIC RATION MATERIAL DISPENSING SYSTEM

India’s Public Distribution System (PDS) comprises ration shops which are responsible for distribution of ration to many poor families. Necessary commodities such as rice, kerosene, sugar, wheat etc., are supplied to the less privileged sections as per the eligibility and at fixed rates by the Indian government. However, the existing system suffers from major drawbacks such as storage of food grains, manual and inconsistent tallying of records against stock, black marketing and housekeeping. Hence, the conventional system needs some modification to ensure there is minimal wastage and food grains reach the right people. The major contribution of the hardware lies in acquiring records from the warehouses located in far off places to a centralized server managed by the administrator. This system also solves the problem of black marketing with the use of Aadhar cards and biometric scanning which are used for the verification of the consumer at the time of purchase. In addition to the above, another aspect of the system is the management of two types of databases, one of them is used for storing the data of the consumer and the other for maintaining the record of the availability of the ration in various warehouses which are located in different parts of the country.

2.3.1 Weaknesses

RFID chip can be hacked by someone in close physical range

RFID chip are highly expensive

2.3.2 How to Overcome

Our system will provide a notification regarding products to user.

User can make complaints and give suggestions for better service

2.4 Market Potential Competitive Advantage

Features	paper1	paper2	paper3	Proposed System
Notifications	No	No	Yes	Yes
Online Payment	No	No	Yes	Yes
Less Cost	No	No	No	Yes

Table 2.1: Competative analysis

Chapter 3

Proposed System

3.1 Analysis

3.1.1 Framework

- React Native

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

3.1.2 Algorithm

- Dijkstra

Dijkstra's algorithm has multiple variants, but one of the most commonly used is this one which is used to find the shortest paths from the source vertex to all other vertices in the graph.

For instance, a particular area as geographical map and take this geographical map as a GRAPH.

Now the locations on the map are our VERTICES in the algorithm. And the roads between locations are our EDGES. So, the WEIGHTS OF EDGES here are the distance between those two locations.

Through this, we can find out the shortest way between the two locations.

It used to know the shortest path between two cities on a map and is used in many applications such as Google maps. Though It uses more complex and efficient algorithms. But Dijkstra is the basic. It's also used in finding the shortest communication path between two nodes connected to a network.

- Sort Algorithms

Sorting is the most common and majorly used algorithms in mobile app development. The basic idea is to arrange the data or items in a definite pattern.It is essential to know when and where which one is to be used. Few instances, where you can find a direct application of sorting algorithm includes: Sorting on the basis of popularity, price, etc. on e-commerce applications It even works in sorting of contacts alphabetically in the mobile contacts application

3.1.3 Feasibility Study

1. Operational Feasibility :-

The necessities for software to work properly are Android 4.0.3 Jellybean device, Internet facility. The speed and performance of software is good enough since it is developed specially for android device.

2. Technical feasibility :-

Hardware feasibility.

Processor: Intel(R) Core 3 Duo or Higher

Installed memory: 4 GB

System type: 64 bit Operating System.

3. Software feasibility :-

Operating system: Android OS.

Software: Android Studio.

Technologies used: Java.

Database: firebase.

4. Economical feasibility :-

Economic analysis is the most frequently used method for evaluating the effectiveness of a new system. The purpose of the economic feasibility assessment is to determine the positive economic benefits to the organization that the proposed system will provide.

It includes quantification and identification of all the benefits expected. This assessment typically involves a cost / benefit analysis. Early estimates based on rule of thumb. Refined successively until we have a complete cost-benefit analysis. Clients and the analysts must evaluate the results of the feasibility study and make decisions together based on the results.

3.2 Project Requirements

3.2.1 Softwere Requirements

- Tools: Android Studio

- Database: firebase

- Programming Language : Java

- Scripting Language: XML

- Operating System: windows7/above

3.2.2 Hardware Requirements

- Processor : Intel Core 3 Duo or Higher

- Ram : 4GB

3.2.3 Project Architecture

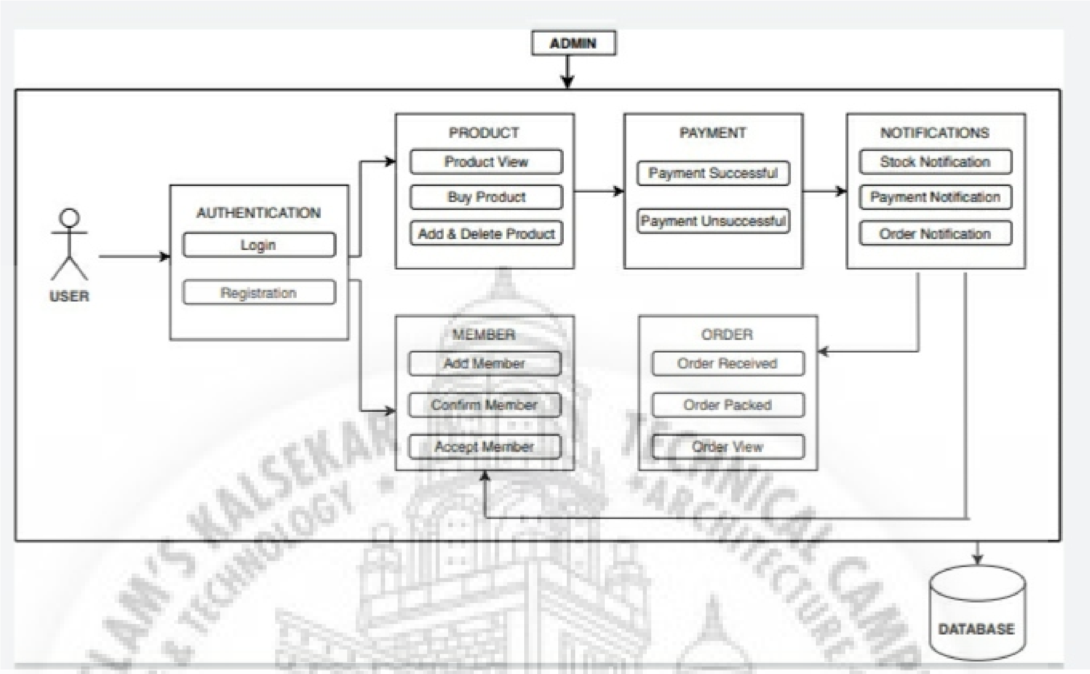


Figure 3.1: Architecture of E-Ration System

3.3 Methodology

3.3.1 Waterfall Method

The waterfall model is a linear sequential flow. In which progress is Seen as flowing steadily downwards (like a waterfall) through the phases of software implementation.This means that any phase in the development process begins only if the previous phase is complete. The waterfall approach does not define the process go to back to the previous phase to handle the changes in requirement. The waterfall approach is the earliest approach that was used for software development.I used waterfall model in my project because:-

1) Waterfall model is simple to implement and also amount of resource required for it are minimal.

2) In this model, output is generated after each stage, therefore it has high visibility. The client and project manager gets a feel that there is considerable progress.

3) This methodology is significantly better than the haphazard approach to develop software. It

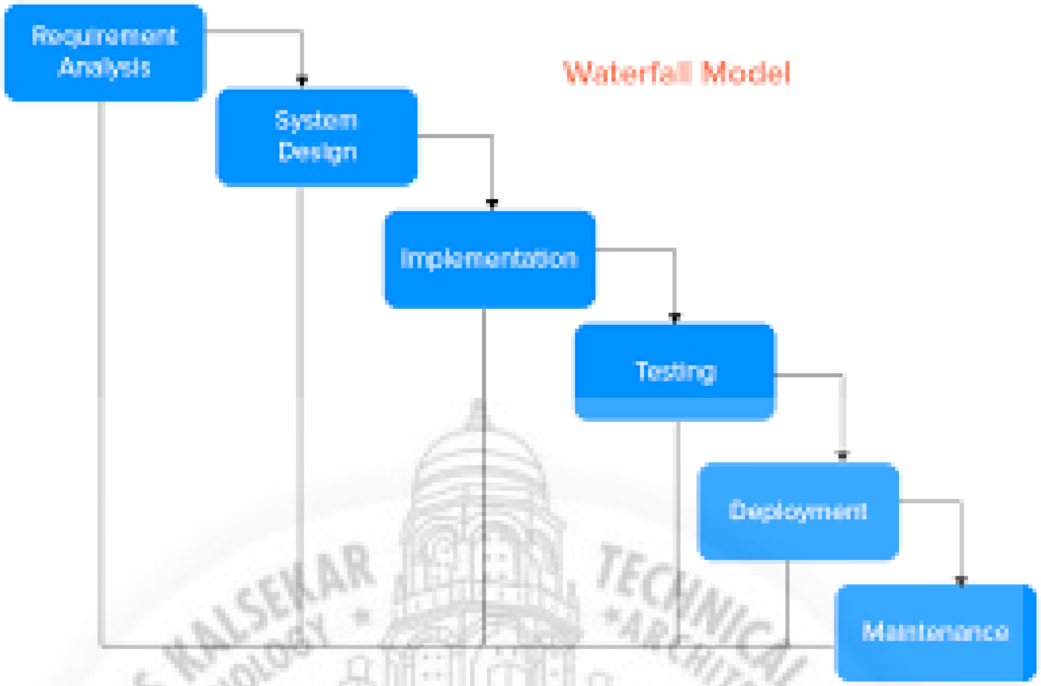


Figure 3.2: waterfall model for e-ration system

provides a template into which methods of analysis, design, and coding, testing, maintenance can be placed.

4) This methodology is preferred in projects where reliability is more important compared to time.

Models Involved

This model shows the complete E-ration system application for the android devices. It keeps the record of product and customer. The model shows several state flow charts interact with each other. Thus the user needs to also account for the order in which checking the regular graph of your android devices which is given in the application. In short I had developed an application which will help you to keep watch on your Ration status and it works like the E-ration pocket system of your android phone.

