

# **AUTOMATED WATER METER READING - AQUAMETER**

**A PROJECT REPORT**

Submitted By

**ANUPAMA REGI**

**TVE20MCA-2017**

**to**

APJ Abdul Kalam Technological University

in partial fulfilment of the requirements for the award of the degree

**of**

Master of Computer Applications



**Department of Computer Applications**

College of Engineering

Trivandrum-695016

MARCH 2022

# Declaration

I hereby declare that the project report titled "**AUTOMATED WATER METER READING - AQUAMETER**" submitted for partial fulfillment of the requirements for the award of degree of Master of Computer Applications of the APJ Abdul Kalam Technological University, Kerala is a bonafide work done by me under supervision of Smt. Deepa S S, Asst.Professor. This submission represents my ideas in my words. I have adequately and accurately cited and referenced the original sources. I also declare that I have adhered to ethics of academic honesty and integrity as directed in the ethics policy of the college and have not misrepresented or fabricated any data or idea or fact or source in my submission. I understand that any violation of the above will be a cause for disciplinary action by the Institute and/or University and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been obtained. This report has not previously formed the basis for the award of any degree, diploma or similar title.

Place : Trivandrum

**Anupama Regi**

Date : 17/02/2022

DEPARTMENT OF COMPUTER APPLICATIONS

COLLEGE OF ENGINEERING

TRIVANDRUM



CERTIFICATE

This is to certify that the report entitled **Automated Water Meter Reading-AquaMeter** submitted by **Anupama Regi** to the APJ Abdul Kalam Technological University in partial fulfillment of the requirements for the award of the Degree of Master of Computer Applications is a bonafide record of the project work carried out by her under the guidance and supervision of Smt. Deepa S S, Asst.Professor. This report in any form has not been submitted to any University or Institute for any purpose.

Internal Supervisor

Head of the Dept

# Acknowledgement

First and foremost I thank **GOD**, the Almighty, for the successful and timely completion of this project. I express my sincere gratitude and heartfelt thanks to everyone who kept aside their precious time and imparted their knowledge for the success of my project.

I am extremely thankful to **Dr. Jiji C V**, Principal, College of Engineering Trivandrum for providing me with the best facilities and atmosphere which was necessary for the successful completion of this project.

I am extremely grateful to **Smt. Deepa S S**, HOD, Dept of Computer Applications, for providing me the support and encouragement for the project work.

I express my sincere thanks to **Smt. Deepa S S**, Asst. Professor, Department of Computer Applications, College of Engineering Trivandrum for her invaluable guidance, support and advice that aided in the successful completion of my project.

I profusely thank all the faculties of the department and all other staff of CET, for their guidance and encouragement throughout my course of study.

I owe sincere thanks to my friends and all others who have directly or indirectly helped me in the completion of this project. No words can express my humble gratitude to my beloved parents who have been guiding me all through my journey.

**Anupama Regi**

# Abstract

Automated Water Meter Reading - AquaMeter is an ideal solution for manual process of meter reading, updating the server with reading and getting the bill for each customers by the meter readers. At present, meter reading looks simple but it involves many issues like calculation errors in bills, difficulty in computing current consumption and cost, difficulties in reaching remote areas, etc. . . are some of the issues. The proposed system for meter reading will reduce the burden of the employees, avoid human errors and incorrectness in bill generation. This will get the meter reading, update the server with reading units and go for billing. This Android application is used to get the reading from the meter, by capturing the image of the meter using Optical Character Recognition. And also with the use of GPS, location of meter is accessed. Any customer with smart phone and network connectivity can access this application. Provide user friendly method for bill generation and payment. With the use of this automated system most of the manual processes and calculations are eliminated so that the meter readings can easily be collected more accurately. The front-end of the project is implemented through XML while JAVA and FIREBASE provide the backend.

# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
<b>2</b>	<b>Literature Review</b>	<b>2</b>
2.1	Existing System . . . . .	2
2.2	Proposed System . . . . .	3
<b>3</b>	<b>System Study</b>	<b>4</b>
3.1	Problem Definition and Motivation . . . . .	4
3.2	Modules . . . . .	5
3.2.1	Meter Reading Module . . . . .	5
3.2.2	Bill Generation Module . . . . .	5
3.3	Feasibility Study . . . . .	5
3.3.1	Technical feasibility . . . . .	6
3.3.2	Economical feasibility . . . . .	7
3.3.3	Operational Feasibility . . . . .	8
3.3.4	Schedule feasibility . . . . .	8
<b>4</b>	<b>Requirement Analysis</b>	<b>9</b>
4.1	Purpose . . . . .	9
4.1.1	Hardware Requirements . . . . .	9
4.1.2	Software Requirements . . . . .	9
4.1.3	Quality Requirements . . . . .	10
<b>5</b>	<b>Design And Implementation</b>	<b>11</b>
5.1	Overall Design . . . . .	11
5.2	System Design . . . . .	11

5.3	Methodology . . . . .	12
5.3.1	GPS . . . . .	12
5.3.2	OCR . . . . .	13
5.4	Input Design . . . . .	15
5.5	Output Design . . . . .	15
5.6	Data Flow Diagram . . . . .	16
5.7	Block Diagram . . . . .	18
5.8	System Flow Chart . . . . .	19
5.9	Firebase Design . . . . .	20
5.10	Use Case Diagram . . . . .	20
5.11	Screenshots . . . . .	22
<b>6</b>	<b>Coding</b>	<b>29</b>
6.1	XML Pages . . . . .	29
6.1.1	First Page . . . . .	29
6.1.2	Login Page . . . . .	30
6.1.3	User Details Fetching Page . . . . .	32
6.1.4	Capturing and Detecting Image Page . . . . .	35
6.1.5	Bill Calculation Page . . . . .	37
6.1.6	Payment Page . . . . .	41
6.2	Java Pages . . . . .	44
6.2.1	First Page . . . . .	44
6.2.2	Login Page . . . . .	44
6.2.3	User Details Fetching Page . . . . .	52
6.2.4	Capturing and Detecting Image Page . . . . .	53
6.2.5	Bill Calculation Page . . . . .	58
6.2.6	Payment Page . . . . .	61
<b>7</b>	<b>Testing and Implementation</b>	<b>63</b>
7.1	Testing and Types of Testing Used . . . . .	63
7.1.1	Unit Testing . . . . .	64
7.1.2	Integration Testing . . . . .	64
7.1.3	System Testing . . . . .	64

7.1.4	Test Case Description . . . . .	64
<b>8</b>	<b>Results and Discussion</b>	<b>66</b>
8.1	Advantages and Limitations . . . . .	66
8.1.1	Advantages . . . . .	66
8.1.2	Limitations . . . . .	67
<b>9</b>	<b>Conclusion and Future Scope</b>	<b>68</b>
9.1	Conclusion . . . . .	68
9.2	Future Scope . . . . .	69



# List of Figures

5.1	Components of an OCR system . . . . .	14
5.2	Level 0 DFD . . . . .	16
5.3	Level 1 DFD . . . . .	17
5.4	Block Diagram . . . . .	18
5.5	System Flow Chart . . . . .	19
5.6	Firebase Structure . . . . .	20
5.7	Use Case Diagram . . . . .	21
5.8	First Page . . . . .	22
5.9	Login Page . . . . .	22
5.10	Login Page - Validation while not entering Customer Id . . . . .	23
5.11	Login Successful . . . . .	23
5.12	User Details Fetched . . . . .	24
5.13	Image Capturing and Detecting . . . . .	24
5.14	Capturing Image Using Camera . . . . .	25
5.15	Image Detected . . . . .	25
5.16	User Entering Present Data and Bill Generation - Validation for less value . . . .	26
5.17	User Entering Present Data and Bill Generation - Validation for greater value . .	26
5.18	Bill Generated Successfully . . . . .	27
5.19	Pending for Admin Verification . . . . .	27
5.20	Payment . . . . .	28

# Chapter 1

## Introduction

Android based meter reading application is an application that will help you monitor your consumption of water. It is used to get the readings from the meter automatically by simply capturing the image of the meter and then performing the OCR technique which is nothing but “optical character recognition”. The OCR technique is used to identify the character from an image and used this character to get the meter readings. Also by tracing the GPS location the application can access the details of registered consumers. And updating the server with the data. You may have potential savings in your budget if you monitor your consumption and change the way you consume water.