Chapter 4

Design Phase

4.1 System Requirements Definition

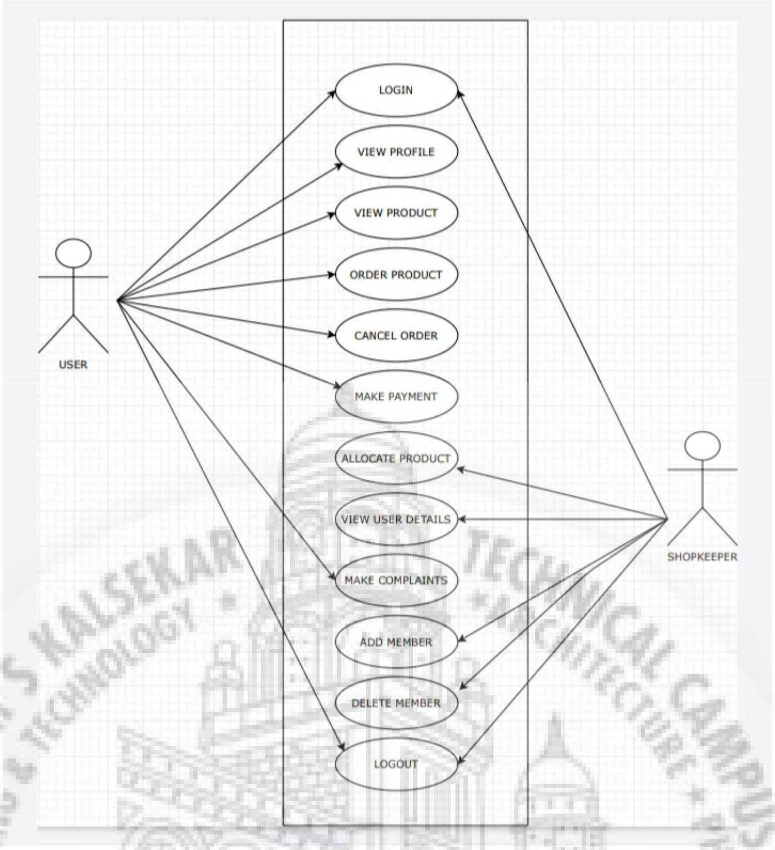
A software requirements specification (SRS) is a document that captures complete description about how the system is expected to perform. It is usually signed off at the end of requirements engineering phase.

4.1.1 Functional requirements

A functional requirement document defines the functionality of a system or one of its subsystems. It also depends upon the type of software, expected users and the type of system where the software is used.

Functional user requirements may be high-level statements of what the system should do but functional system requirements should also describe clearly about the system services in detail.

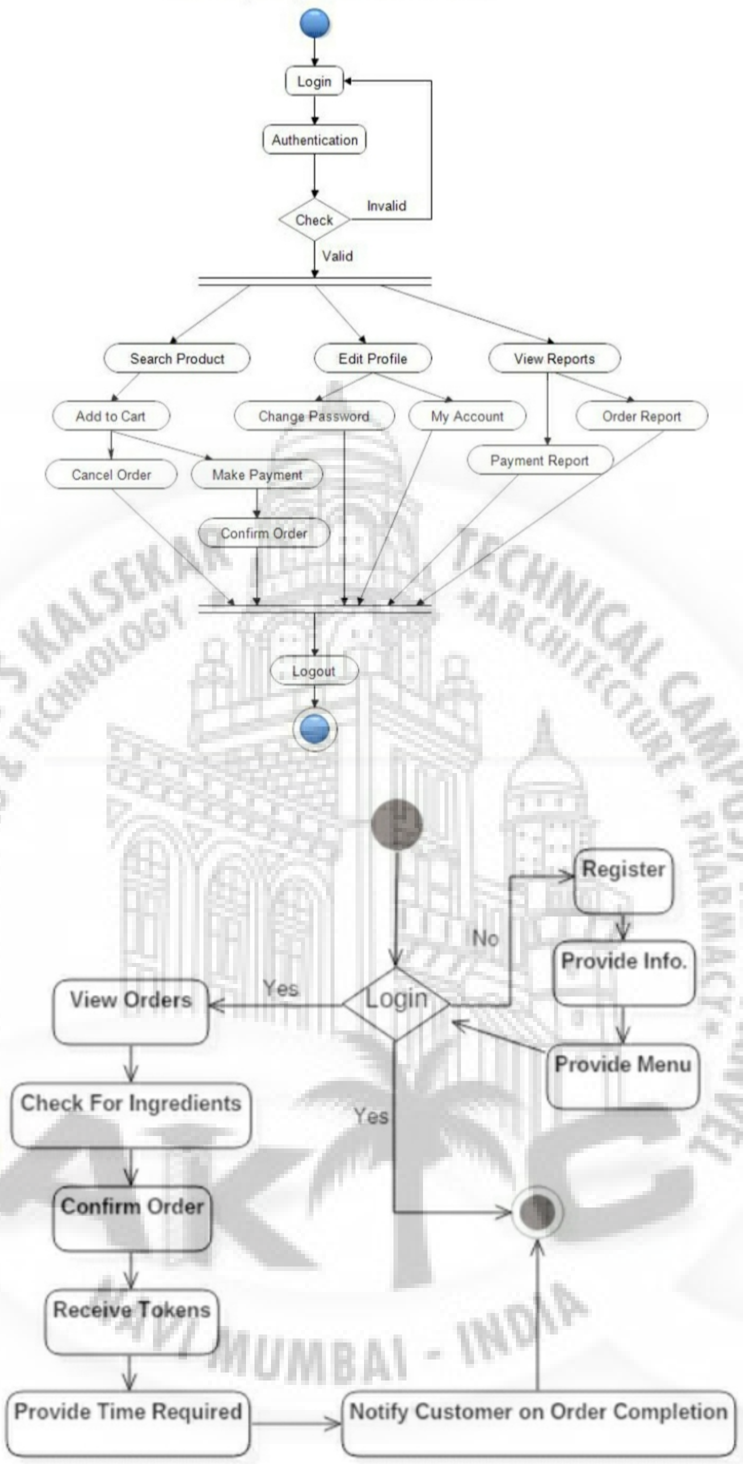
4.1.1.1 Use-case Diagram



The main purpose of a use case diagram is to show who interacts with your system, and the main goals they achieve with it. With the help of a use case diagram, you can discuss and communicate: The scenarios in which your system or application interacts with people, organizations, or external systems. The goals that it helps those actors achieve. The scope of your system.

Create Actors to represent classes of people, organizations, other systems, software or devices that interact with your system or subsystem. For each distinct set of goals, identify actors by their type or role, even though the physical persons or entities might be the same. Create Use Cases for each of the goals that each actor seeks to achieve with the system. Name and describe the use cases in words that the actor would understand, instead of implementation terms. Use Associations to link actors to use cases. The association between an actor and a use case can show a multiplicity at each end. A use case can be described in different levels of detail. At an early stage of design, just the name on the use case diagram is sufficient. Later, more detailed descriptions of the scenarios can be written.

4.1.1.2 Flow Diagram



A data-flow diagram is a way of representing a flow of data through a process or a system (usually an information system). The DFD also provides information about the outputs and inputs of each entity and the process itself. A data-flow diagram has no control flow, there are no decision rules and no loops. Specific operations based on the data can be represented by a flowchart.

4.1.2 System requirements (non-functional requirements)

Non-Functional Requirements are the constraints or the requirements imposed on the system. They specify the quality attribute of the software. Non-Functional Requirements deal with issues like scalability, maintainability, performance, portability, security, reliability, and many more.

4.2 System Architecture Design

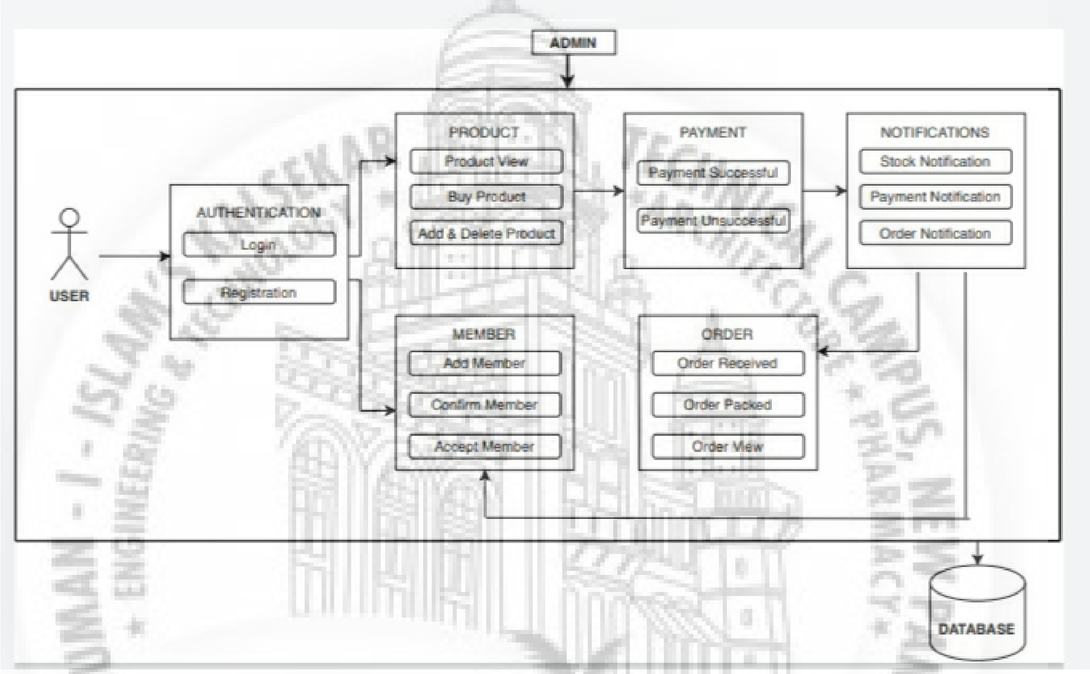


Figure 4.1: Architecture of E-Ration System

4.3 Sub-system Development

The Indian government distributes food for the poor people who are valid ration card holders and this manual system is facing many problems such as, Cardholders don't know the opening and closing of Ration shop and there is no mechanism to find errors such as duplicate, ineligible beneficiaries, Human intervention in updating transactions, Maintenance of thousands of all records in the form of hard copy is time-taking and difficult. No complaint forum. To Overcome these Problems, The "e-Ration shop" android application avoids above-mentioned problems. Each and every RationShop Geolocation will be located on the map and the information regarding RationShop in respective to the ration card and region is available in the Application and Shop opening and closing details will be available and cardholders details are available and Stock provided based on The estimate consumption of FPS grains in India is only 1 kg per person per month and changes depending on the Commodity. The PDS has been criticised for its inefficiences to serve the poorer sections of the population effectively

4.3.1 Authentication

User - User will enter their details on the registration page if the details matches with the shopkeeper database then the registration will be successful else registration will be invalid. Shopkeeper - Shopkeeper will enter the details on the registration page and that details get stored in database of shopkeeper and it Will be fetched for assigning the particular shopkeeper to user.

4.3.2 Product

Shopkeeper will notify the user about the availability of the product. And also shopkeeper can add or delete products. User checks the availability of products and can place the order.

4.3.3 Payment

User can use online mode for payment. On successful payment user will get the notification.

4.3.4 Notification

User will get notifications for everything like stock notification, payment notification, order notification.

4.3.5 Member

Shopkeeper can add member. Shopkeeper can delete member.

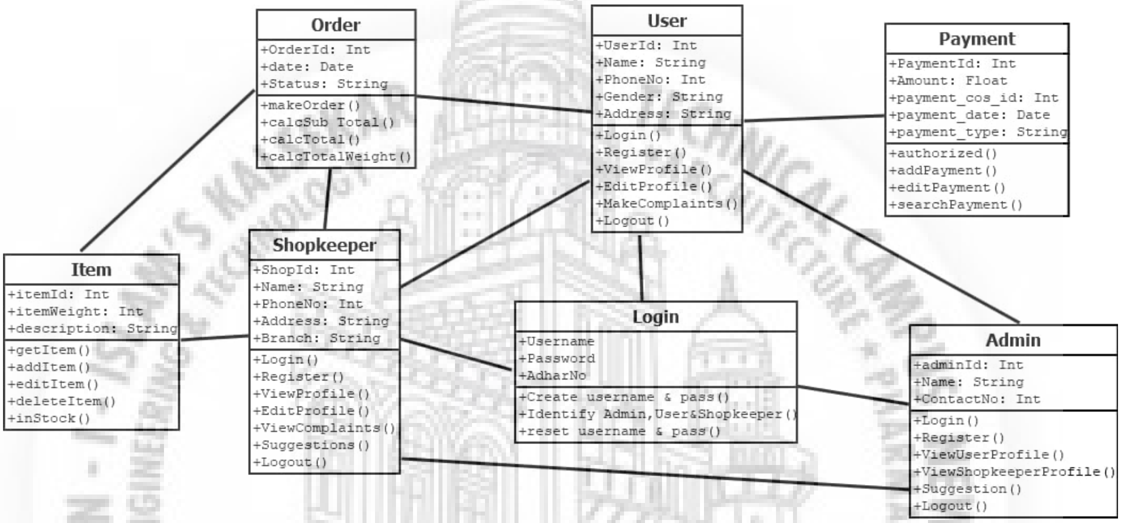
4.3.6 Order

User will get notified on successful ordered.

4.4 Systems Integration

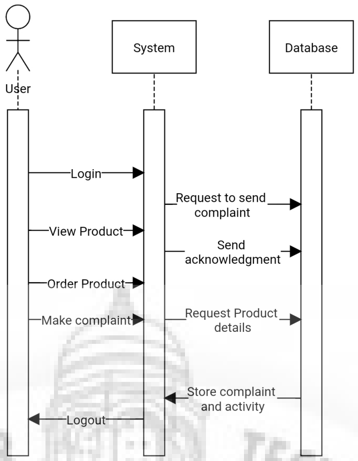
System integration is defined in engineering as the process of bringing together the component sub-systems into one system (an aggregation of subsystems cooperating so that the system is able to deliver the overarching functionality) and ensuring that the subsystems function together as a system, and in information technology as the process of linking together different computing systems and software applications physically or functionally, to act as a coordinated whole.

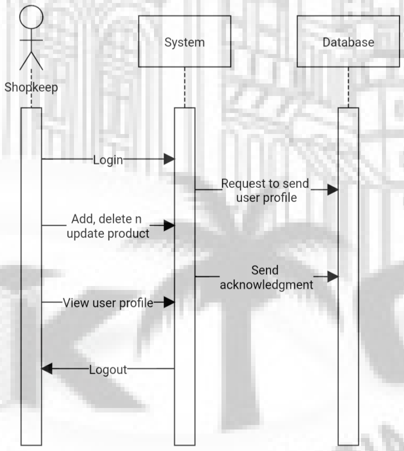
4.4.1 Class Diagram



Class diagrams are the main building block in object-oriented modeling. They are used to show the different objects in a system, their attributes, their operations and the relationships among them.

4.4.2 Sequence Diagram

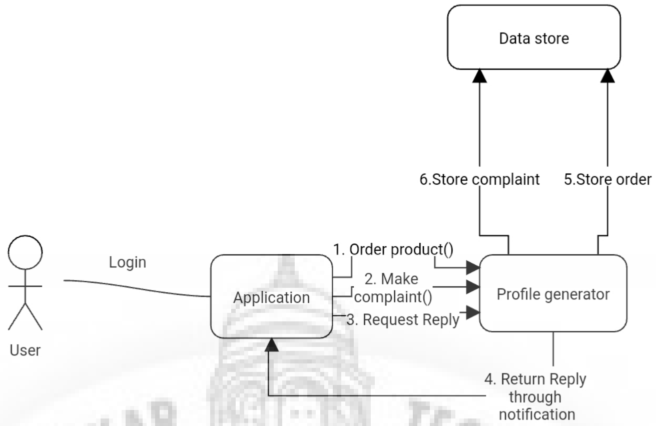


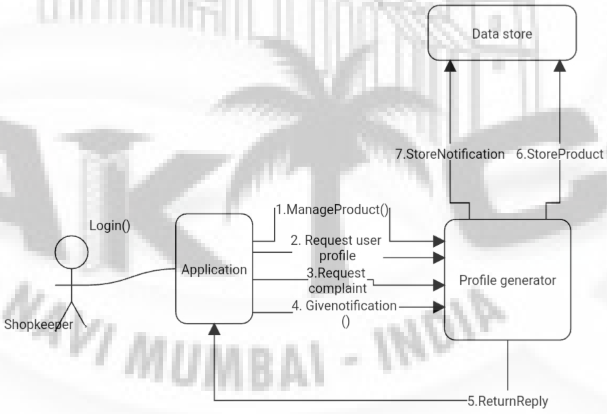


Sequence Diagram is a "Connection Diagram" that represents a single structure or storyline executing in a system. It is the second most used UML diagram behind the class diagram. Sequence Diagram shows what message is to be sent and when.

A sequence diagram is a good way to visualize and ratify various runtime framework.

4.4.3 Collaboration Diagram





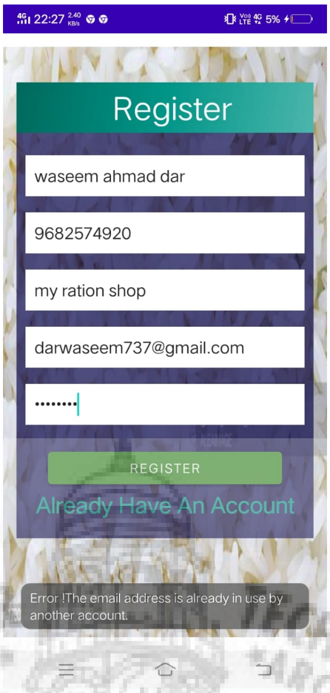
A collaboration diagram, also known as a communication diagram, is an illustration of the relationships and interactions among software objects in the Unified Modeling Language (UML). These diagrams can be used to portray the dynamic behavior of a particular use case and define the role of each object.

Chapter 5

Implementation

5.1 Authentication

User - user will enter the their details on the reg page if the details matches with the shopkeeper database then the registration will be successful else registration will be invalid. Shopkeeper - shopkeeper will enter the details on the registration page and that details get stored in database of shopkeeper and it Will be fetched for assigning the particular shopkeeper to user.



22:27

Register

waseem ahmad dar

9682574920

my ration shop

darwaseem737@gmail.com

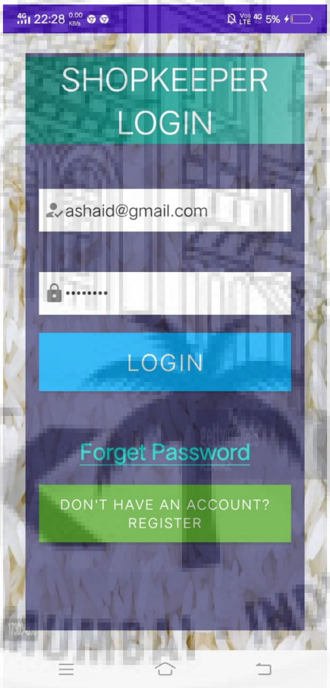
••••••••

REGISTER

Already Have An Account

Error !The email address is already in use by another account.