Billing is a critical function of the Water Boards towards getting a meter read. Meter reading, even though it looks simple it involves processes that can give various problems. Calculation errors, delays in system updating and fault tracking issues are the major problems that companies find difficult to find answers for. The current procedure with regard to the billing process for water is not a fully automated system. It involves manual processes from the time the Meter reader starts reading the meter until the system is updated with the current reading. A meter reader visits a house, does the meter reading, and note down the units consumed. Back in the office a data entry officer enters the meter readings into the system manually. This procedure is not up to satisfactory level and has many human errors. However, to meet the problems associated with the major problems related to the Manual Billing process a mobile application is proposed in this project. A meter image is captured from android application. Further OCR technique is applied on the image to extract the meter units. It is a technology that recognizes and captures alphanumeric characters on a computer at high speed.

# Chapter 2

## Literature Review

Meter reading and billing are complex tasks of water supplier and other type of meter reading like electricity board. The current technology of billing process uses manual process of meter reading, updating the server with reading and billing the amount for each customer. In this application I proposed an android application. The proposed system will get the meter reading, update the server with reading units and inform consumers about bill units and amount. Android application is used to get the readings from the meter automatically by simply capturing the image of the meter. An OCR (Optical Character Recognition) technique is used to extract meter units from the image. The meter units are sent to the server and the bill amount is calculated. As a part of my literature review I went through various papers and presentations on this topic. The quick summary of my findings are specified in this chapter

### 2.1 Existing System

The suppliers of water use a manual process for billing purposes because they think that it is an easy process and don't require any skills. These authority cannot invest a huge amount of money for a new solution. However, the customer has to face many problems with the current procedure used by these authority to calculate Bills. According to meter reader there is difficulty in finding all the houses where meters are situated and mostly to identify meters that are located in rural areas. According to customer point of view there is no facility provided by authority of knowing their current consumption units or calculating it manually. There is also no facility to compare the previous month's consumption units with the current month. Customers are facing difficulty in contacting with authority to make any complaints about incorrect bill or device failure. Finally

from authority's point of view, all these process are manually maintained which is a big burden for them. **Problems in Existing System are** Already existing is highly Person dependant which might lead to Human errors. Billing is done mainly on estimated/monthly average basis. As human lab hours are involved Billing cycle requires excessive time.

## 2.2 Proposed System

The development of this application demonstrates the concept and implementation of new, it has a low infrastructure cost, low operating costs, more data security and less man power required. Therefore it not only solves the problem of conventional meter reading but also provides additional feature bill generation on mobile. The inputted image is preprocessed with help of OCR. GPS provide more secure login and also provide security to the application. The result is updated in Database. With help of result of OCR technique the bill is generated.

# Chapter 3

## System Study

### 3.1 Problem Definition and Motivation

AquaMeter is a Automated Water Meter Reading application which has been designed basically to reduce the tedious work and avoid the Human errors in meter reading. The Application is built essentially to monitor the usage, bill payment and accessing the data which can result in better management. The android based meter reading using OCR helps for this and the use of GPS help for accessing the location for security. Android app is for customers for reading the meter and paying the bill. This solution gives best benefits to customers as well as the authority.

The major motivation behind choosing the project was the drawbacks of the existing system. The system that we have now is a complete manual one. And which is followed from the very beginning. The major drawbacks of the existing system are mentioned below,

The present revenue system of Kerala Water Authority

- The process of manual meter reading is time consuming and expensive. .
- The amount of human resource which is spend on the meter reading is quite high.
- Highly person dependant and deficiency of meter readers.
- Billing cycle requires excessive time

This gives the solutions to address the problems related to manual water billing process. The customers also complaint about incorrectness of bill, this is because the assumption of reading when not available and leads to major problem of current technology. Most of the time bill calculation and system updating are done manually. There can be some resulting human errors to frustrated customers who are not satisfied about the service of the Water Authority at all. The suggested system has come up with solutions which address all the problems. With this service the burdens of the Meter reader as well as the Water Authority get lessened and are made more efficient. A mobile solution is given for the Meter reader so that the day to day work becomes less tiresome. Most of the manual processes and calculations are eliminated so that the meter readings can easily be collected more accurately to be updated to the system. On the other hand, the Android customers are also given a mobile solution so that they can have their latest bills and make payments and other important tasks efficiently.

## **3.2 Modules**

There are 2 modules in the proposed system:

### **3.2.1 Meter Reading Module**

The user or the customer needs to login to the system using the GPS location accessing and Customer Id and capture the image using camera and detect the image for meter reading.

### **3.2.2 Bill Generation Module**

The user needs to enter the meter reading once more and if detected and entered values are equal then payment is done.

## **3.3 Feasibility Study**

Depending on the results of the initial investigation the survey is now expanded to a more detailed feasibility study. Feasibility study is a test of system proposal according to its workability, impact of the organization, ability to meet needs and effective use of the resources. It focuses on these major questions:

1. What are the user's demonstrable needs and how does a candidate system meet them?
2. What resources are available for given candidate system?
3. What are the likely impacts of the candidate system on the organization?
4. Whether it is worth to solve the problem?

During feasibility analysis for this project, following primary areas of interest are to be considered.

Steps in feasibility analysis:

Eight steps involved in the feasibility analysis are:

- Form a project team and appoint a project leader.
- Prepare system flowcharts.
- Enumerate potential proposed system.
- Define and identify characteristics of proposed system.
- Determine and evaluate performance and cost effective of each proposed system.
- Weight system performance and cost data.
- Select the best-proposed system.
- Prepare and report final project directive to management.

### **3.3.1 Technical feasibility**

A study of resource availability that may affect the ability to achieve an acceptable system. This evaluation determines whether the technology needed for the proposed system is available or not. This is concerned with specifying equipment and software that will successfully satisfy the user requirement. The technical needs of the system may include: Front-end and back-end selection. An important issue for the development of a project is the selection of suitable front-end and back-end. When I decided to develop the project I went through an extensive study to determine the most suitable platform that suits the development of the project. The aspects of my study included the following factors. Front-end selection:

1. It must have a GUI that assists employees that are not from IT background.

2. Scalability and extensibility.
3. Flexibility.
4. Robustness.
5. Platform independent.
6. Easy to debug and maintain.
7. Event driven programming facility.

Back-end Selection:

1. Multiple user support.
2. Efficient data handling.
3. Provide inherent features for security.
4. Efficient data retrieval and maintenance.
5. Easy to implant with the Front-end.

According to above stated features I selected Java as the backend. The technical feasibility is frequently the most difficult task encountered at this stage. It is essential that the process of analysis and definition be conducted in parallel with an assessment to technical feasibility. It centers on the existing system and to what extent it can support the proposed system.

### **3.3.2 Economical feasibility**

Economic justification is generally the Bottom Line consideration for most systems. Economic justification includes a broad range of concerns that includes cost benefit analysis. In this, the cost and the benefits associated with the proposed system are weighed and if it suits the basic purpose of the project, the project is moved to the analysis and design phase. The financial and the economic questions during the preliminary investigation are verified to estimate the following:

- The cost to conduct a full system investigation.
- The cost of hardware and software for the class of application being considered.



- The proposed system will give the minute information, as a result the performance is improved
- This feasibility checks whether the system can be developed with the available funds.

### 3.3.3 Operational Feasibility

It is mainly related to human organizations and political aspects. The points to be considered are:

- What changes will be brought with the system?
- What organization structures are disturbed?
- What new skills will be required?
- Do the existing staff members have these skills?
- If not, can they be trained in due course of time?

The system is operationally feasible as it very easy for the End users to operate it. It needs basic information about Android application.

### 3.3.4 Schedule feasibility

Time evaluation is the most important consideration in the development of project. The time schedule required for the developed of this project is very important since more development time effect machine time, cost and cause delay in the development .