Figure 5.1: **Registration Page**



22:28

SHOPKEEPER LOGIN

ashaid@gmail.com

••••••••

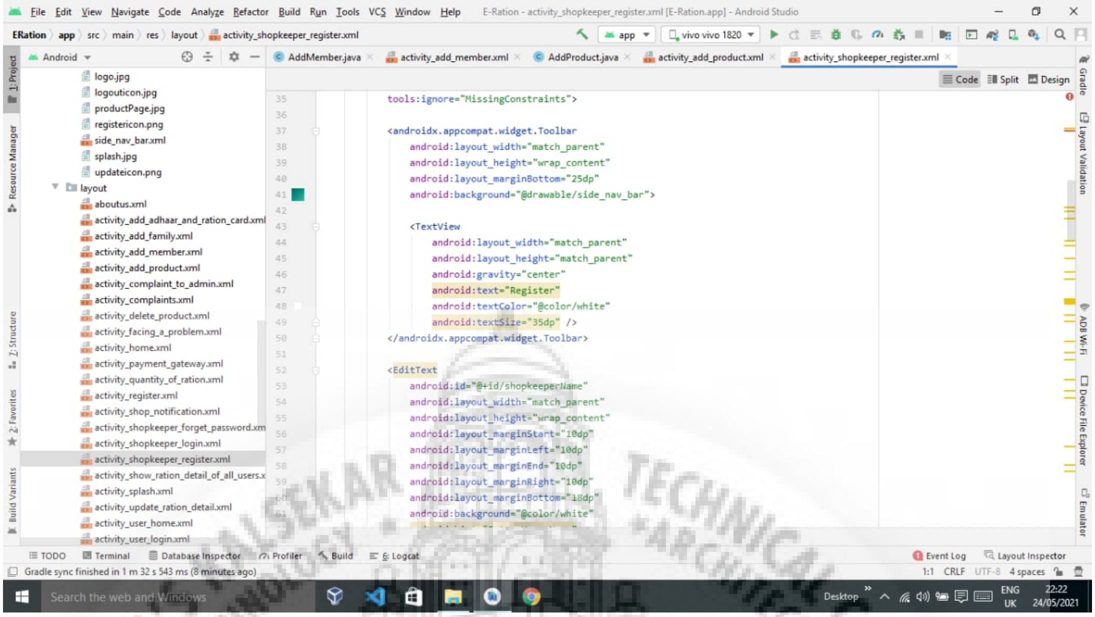
LOGIN

Forget Password

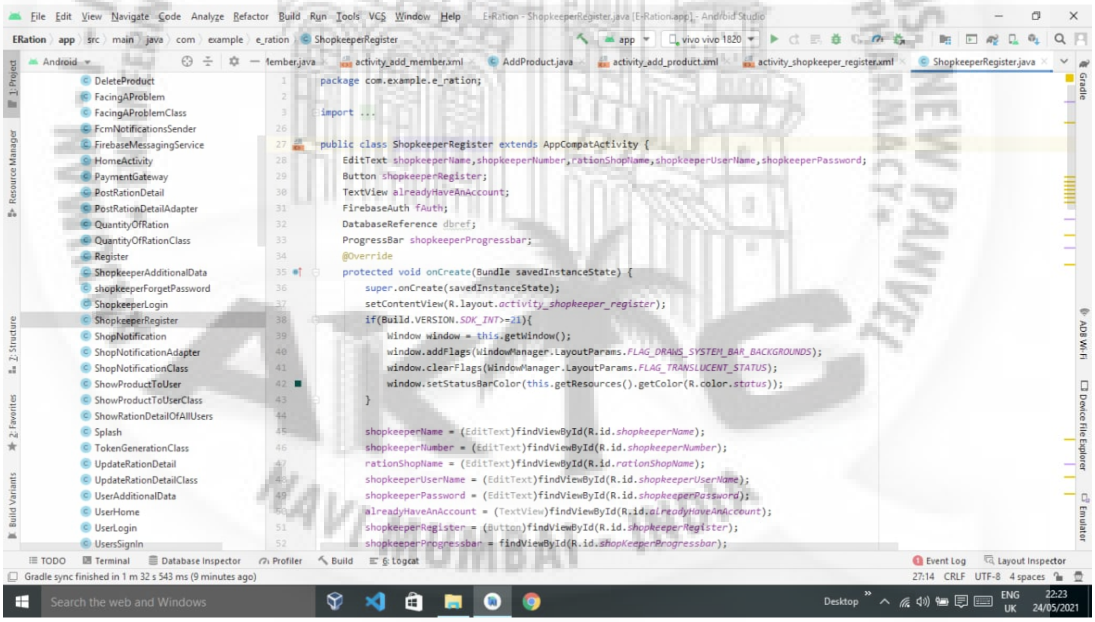
DON'T HAVE AN ACCOUNT? REGISTER

Figure 5.2: **Login Page**

CODE-REGISTER

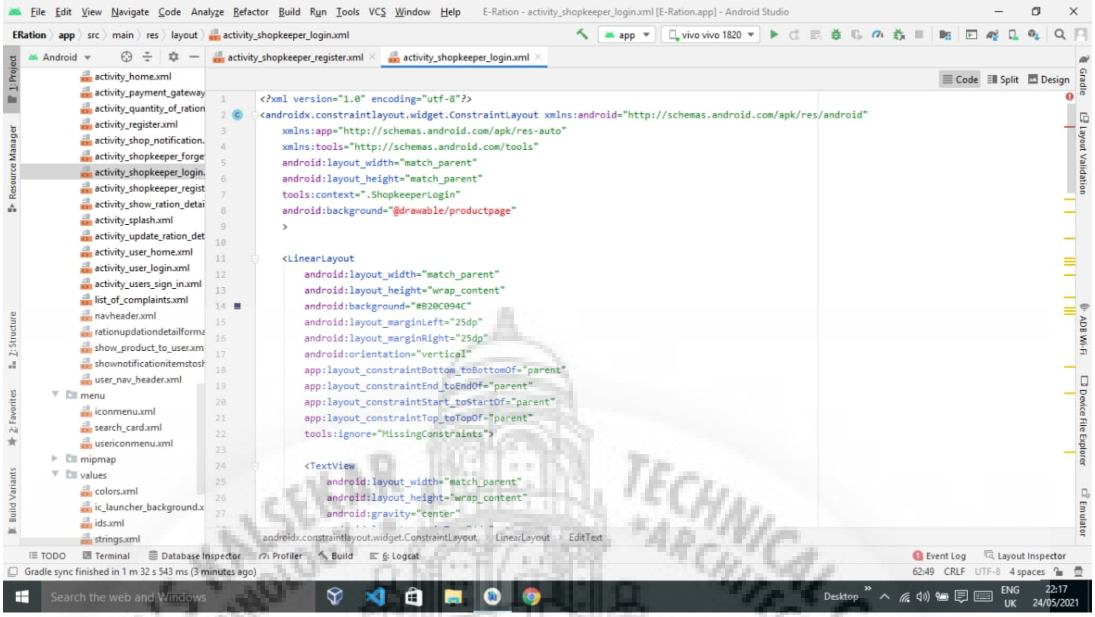

E-Ration - activity_shopkeeper_register.xml [E-Ration.app] - Android Studio

```
            tools:ignore="MissingConstraints">

            <androidx.appcompat.widget.Toolbar
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:layout_marginBottom="25dp"
                android:background="@drawable/side_nav_bar">

                <TextView
                    android:layout_width="match_parent"
                    android:layout_height="match_parent"
                    android:gravity="center"
                    android:text="Register"
                    android:textColor="@color/white"
                    android:textSize="35dp" />
            </androidx.appcompat.widget.Toolbar>

            <EditText
                android:id="@+id/shopkeeperName"
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:layout_marginStart="10dp"
                android:layout_marginLeft="10dp"
                android:layout_marginEnd="10dp"
                android:layout_marginRight="10dp"
                android:layout_marginBottom="18dp"
                android:background="@color/white"
```