# Chapter 4

## Requirement Analysis

### 4.1 Purpose

Requirement Specification is the part of the project which gives the details about the hardware and software requirements of our project. It also details the features of the programming language used.

#### 4.1.1 Hardware Requirements

- Processor : Intel Core i3
- Storage : 512 GB Hard Disk space
- Memory : 8 GB RAM

#### 4.1.2 Software Requirements

- Operating System : Windows/Linux
- Back end : Java, Firebase
- Technologies used : OCR, GPS
- Platform used : Android Studio
- Designer tool : XML

### 4.1.3 Quality Requirements

- Scalability : The software will meet all of the functional requirements.
- Maintainability : The system should be maintainable. It should keep backups to at one for system failures, and should log its activities periodically.
- Reliability : The acceptable threshold for down-time should be large as possible. i.e. mean time between failures should be large as possible. And if the system is broken, time required to get the system backup again should be minimum.
- Availability: This system is easily available as the core equipments in building the software is easily obtained.
- High- Functionality: This system is highly functional in all environment since, They are highly adaptable.

# Chapter 5

## Design And Implementation

The proposed system is used to capture the meter image using OCR , process it and extract the details for bill generation and payment. The application also uses GPS for more secure use of the application.

### 5.1 Overall Design

GPS location is accessed and the customer login to the application if and only if the customer id and location of meter matches.After login a meter image is captured from android application. Further OCR technique is applied on the image to extract the meter units. OCR is a system that provides a full alphanumeric recognition of printed or handwritten characters at electronic speed by simply scanning the form. It is a technology that recognizes and captures alphanumeric characters on a computer at high speed.Afterwards the bill is calculated and payment happens.

### 5.2 System Design

Android based meter reading application is an application that will help you monitor your consumption of water. But in fact anything that has a meter can be monitored with Meter Reading. You may have potential savings in your budget if you monitor your consumption and change the way you consume energy.

Android based meter reading Application is used to get the readings from the meter automatically by simply capturing the image of the meter and then performing the OCR technique which is nothing but “optical character recognition”. The OCR technique is used to identify the

character from an image and used this character to get the meter readings.

The android based meter reading using OCR consists of an Android application. Android app is for customers for reading the meter. Before reading the meter the customer needs to login with their Customer Id and GPS location. This solution gives best benefits to customers as well as authority.

All that needs to be done is to capture the image then android app will perform operation of extracting the meter reading text from image and send to the server. Then the application does the calculation and proceed bills are shown to the relevant consumers via application at the same instance. In case if the entered reading and OCR detected characters are not same, an image of that particular meter can be sent to the admin for verification. Using this application a customer can view bill having all details related to any of particular month for easy comparison with previous month consumption. Along with that he also has an option to make payments online.

## 5.3 Methodology

There are five parts in this project. The first part is the login using GPS location and Customer Id. The second one is the capturing of image using camera and detecting the image for meter reading. Then the next part is that the customer needs to enter the meter reading by themselves for system level verification. And at last, if OCR detected and entered values are same then payment is done.

### 5.3.1 GPS

The Global Positioning System (GPS) is a navigation system using satellites, a receiver and algorithms to synchronize location, velocity and time data for air, sea and land travel.

GPS is made up of three different components, called segments, that work together to provide location information. The three segments of GPS are:

**Space (Satellites):** The satellites circling the Earth, transmitting signals to users on geographical position and time of day.

**Ground control :** The Control Segment is made up of Earth-based monitor stations, master control stations and ground antenna. Control activities include tracking and operating the

satellites in space and monitoring transmissions. There are monitoring stations on almost every continent in the world, including North and South America, Africa, Europe, Asia and Australia.

**User equipment :** GPS receivers and transmitters including items like watches, smartphones and telematic devices

### **5.3.2 OCR**

The main process of the Automated Water Meter Reading - AquaMeter is the detection of meter reading by using OCR. The image is pre processed with help of OCR technique. Optical Character Recognition deals with the problem of recognizing optically processed characters. Both hand printed and printed characters may be recognized, but the performance is directly dependent upon the quality of the input documents. OCR is software which takes image as an input and recognizes the text in the given image as an output. For character recognition, offline or online, there are two basic types of core OCR algorithm. Optical Character Recognition (OCR) is a process of converting printed or handwritten scanned documents into ASCII characters that a computer can recognize. In other words, automatic text recognition using OCR is the process of converting an image of textual documents into its digital textual equivalent. Developing an OCR is a very difficult task. The advantage is that the textual material can be edited, which otherwise is not possible in scanned documents in which these are image files. The document image itself can be either machine-printed, image or handwritten, or a combination of the two. The simplified image is given as input to the OCR. It recognizes the devnagari text and passes it to the translator.

The main idea of the proposed system is to extract a meter count from given meter image. This process of automatic meter number recognition is composed of four stages. The stages are as follows:

- 1] Image Acquisition
- 2] Region Extraction
- 3] Segmentation
- 4] Optical Character Recognition (OCR)

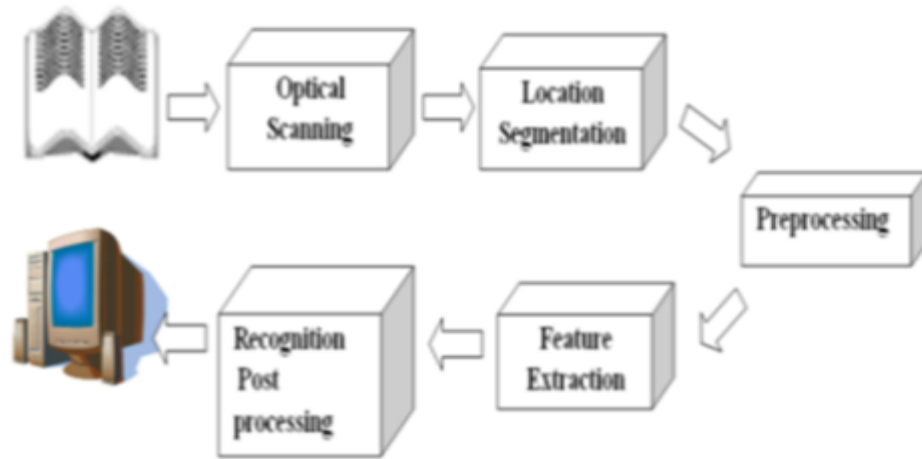


Figure 5.1: Components of an OCR system

**Image Acquisition:** In first stage, the meter image is acquired using a camera. The parameters of the camera, such as the type of camera, camera resolution, orientation, and light, have to be considered.

**Region Extraction:** The second stage is to extract the meter number from the image based on some features. In the proposed system boundary based extraction is performed. The Boundary-based extraction technique uses Hough transform (HT). It detects straight lines in the image to locate the meter number. The Hough transform has the advantage of detecting straight lines with up to  $30^\circ$  inclination.

**Segmentation:** In this stage, the meter image is segmented and the characters are extracted by projecting their color information, labeling them, or matching their positions with templates. In this stage, binarization is performed. Binarization is usually performed in the preprocessing stage of different document image processing related applications such as optical character recognition (OCR) and document image retrieval. It converts a gray-scale document image into a binary document image and accordingly facilitates the ensuing tasks such as document skew estimation and document layout analysis.

**Optical Character Recognition (OCR):** In this stage, the extracted characters are recognized and the output is the meter number. Character recognition in proposed system may have some difficulties. Due to the camera zoom factor, the extracted characters do not have the same size and the same thickness. Resizing the characters into one size before recognition helps overcome

this problem.

## 5.4 Input Design

The input design is the link between the information system and the user. It comprises the developing specification and procedures for data preparation and those steps are necessary to put transaction data into a usable form for processing can be achieved by inspecting the computer to read data from a written or printed document or it can occur by having people keying the data directly into the system. The design of input focuses on controlling the amount of input required, controlling the errors, avoiding delay, avoiding extra steps and keeping the process simple. The input is designed in such a way so that it provides security and ease of use with retaining the privacy. Input Design considered the following things:

- What data should be given as input?
- How the data should be arranged or coded?
- The dialog to guide the operating personnel in providing input.
- Methods for preparing input validations and steps to follow when error occur.