E-Ration - ShopkeeperRegister.java [E-Ration.app] - Android Studio

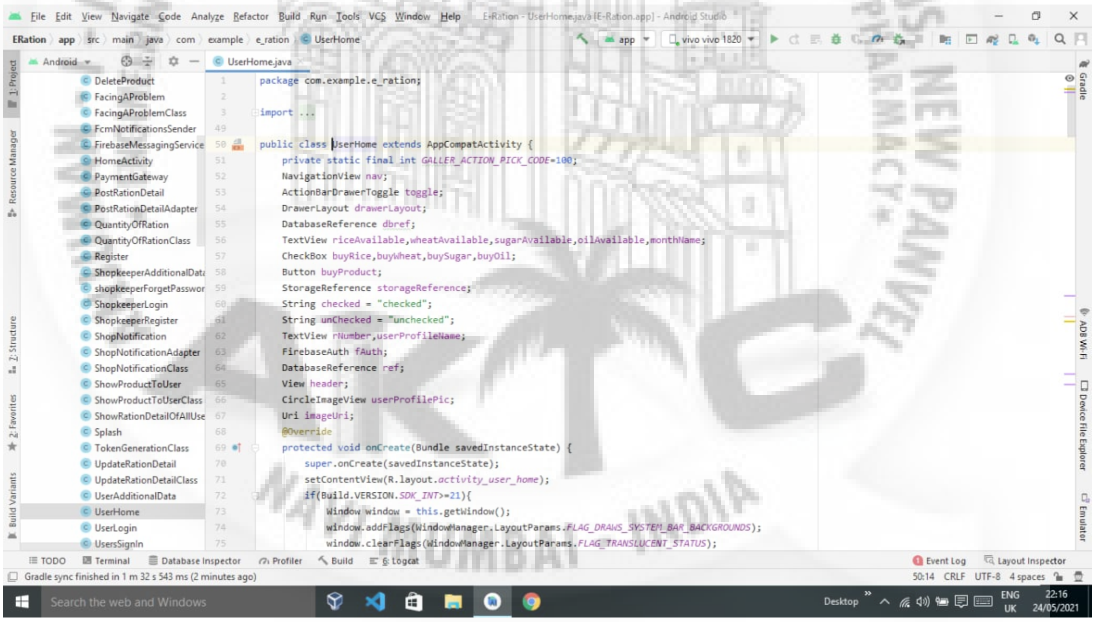
```
package com.example.e_ration;

import ...

public class ShopkeeperRegister extends AppCompatActivity {
    EditText shopkeeperName,shopkeeperNumber,rationShopName,shopkeeperUserName,shopkeeperPassword;
    Button shopkeeperRegister;
    TextView alreadyHaveAnAccount;
    FirebaseAuth fAuth;
    DatabaseReference dbref;
    ProgressBar shopkeeperProgressbar;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_shopkeeper_register);
        if(Build.VERSION.SDK_INT>=21){
            Window window = this.getWindow();
            window.addFlags(WindowManager.LayoutParams.FLAG_DRAWS_SYSTEM_BAR_BACKGROUNDS);
            window.clearFlags(WindowManager.LayoutParams.FLAG_TRANSLUCENT_STATUS);
            window.setStatusBarColor(this.getResources().getColor(R.color.status));
        }

        shopkeeperName = (EditText)findViewById(R.id.shopkeeperName);
        shopkeeperNumber = (EditText)findViewById(R.id.shopkeeperNumber);
        rationShopName = (EditText)findViewById(R.id.rationShopName);
        shopkeeperUserName = (EditText)findViewById(R.id.shopkeeperUserName);
        shopkeeperPassword = (EditText)findViewById(R.id.shopkeeperPassword);
        alreadyHaveAnAccount = (TextView)findViewById(R.id.alreadyHaveAnAccount);
        shopkeeperRegister = (Button)findViewById(R.id.shopkeeperRegister);
        shopkeeperProgressbar = findViewById(R.id.shopKeeperProgressbar);
```

CODE-LOGIN


E-Ration - activity_shopkeeper_login.xml [E-Ration.app] - Android Studio

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".ShopkeeperLogin"
    android:background="@drawable/productpage"
    >

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:background="#B20C094C"
        android:layout_marginLeft="25dp"
        android:layout_marginRight="25dp"
        android:orientation="vertical"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        tools:ignore="MissingConstraints">

        <TextView
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:gravity="center"
```

Gradle sync finished in 1 m 32 s 543 ms (3 minutes ago)

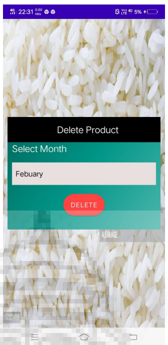

E-Ration - UserHome.java [E-Ration.app] - Android Studio

```
package com.example.e_ration;

import ...

public class UserHome extends AppCompatActivity {
    private static final int GALLER_ACTION_PICK_CODE=100;
    NavigationView nav;
    ActionBarDrawerToggle toggle;
    DrawerLayout drawerLayout;
    DatabaseReference dbref;
    TextView riceAvailable,wheatAvailable,sugarAvailable,oilAvailable,monthName;
    CheckBox buyRice,buyWheat,buySugar,buyOil;
    Button buyProduct;
    StorageReference storageReference;
    String checked = "checked";
    String unChecked = "unchecked";
    TextView rNumber,userProfileName;
    FirebaseAuth fAuth;
    DatabaseReference ref;
    View header;
    CircleImageView userProfilePic;
    Uri imageUri;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_user_home);
        if(Build.VERSION.SDK_INT>=21){
            Window window = this.getWindow();
            window.addFlags(WindowManager.LayoutParams.FLAG_DRAWS_SYSTEM_BAR_BACKGROUNDS);
            window.clearFlags(WindowManager.LayoutParams.FLAG_TRANSLUCENT_STATUS);
```

Gradle sync finished in 1 m 32 s 543 ms (2 minutes ago)

5.2 Product

Shopkeeper will notify the user about the availability of the product. And also shopkeeper can add or delete products. User checks the availability of products and can place the order.

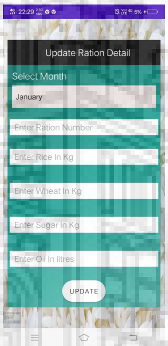


Delete Product

Select Month

Febuary

Figure 5.3: **Delete Product**



Update Ration Detail

Select Month

January

Enter Ration Number

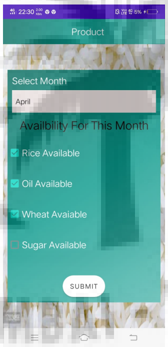
Enter Rice In Kg

Enter Wheat In Kg

Enter Sugar In Kg

Enter Oil In litres

Figure 5.4: **Product Availability**



Product

Select Month

April

Availbility For This Month

☑ Rice Available

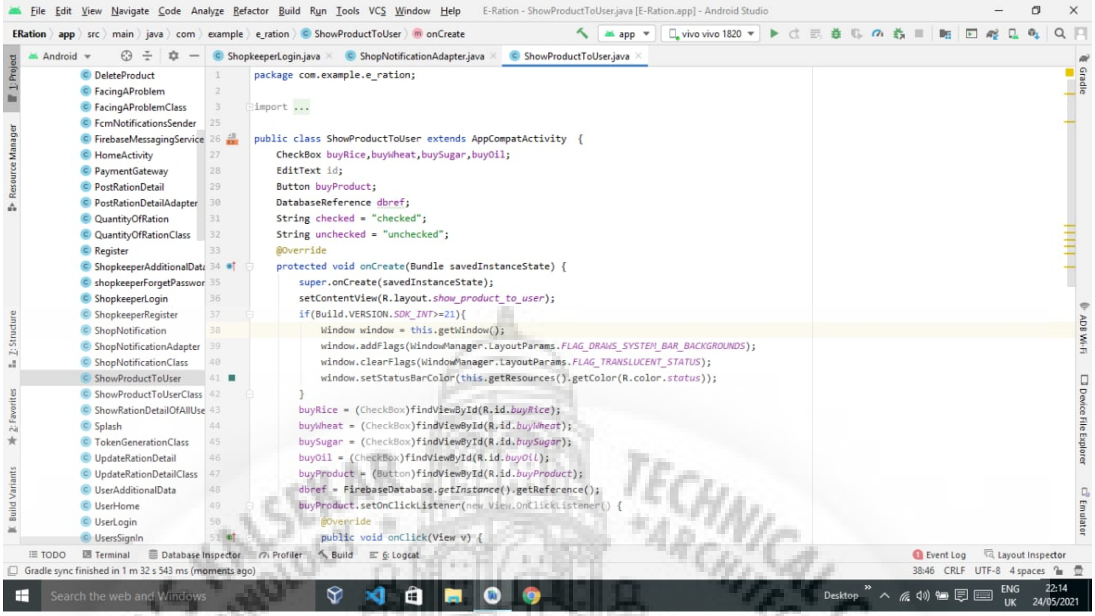
☑ Oil Available

☑ Wheat Avaiable

□ Sugar Available

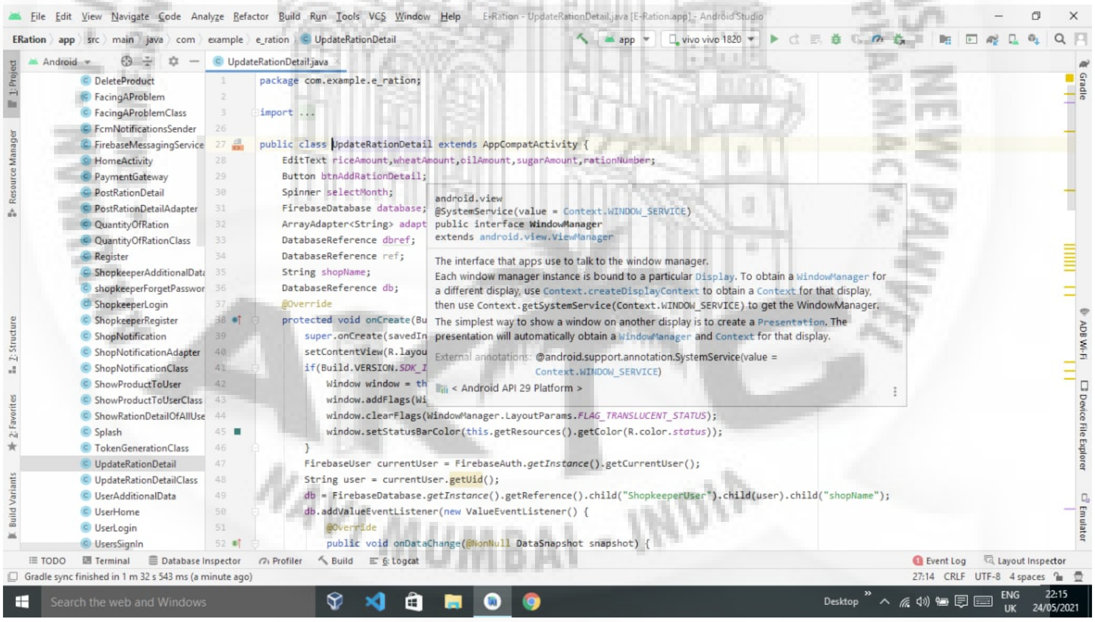
Figure 5.5: **Update Ration Details**

CODE-PRODUCT



```java
package com.example.e_ration;

import ...

public class ShowProductToUser extends AppCompatActivity  {
    CheckBox buyRice,buyWheat,buySugar,buyOil;
    EditText id;
    Button buyProduct;
    DatabaseReference dbref;
    String checked = "checked";
    String unchecked = "unchecked";
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.show_product_to_user);
        if(Build.VERSION.SDK_INT>=21){
            Window window = this.getWindow();
            window.addFlags(WindowManager.LayoutParams.FLAG_DRAWS_SYSTEM_BAR_BACKGROUNDS);
            window.clearFlags(WindowManager.LayoutParams.FLAG_TRANSLUCENT_STATUS);
            window.setStatusBarColor(this.getResources().getColor(R.color.status));
        }
        buyRice = (CheckBox)findViewById(R.id.buyRice);
        buyWheat = (CheckBox)findViewById(R.id.buyWheat);
        buySugar = (CheckBox)findViewById(R.id.buySugar);
        buyOil = (CheckBox)findViewById(R.id.buyOil);
        buyProduct = (Button)findViewById(R.id.buyProduct);
        dbref = FirebaseDatabase.getInstance().getReference();
        buyProduct.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
```

Gradle sync finished in 1 m 32 s 543 ms (moments ago)


E-Ration - UpdateRationDetail.java [E-Ration.app] - Android Studio
ERation › app › src › main › java › com › example › e_ration › UpdateRationDetail
UpdateRationDetail.java

```java
package com.example.e_ration;

import ...

public class UpdateRationDetail extends AppCompatActivity {
    EditText riceAmount,wheatAmount,oilAmount,sugarAmount,rationNumber;
    Button btnAddRationDetail;
    Spinner selectMonth;
    FirebaseDatabase database;
    ArrayAdapter<String> adapt
    DatabaseReference dbref;
    DatabaseReference ref;
    String shopName;
    DatabaseReference db;
    @Override
    protected void onCreate(Bu
        super.onCreate(savedIn
        setContentView(R.layou
        if(Build.VERSION.SDK_I
            Window window = th
            window.addFlags(Wi
            window.clearFlags(WindowManager.LayoutParams.FLAG_TRANSLUCENT_STATUS);
            window.setStatusBarColor(this.getResources().getColor(R.color.status));
        }
        FirebaseUser currentUser = FirebaseAuth.getInstance().getCurrentUser();
        String user = currentUser.getUid();
        db = FirebaseDatabase.getInstance().getReference().child("ShopkeeperUser").child(user).child("shopName");
        db.addValueEventListener(new ValueEventListener() {
            @Override
            public void onDataChange(@NonNull DataSnapshot snapshot) {
```

```
android.view
@SystemService(value = Context.WINDOW_SERVICE)
public interface WindowManager
extends android.view.ViewManager
```

The interface that apps use to talk to the window manager.
Each window manager instance is bound to a particular Display. To obtain a WindowManager for a different display, use Context.createDisplayContext to obtain a Context for that display, then use Context.getSystemService(Context.WINDOW_SERVICE) to get the WindowManager.
The simplest way to show a window on another display is to create a Presentation. The presentation will automatically obtain a WindowManager and Context for that display.

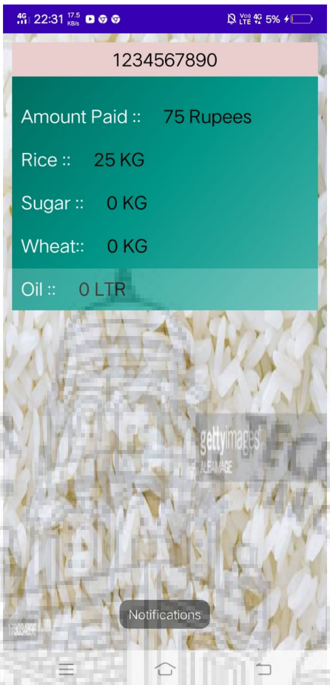
External annotations: @android.support.annotation.SystemService(value = Context.WINDOW_SERVICE)

< Android API 29 Platform >

Gradle sync finished in 1 m 32 s 543 ms (a minute ago)

5.3 Payment

User can use online mode for payment. On successful payment user will get the notification.



1234567890

Amount Paid :: 75 Rupees

Rice :: 25 KG

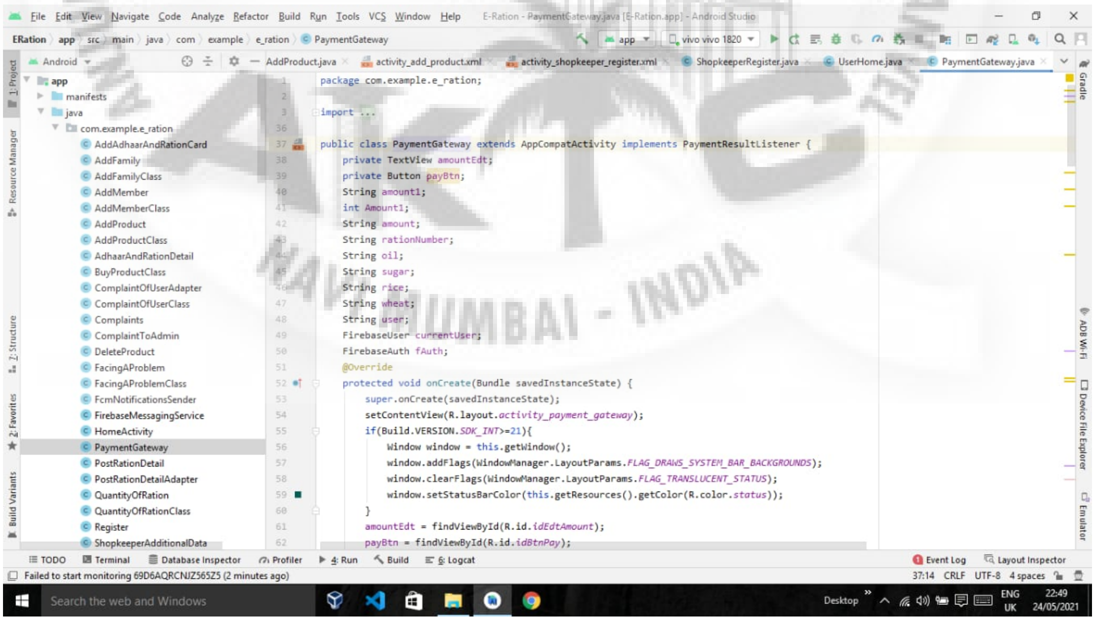
Sugar :: 0 KG

Wheat:: 0 KG

Oil :: 0 LTR

Figure 5.6: **Payment Page**

CODE-PAYMENT



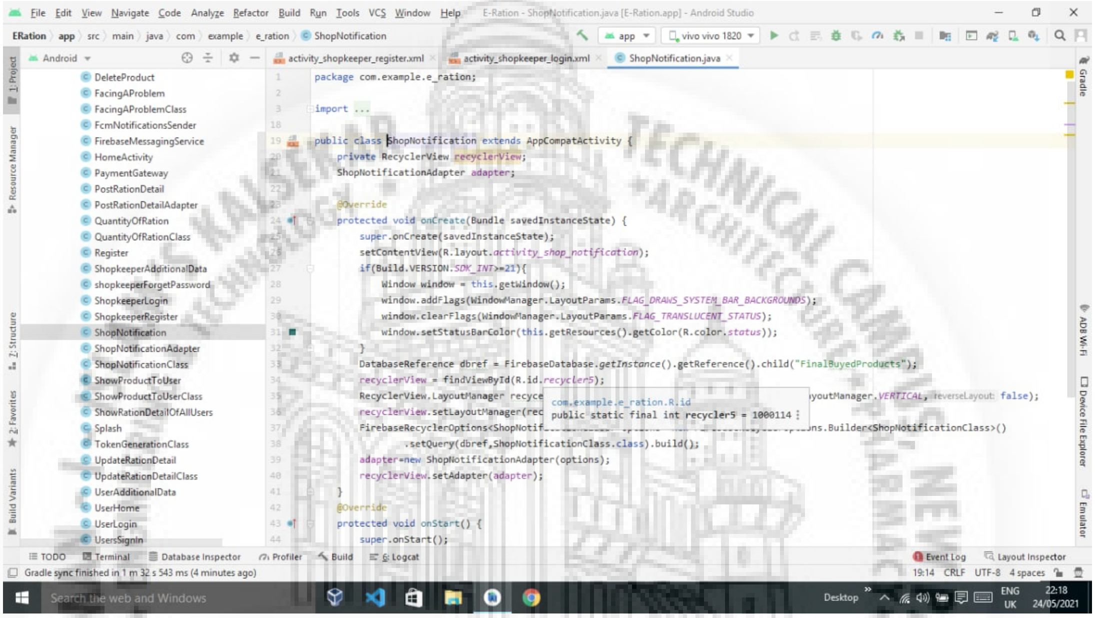
```
package com.example.e_ration;

import ...