## 5.5 Output Design

A quality output is one, which meets the requirements of the end user and presents the information clearly. In any system results of processing are communicated to the users and to other system through outputs. In output design it is determined how the information is to be displaced for immediate need and also the hard copy output. It is the most important and direct source information to the user. Efficient and intelligent output design improves the system's relationship to help user decision-making.

- Designing computer output should proceed in an organized, well thought out manner; the right output must be developed while ensuring that each output element is designed so that people will find the system can use easily and effectively. When analysing and designing computer output, they should identify the specific output that is needed to meet the requirements.



- Select methods for presenting information.
- Create document, report, or other formats that contain information produced by the system.

## 5.6 Data Flow Diagram

DFD is one of the graphical representation techniques used in a project to show the flow of the data through a project. DFD helps us to obtain an idea about the input, output, and process involved. The things absent in a DFD are control flow, decision rules, and loops. It can be described as a representation of functions, processes that capture, manipulate, store, and distribute data between a system and the surrounding and between the components of the system. The visual representation helps for good communication.

It shows the journey of the data and how will it be stored in the last. It does not provide details about the process timings or if the process shall have a parallel or sequential operation. It is very different from a traditional flow chart or a UML that shows the control flow or the data flow.

In level 0 the basic data flow of the application is showcased. It does not show the flow of data much deeper. It will be evaluated in the higher levels of Data Flow Diagram. The Data Flow Diagram of Automated Water Meter Reading - AquaMeter is shown below.