public class PaymentGateway extends AppCompatActivity implements PaymentResultListener {
    private TextView amountEdt;
    private Button payBtn;
    String amount1;
    int Amount1;
    String amount;
    String rationNumber;
    String oil;
    String sugar;
    String rice;
    String wheat;
    String user;
    FirebaseUser currentUser;
    FirebaseAuth fAuth;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_payment_gateway);
        if(Build.VERSION.SDK_INT>=21){
            Window window = this.getWindow();
            window.addFlags(WindowManager.LayoutParams.FLAG_DRAWS_SYSTEM_BAR_BACKGROUNDS);
            window.clearFlags(WindowManager.LayoutParams.FLAG_TRANSLUCENT_STATUS);
            window.setStatusBarColor(this.getResources().getColor(R.color.status));
        }
        amountEdt = findViewById(R.id.idEdtAmount);
        payBtn = findViewById(R.id.idBtnPay);
```

5.4 Notification

User will get notifications for everything like stock notification, payment notification, order notification.

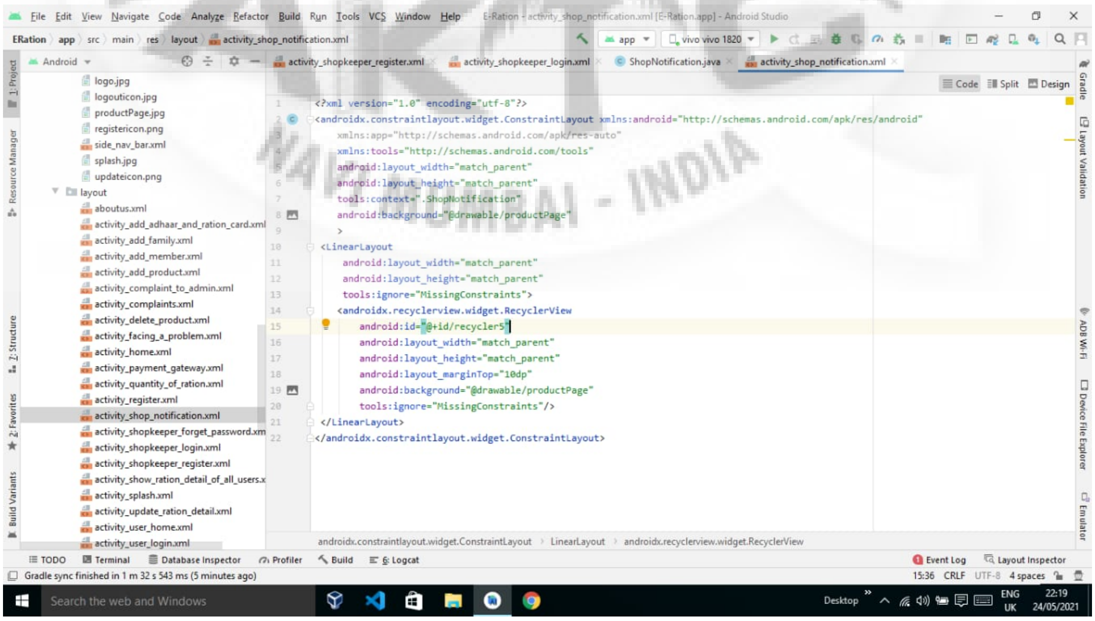
Figure 5.7: **Notification Page**

CODE-NOTIFICATIONS



```
package com.example.e_ration;

import ...

public class ShopNotification extends AppCompatActivity {
    private RecyclerView recyclerView;
    ShopNotificationAdapter adapter;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_shop_notification);
        if(Build.VERSION.SDK_INT>=21){
            Window window = this.getWindow();
            window.addFlags(WindowManager.LayoutParams.FLAG_DRAWS_SYSTEM_BAR_BACKGROUNDS);
            window.clearFlags(WindowManager.LayoutParams.FLAG_TRANSLUCENT_STATUS);
            window.setStatusBarColor(this.getResources().getColor(R.color.status));
        }
        DatabaseReference dbref = FirebaseDatabase.getInstance().getReference().child("FinalBuyedProducts");
        recyclerView = findViewById(R.id.recycler5);
        RecyclerView.LayoutManager recyce                                          youtManager.VERTICAL, reverseLayout: false);
        recyclerView.setLayoutManager(rec
        FirebaseRecyclerOptions<ShopNotif                                          ns.Builder<ShopNotificationClass>()
                .setQuery(dbref,ShopNotificationClass.class).build();
        adapter=new ShopNotificationAdapter(options);
        recyclerView.setAdapter(adapter);
    }
    @Override
    protected void onStart() {
        super.onStart();
```

```
com.example.e_ration.R.id
public static final int recycler5 = 1000114
```



```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".ShopNotification"
    android:background="@drawable/productPage"
    >
 <LinearLayout
     android:layout_width="match_parent"
     android:layout_height="match_parent"
     tools:ignore="MissingConstraints">
    <androidx.recyclerview.widget.RecyclerView
        android:id="@+id/recycler5"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:layout_marginTop="10dp"
        android:background="@drawable/productPage"
        tools:ignore="MissingConstraints"/>
 </LinearLayout>
</androidx.constraintlayout.widget.ConstraintLayout>
```

5.5 Member

Shopkeeper can add member. Shopkeeper can delete member.

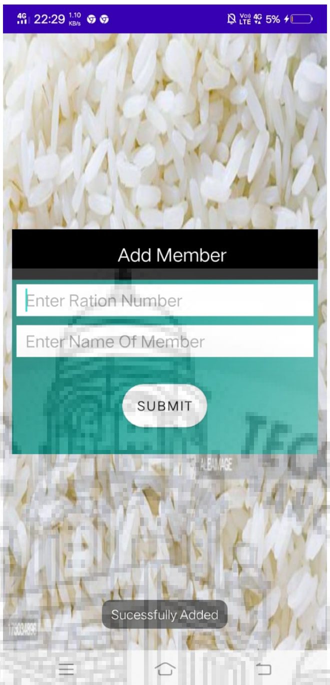
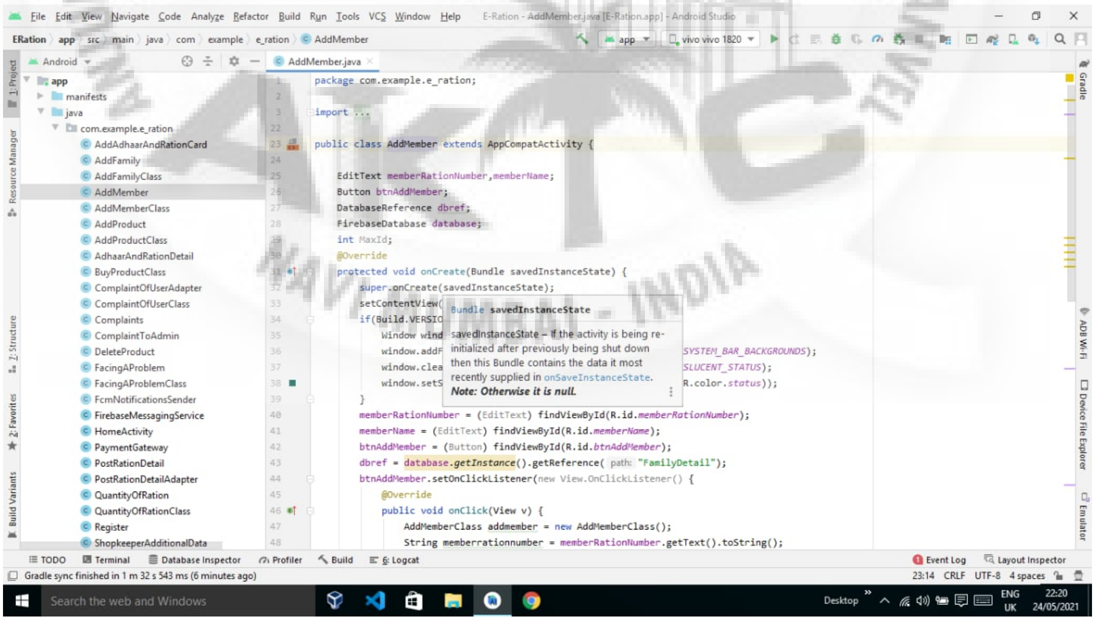


Figure 5.8: **Member Page**

CODE-MEMBER

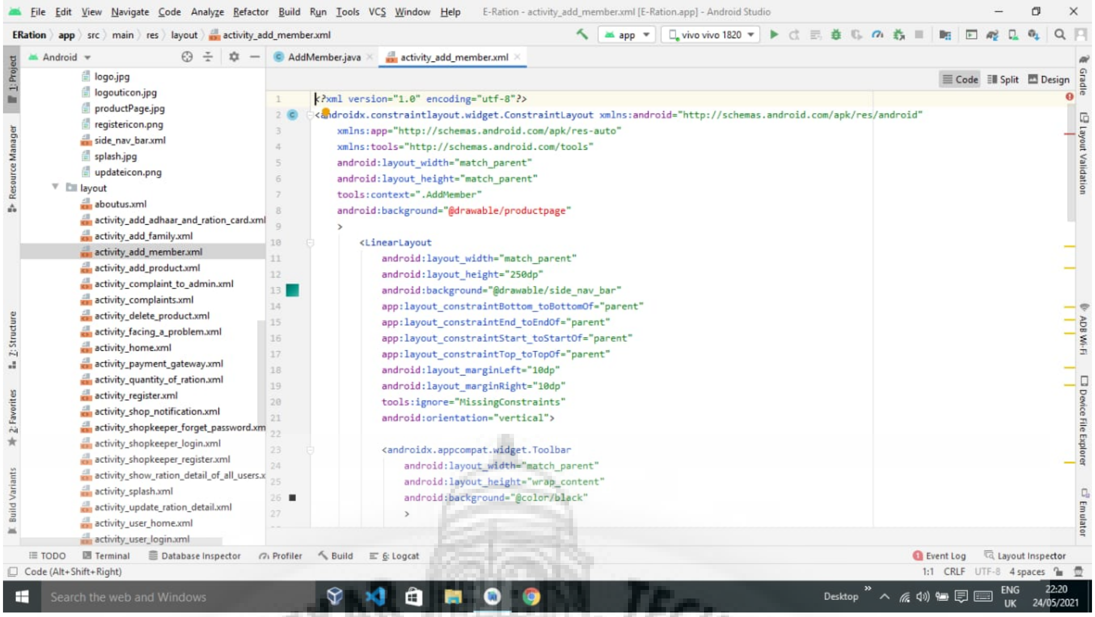


```
package com.example.e_ration;

import ...