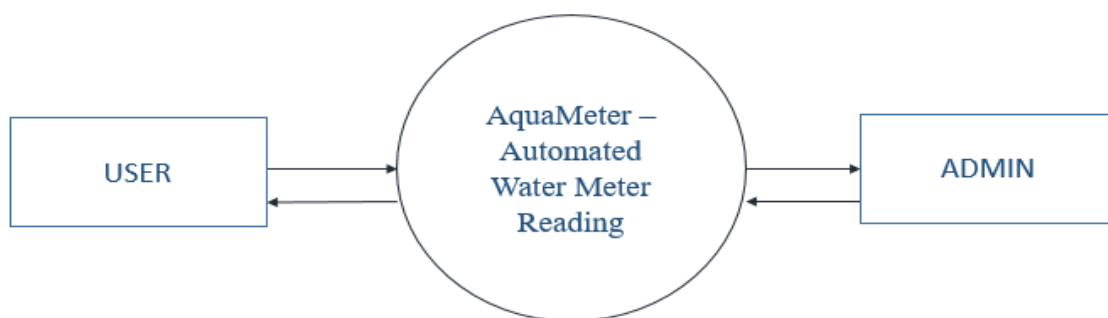


Figure 5.2: Level 0 DFD

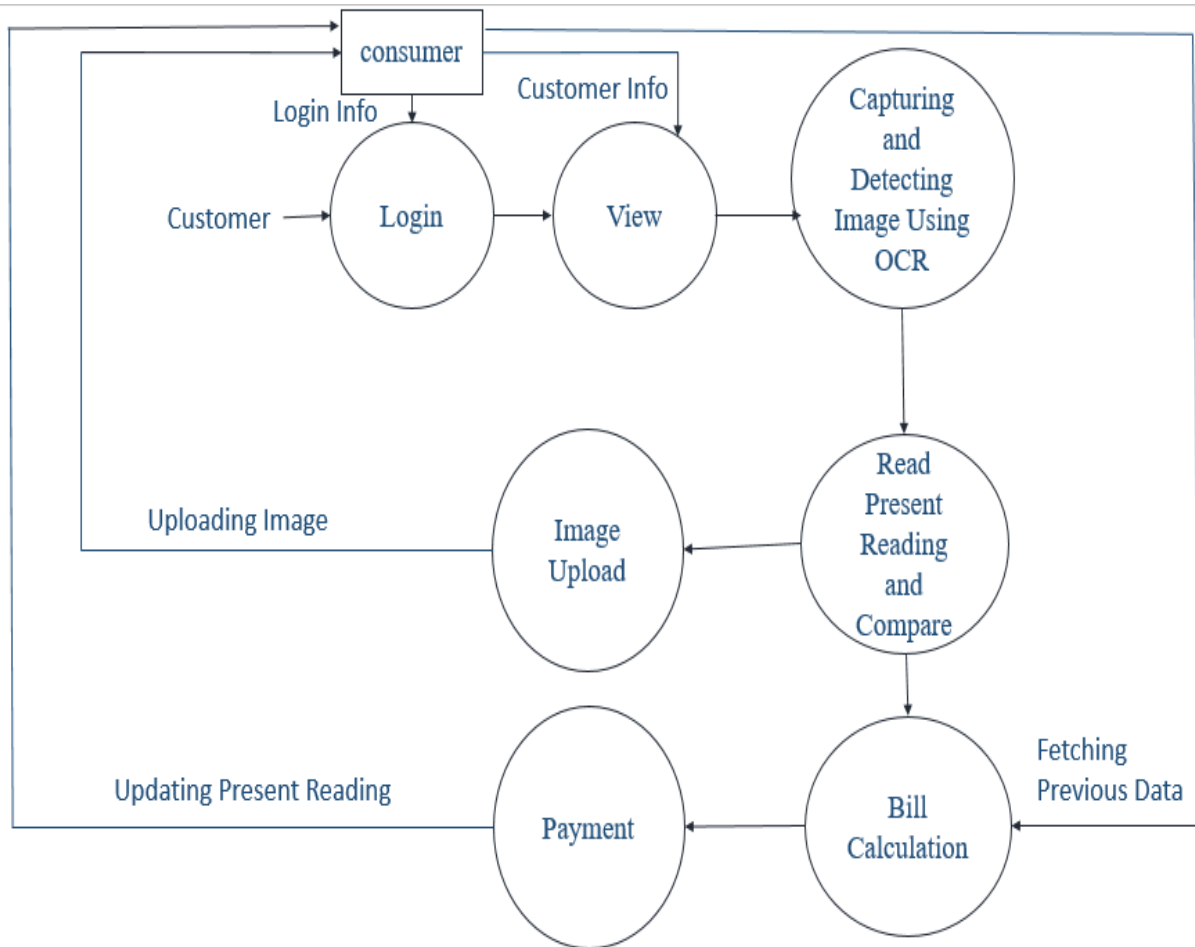


Figure 5.3: Level 1 DFD

## 5.7 Block Diagram

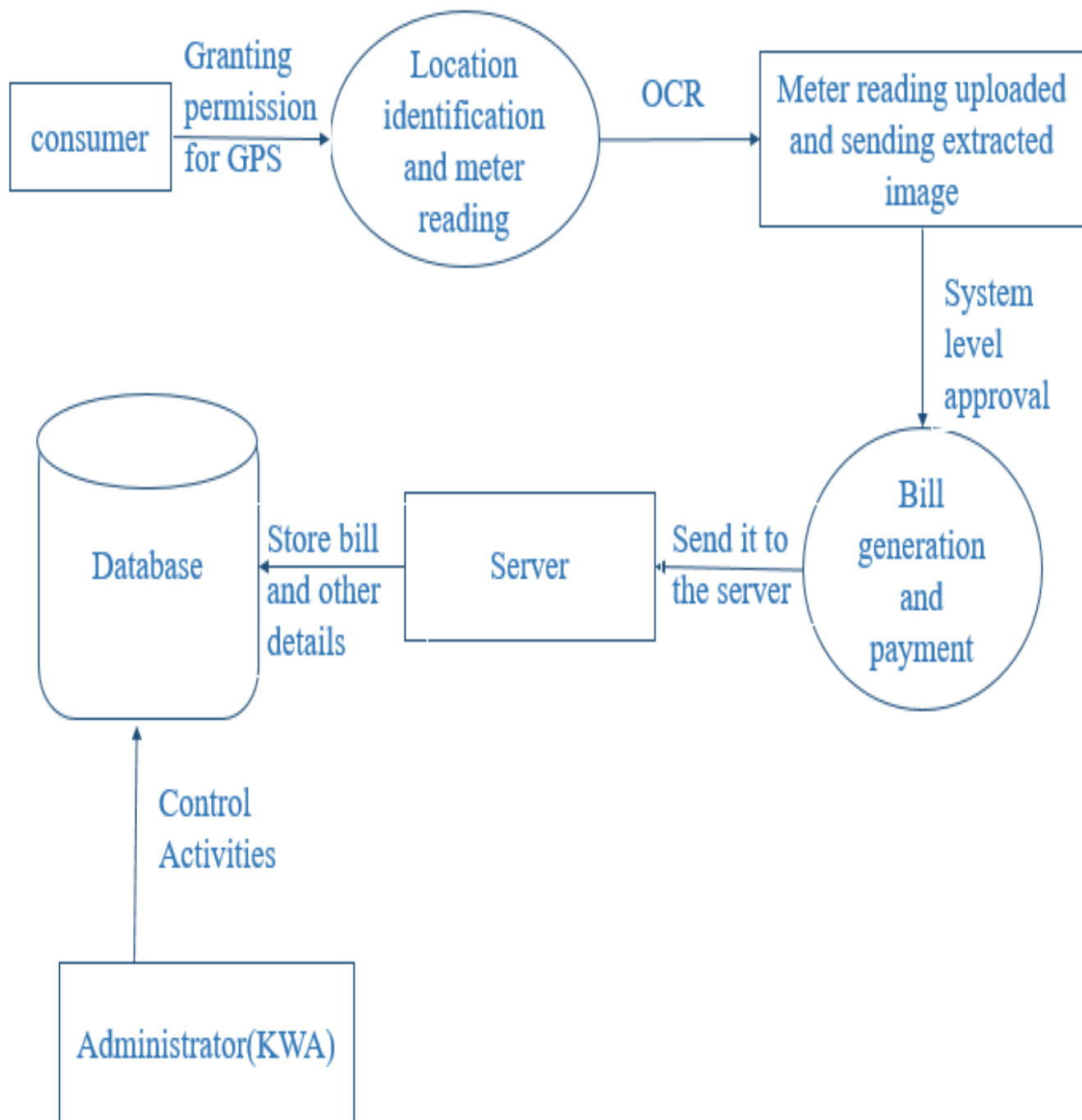


Figure 5.4: Block Diagram

## 5.8 System Flow Chart

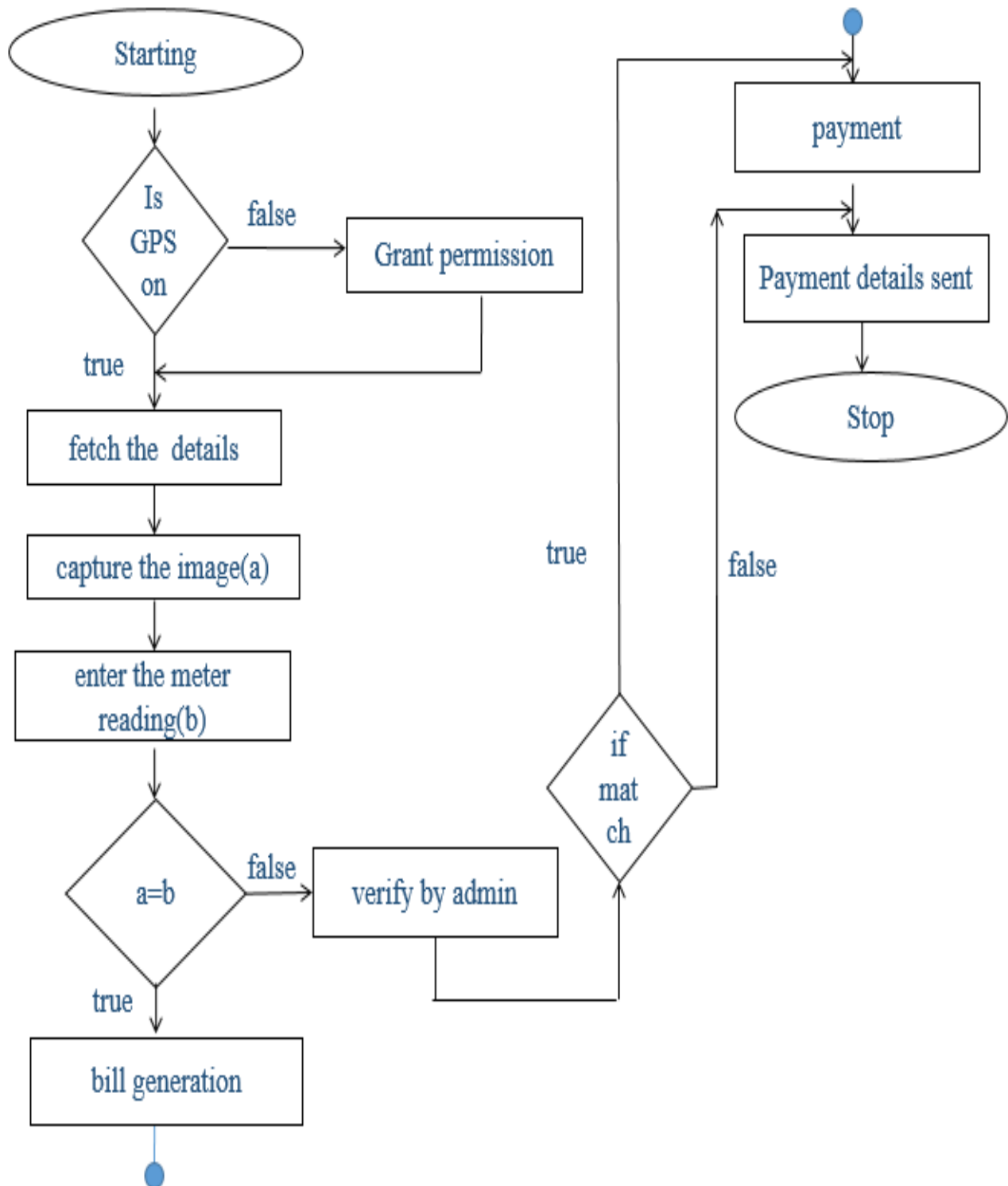


Figure 5.5: System Flow Chart

## 5.9 Firebase Design

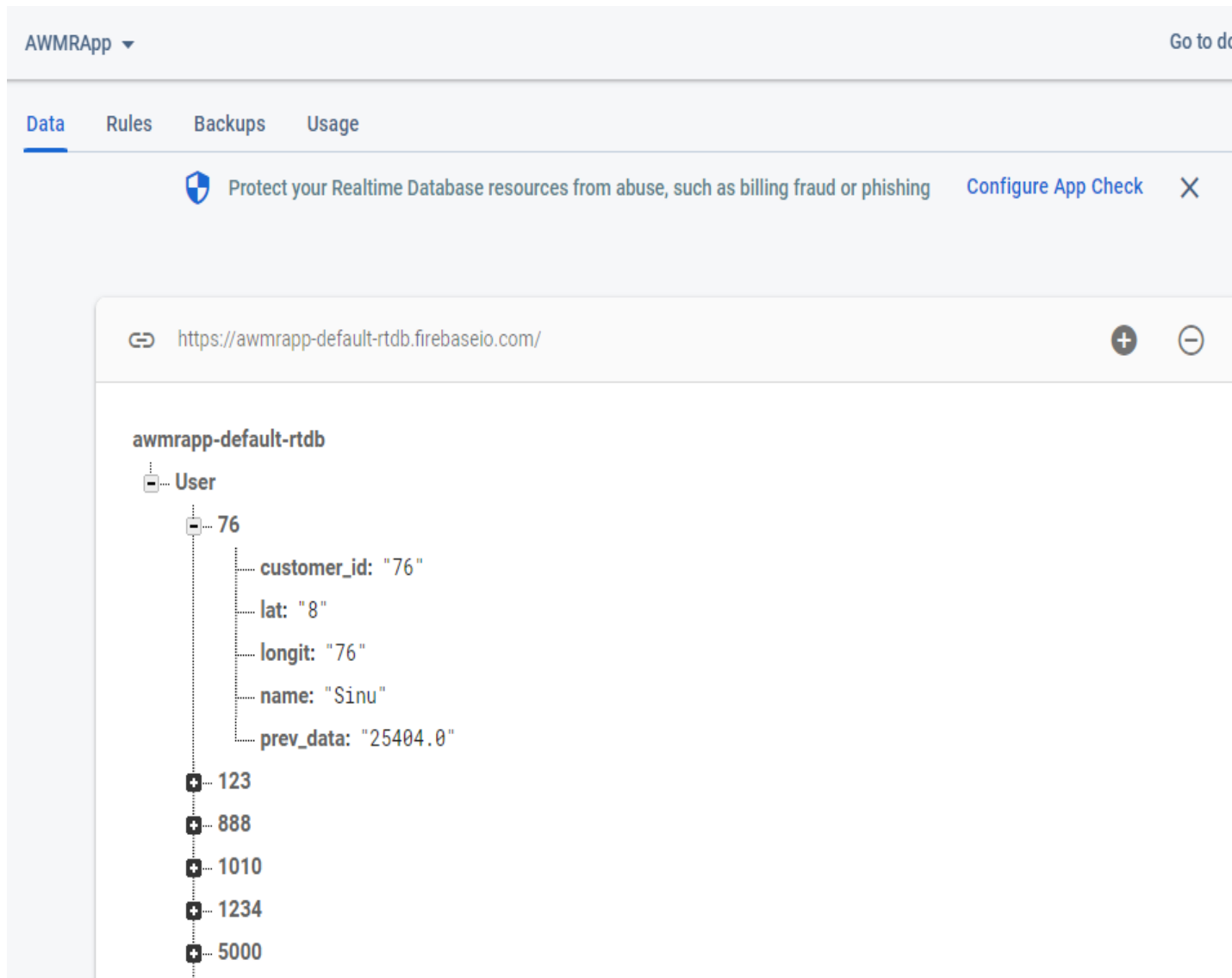


Figure 5.6: Firebase Structure

## 5.10 Use Case Diagram

The use case model for any system consists of a set of ‘use cases’. Intuitively, use cases represent the different ways in which a system can be used by the users. The use case corresponds to the high level requirements. The use case does not mention any specific algorithm to be used, nor the internal data representation, internal structure of the software, etc. A use case typically represents a sequence of interactions between the user and the system. These interactions consist

of one main line sequence .The main line sequence represents the normal interaction between the user and the system .Several variations to the mainline sequence may also exist .Typically a variation from the mainline sequence occur when some specific conditions hold.

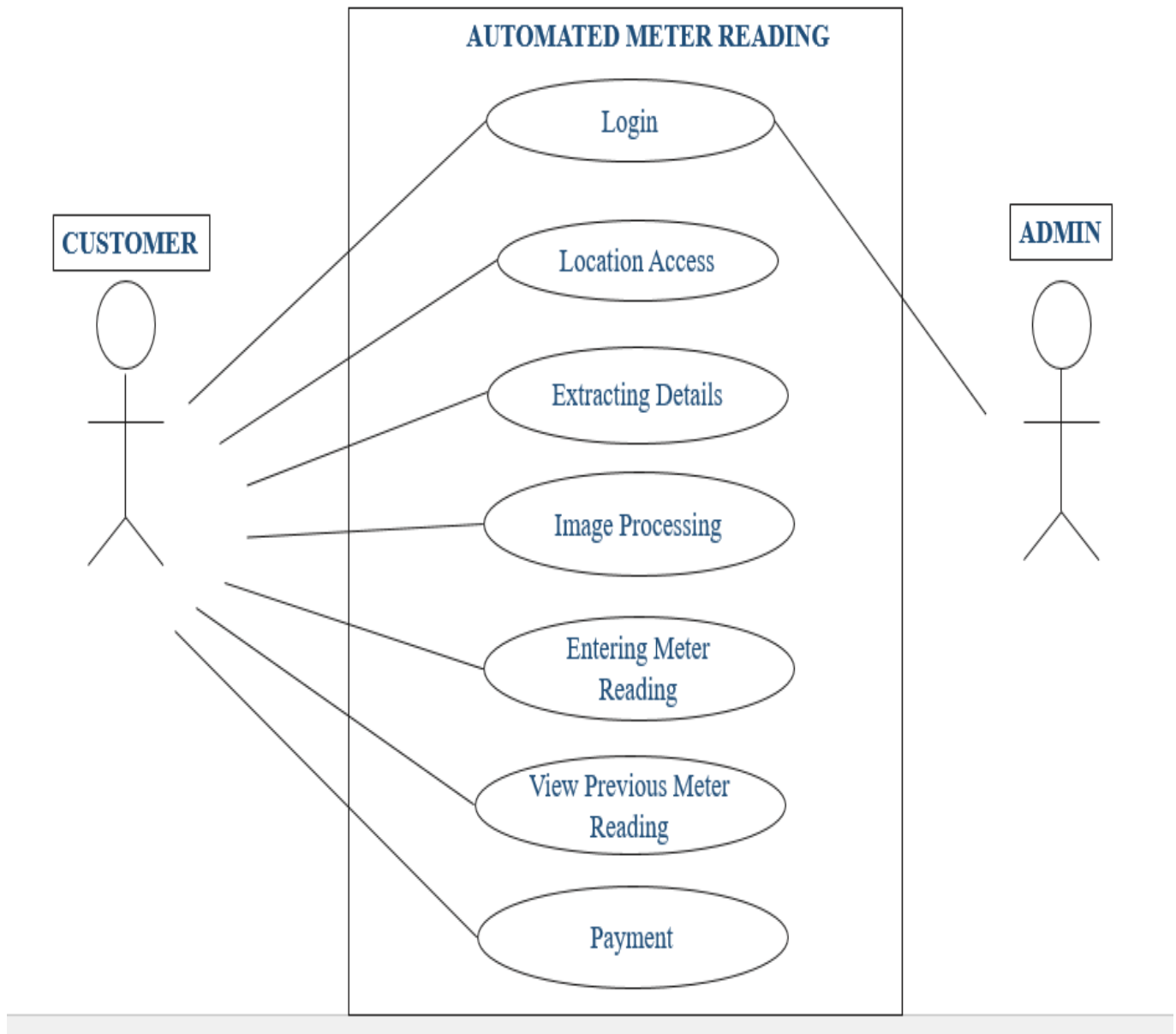


Figure 5.7: Use Case Diagram

## 5.11 Screenshots



Figure 5.8: First Page

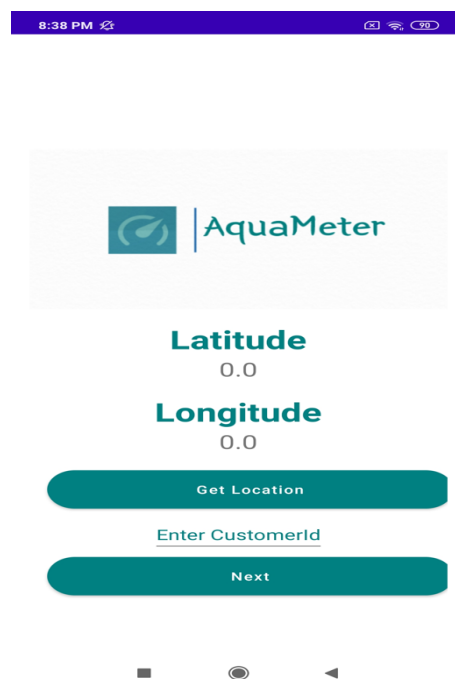


Figure 5.9: Login Page

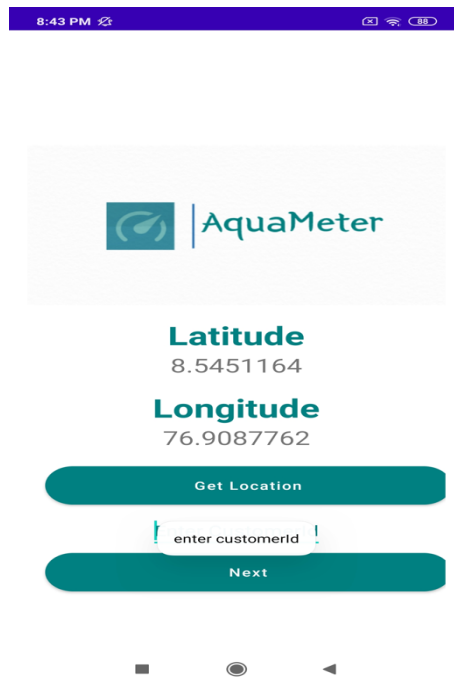


Figure 5.10: Login Page - Validation while not entering Customer Id

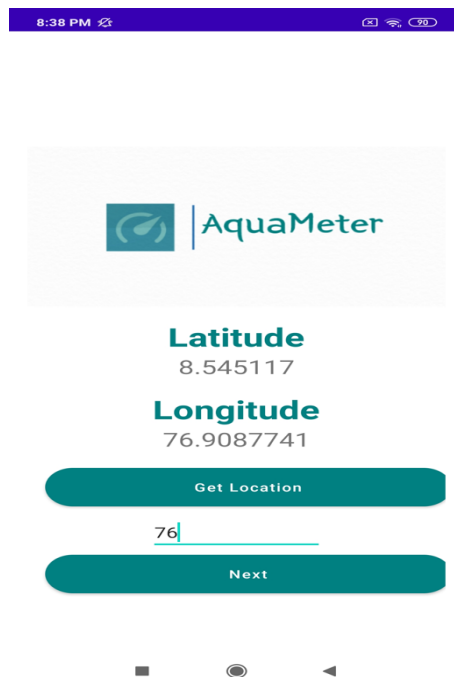


Figure 5.11: Login Successful



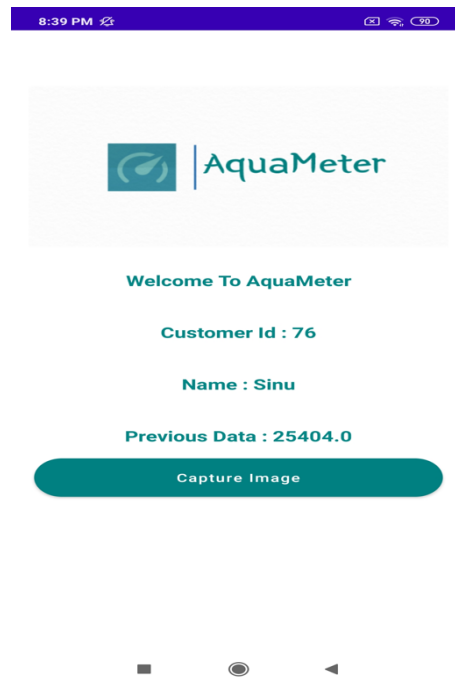


Figure 5.12: User Details Fetched

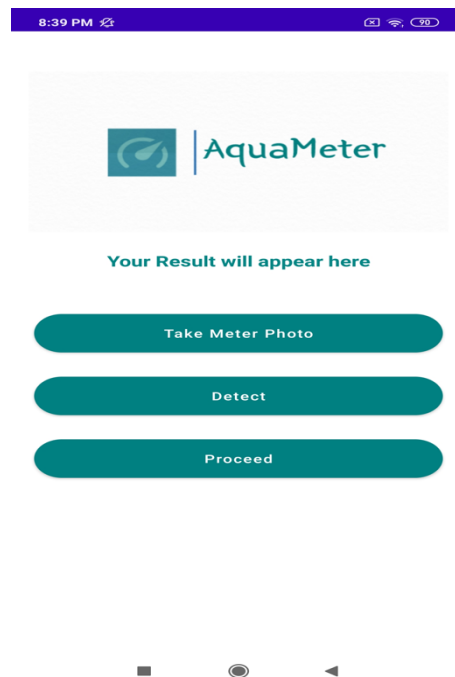


Figure 5.13: Image Capturing and Detecting

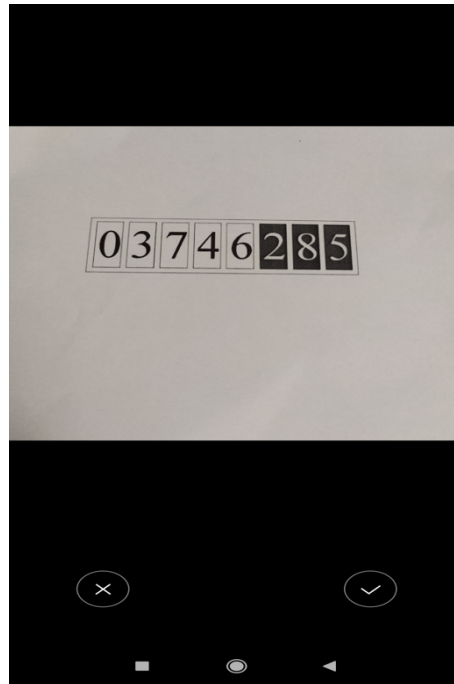


Figure 5.14: Capturing Image Using Camera



Figure 5.15: Image Detected

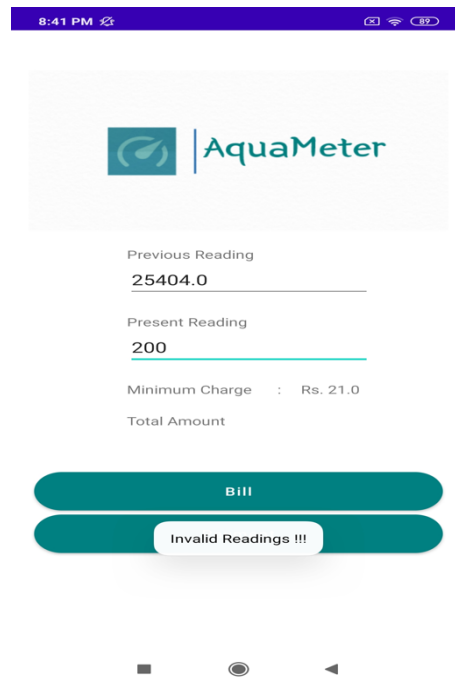


Figure 5.16: User Entering Present Data and Bill Generation - Validation for less value

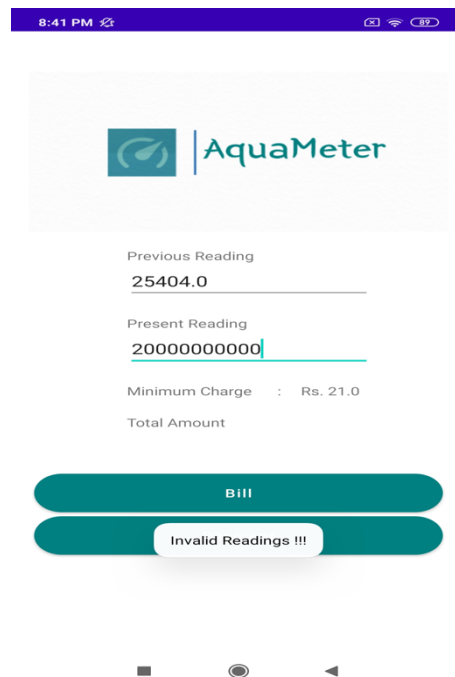


Figure 5.17: User Entering Present Data and Bill Generation - Validation for greater value