public class AddMember extends AppCompatActivity {

    EditText memberRationNumber,memberName;
    Button btnAddMember;
    DatabaseReference dbref;
    FirebaseDatabase database;
    int MaxId;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(
        if(Build.VERSIO
            Window wind
            window.addF                                                   SYSTEM_BAR_BACKGROUNDS);
            window.clea                                                   SLUCENT_STATUS);
            window.setS                                                   R.color.status));
        }
        memberRationNumber = (EditText) findViewById(R.id.memberRationNumber);
        memberName = (EditText) findViewById(R.id.memberName);
        btnAddMember = (Button) findViewById(R.id.btnAddMember);
        dbref = database.getInstance().getReference( path: "FamilyDetail");
        btnAddMember.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                AddMemberClass addmember = new AddMemberClass();
                String memberrationnumber = memberRationNumber.getText().toString();
```

Bundle savedInstanceState

savedInstanceState – If the activity is being re-initialized after previously being shut down then this Bundle contains the data it most recently supplied in onSaveInstanceState. ***Note: Otherwise it is null.***

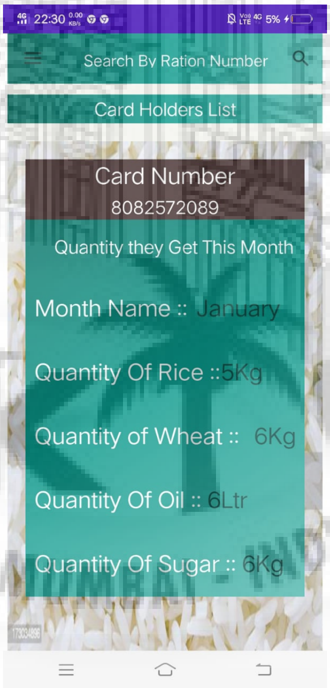
Gradle sync finished in 1 m 32 s 543 ms (6 minutes ago)



```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".AddMember"
    android:background="@drawable/productpage"
    >
        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="250dp"
            android:background="@drawable/side_nav_bar"
            app:layout_constraintBottom_toBottomOf="parent"
            app:layout_constraintEnd_toEndOf="parent"
            app:layout_constraintStart_toStartOf="parent"
            app:layout_constraintTop_toTopOf="parent"
            android:layout_marginLeft="10dp"
            android:layout_marginRight="10dp"
            tools:ignore="MissingConstraints"
            android:orientation="vertical">

            <androidx.appcompat.widget.Toolbar
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:background="@color/black"
                >
```

5.6 Order

User will get notified on successful ordered.



Search By Ration Number

Card Holders List

Card Number

8082572089

Quantity they Get This Month

Month Name :: January

Quantity Of Rice ::5Kg

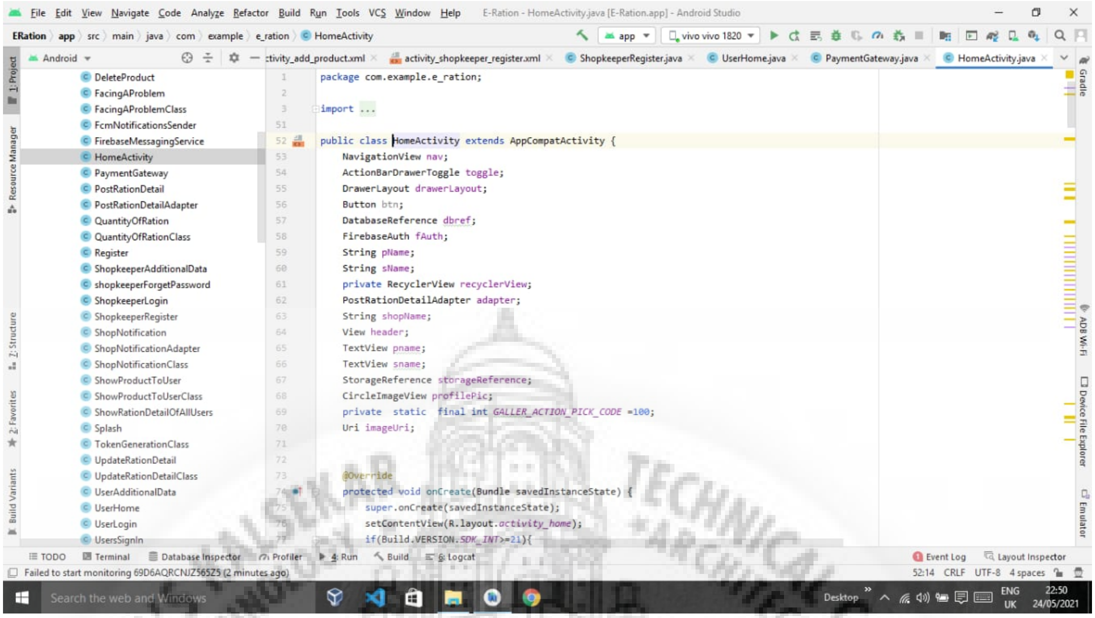
Quantity of Wheat :: 6Kg

Quantity Of Oil :: 6Ltr

Quantity Of Sugar :: 6Kg

Figure 5.9: **Order Page**

CODE-ORDER



- DeleteProduct
- FacingAProblem
- FacingAProblemClass
- FcmNotificationsSender
- FirebaseMessagingService
- HomeActivity
- PaymentGateway
- PostRationDetail
- PostRationDetailAdapter
- QuantityOfRation
- QuantityOfRationClass
- Register
- ShopkeeperAdditionalData
- shopkeeperForgetPassword
- ShopkeeperLogin
- ShopkeeperRegister
- ShopNotification
- ShopNotificationAdapter
- ShopNotificationClass
- ShowProductToUser
- ShowProductToUserClass
- ShowRationDetailOfAllUsers
- Splash
- TokenGenerationClass
- UpdateRationDetail
- UpdateRationDetailClass
- UserAdditionalData
- UserHome
- UserLogin
- UsersSignIn

```java
package com.example.e_ration;

import ...

public class HomeActivity extends AppCompatActivity {
    NavigationView nav;
    ActionBarDrawerToggle toggle;
    DrawerLayout drawerLayout;
    Button btn;
    DatabaseReference dbref;
    FirebaseAuth fAuth;
    String pName;
    String sName;
    private RecyclerView recyclerView;
    PostRationDetailAdapter adapter;
    String shopName;
    View header;
    TextView pname;
    TextView sname;
    StorageReference storageReference;
    CircleImageView profilePic;
    private  static  final int GALLER_ACTION_PICK_CODE =100;
    Uri imageUri;


    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_home);
        if(Build.VERSION.SDK_INT>=21){
```

Failed to start monitoring 69D6AQRCNJZ565Z5 (2 minutes ago)

52:14 CRLF UTF-8 4 spaces

Chapter 6

Maintenance

6.1 Future enhancements.

- Keyboard functions will be added.
- We want to design more user friendly app later on.

6.2 User manual

How to USE?

- User must have a .apk file of E-RATION APP installed in their respective android phone.
- This app runs on 3.0 android version(Honeycomb) and above.
- User can order product by staying at home.
- It shows availibility of product.
- It also shows notifications given by shopkeeper.
- It shows status of User to shopkeeper.
- By using this shopkeeper can maintain it in database which make easy for shokeeper to keep record of their users.

Chapter 7

Conclusion

Lastly, E-Ration Shop is to ensure that only the authorised people receives the subsidized food materials and other routine chores and reporting are done with very least human intervention to ensure the error free operation of the PDS scheme, Authorised people Should get all the benefits of the scheme easily and there is no bogus entries of details and all the schemes provided by the government will go in to the public without any bogus or malpractices by the dealer.

References

[1] Automatic ration distribution system 2016 3rd International Conference on Computing for Sustainable Global Development (INDIACom).

[2] An Automated Approach to Public Distribution System Using Internet of Things.Department of CSE IT, The NorthCap University, Gurugram-122001, India.

[3] SMART RATIONING SYSTEM.International Journal of Advanced Research in Electrical,Electronics and Instrumentation Engineering

[4] Public Distribution System in Rural India: Implications for Food Safety and Consumer Protection.Utkal University, Bhubaneswar, -751010, India Xavier Institute of Management,Bhubaneswar,-751010, India

[6] www.javatpoint.com/android-tutorial

[7] firebase.google.com/docs

[8]www.tutorialspoint.com/react$_{n}ative/index.html$

[9] developer.android.com/training/basics/firstapp

ACKNOWLEDGMENT

I would like to take the opportunity to express my sincere thanks to my guide **Prof. Mukhtar Ansari**, Assistant Professor, Department of Computer Engineering, AIKTC, School of Engineering, Panvel for his invaluable support and guidance throughout my project research work. Without his kind guidance & support this was not possible.

I am grateful to him/her for his timely feedback which helped me track and schedule the process effectively. His/her time, ideas and encouragement that he gave is help me to complete my project efficiently.

I would also like to thank **Dr. Abdul Razak Honnutagi**, AIKTC, Panvel, for his encouragement and for providing an outstanding academic environment, also for providing the adequate facilities.

I am thankful to **Prof. Tabrez Khan**, HOD, Department of Computer Engineering, AIKTC, School of Engineering, Panvel and all my B.E. teachers for providing advice and valuable guidance.

I also extend my sincere thanks to all the faculty members and the non-teaching staff and friends for their cooperation.

Last but not the least, I am thankful to all my family members whose constant support and encouragement in every aspect helped me to complete my project.

Shaikh Mumtaz Bano Abdul Aziz
Roll No. 17CO13
(Department of Computer Engineering)
University of Mumbai.

Dar Waseem Ahmad Misra Bano

Roll No.17CO54

(Department of Computer Engineering)

University of Mumbai.

Dev Shahid Ahmad Zoona

Roll No.17CO43

(Department of Computer Engineering)

University of Mumbai.

Patel Zulekha Ibrahimsha

Roll No.17CO06

(Department of Computer Engineering)

University of Mumbai.