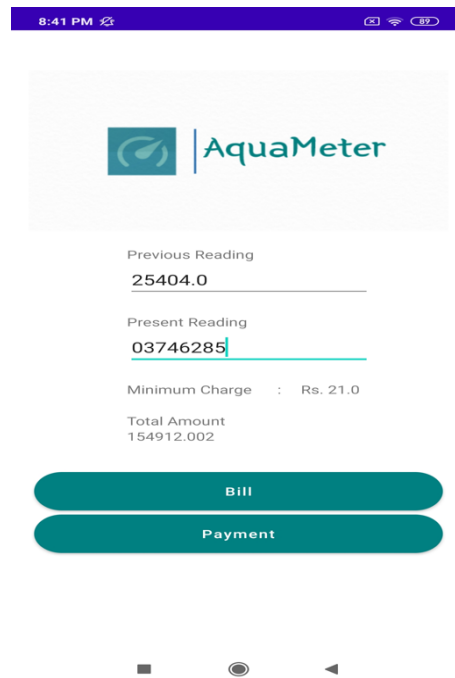


Figure 5.18: Bill Generated Successfully

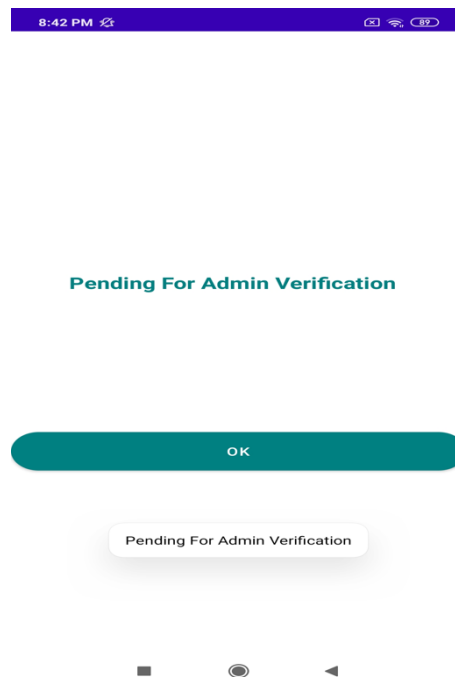
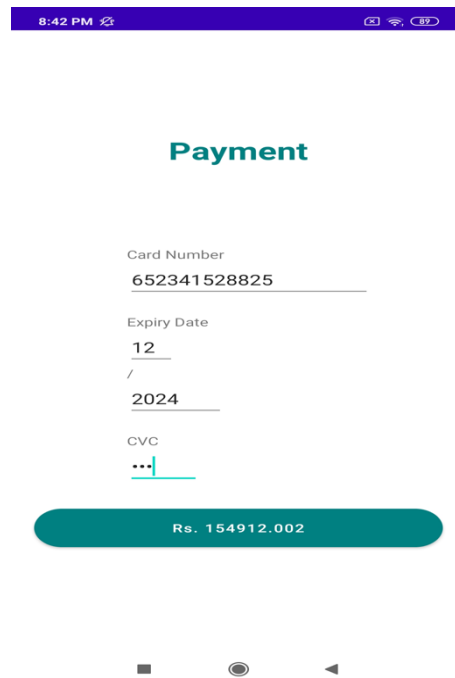


Figure 5.19: Pending for Admin Verification



A screenshot of a mobile application's payment screen. At the top, a purple status bar shows the time 8:42 PM and various icons. The main heading is "Payment" in a bold, teal font. Below it, there are three input fields: "Card Number" with the value "652341528825", "Expiry Date" with "12 / 2024", and "CVC" with three dots. A teal button with the text "Rs. 154912.002" is positioned below the inputs. At the bottom, there are three small, dark icons: a square, a circle, and a triangle.

Figure 5.20: Payment

# Chapter 6

## Coding

### 6.1 XML Pages

#### 6.1.1 First Page

---

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout 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"
    android:background="@color/white"
    tools:context=".Login_page">
    <ImageView
        android:id="@+id/idIVLogo"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:layout_alignParentTop="true"
        android:layout_centerHorizontal="true"
        android:layout_marginTop="70dp"
        android:src="@drawable/logo" />
</RelativeLayout>
```

### 6.1.2 Login Page

---

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout 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"
    android:orientation="vertical"
    android:padding="16dp"
    android:gravity="center"
    tools:context=".MainActivity">
    <ImageView
        android:id="@+id/idfirstLogo"
        android:layout_width="wrap_content"
        android:layout_height="200dp"
        android:layout_centerHorizontal="true"
        android:layout_marginTop="70dp"
        android:src="@drawable/logo" />
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Latitude"
        android:textSize="32sp"
        android:textStyle="bold"
        android:textColor="#008080"
        android:layout_marginTop="16dp" />
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:id="@+id/tv_latitude"
```

```
        android:text="0.0"
        android:textSize="24sp"/>
<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Longitude"
    android:textSize="32sp"
    android:textStyle="bold"
    android:textColor="#008080"
    android:layout_marginTop="16dp" />
<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:id="@+id/tv_longitude"
    android:text="0.0"
    android:textSize="24sp"/>
<Button
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="Get Location"
    android:id="@+id/bt_location"
    android:layout_marginStart="10dp"
    android:layout_marginTop="20dp"
    android:layout_marginEnd="0dp"
    android:background="@drawable/button_back"
    android:textAllCaps="false"
    app:backgroundTint="@color/teal" />
<EditText
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:hint="Enter CustomerId"
    android:textColorHint="#008080"
    android:id="@+id/cuid"
    android:layout_marginTop="10dp"
```



```
    />
    <Button
        android:id="@+id/BtnNext1"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginStart="10dp"
        android:layout_marginTop="5dp"
        android:layout_marginEnd="0dp"
        android:background="@drawable/button_back"
        android:text="Next"
        android:textAllCaps="false"
        app:backgroundTint="@color/teal" />
</LinearLayout>
```

---

### 6.1.3 User Details Fetching Page

---

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout 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"
    android:background="@color/white"
    tools:context=".MainActivity_ocr1">
    <ImageView
        android:layout_width="wrap_content"
        android:layout_height="200dp"
        android:layout_centerHorizontal="true"
        android:src="@drawable/logo"
        android:layout_marginTop="70dp"
        android:id="@+id/idIVLogo"/>
    <TextView
        android:id="@+id/idTVHead"
```

---

```
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_below="@+id/idIVLogo"
        android:layout_centerHorizontal="true"
        android:layout_marginStart="20dp"
        android:layout_marginTop="30dp"
        android:layout_marginEnd="20dp"
        android:layout_marginBottom="10dp"
        android:gravity="center"
        android:text="Welcome To AquaMeter"
        android:textAlignment="center"
        android:textColor="@color/teal_700"
        android:textSize="18sp"
        android:textStyle="bold" />
<TextView
    android:id="@+id/idcustomer"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_below="@+id/idTVHead"
    android:layout_centerHorizontal="true"
    android:layout_marginStart="20dp"
    android:layout_marginTop="30dp"
    android:layout_marginEnd="20dp"
    android:layout_marginBottom="10dp"
    android:gravity="center"
    android:text="Customer Id"
    android:textAlignment="center"
    android:textColor="@color/teal_700"
    android:textSize="18sp"
    android:textStyle="bold" />
<TextView
    android:id="@+id/idname"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
```

```
        android:layout_below="@+id/idcustomer"
        android:layout_centerHorizontal="true"
        android:layout_marginStart="20dp"
        android:layout_marginTop="30dp"
        android:layout_marginEnd="20dp"
        android:layout_marginBottom="10dp"
        android:gravity="center"
        android:text="Name"
        android:textAlignment="center"
        android:textColor="@color/teal_700"
        android:textSize="18sp"
        android:textStyle="bold" />
<TextView
    android:id="@+id/idprevious"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_below="@+id/idname"
    android:layout_centerHorizontal="true"
    android:layout_marginStart="20dp"
    android:layout_marginTop="30dp"
    android:layout_marginEnd="20dp"
    android:layout_marginBottom="10dp"
    android:gravity="center"
    android:text="Previous Data"
    android:textAlignment="center"
    android:textColor="@color/teal_700"
    android:textSize="18sp"
    android:textStyle="bold" />
<Button
    android:id="@+id/idBtnCapture"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_below="@+id/idTVHead"
    android:layout_marginStart="20dp"
```

```
        android:layout_marginTop="200dp"
        android:layout_marginEnd="20dp"
        android:background="@drawable/button_back"
        android:text="Capture Image"
        android:textAllCaps="false"
        app:backgroundTint="@color/teal" />
</RelativeLayout>
```

---

### 6.1.4 Capturing and Detecting Image Page

---

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout 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"
    android:background="@color/white"
    tools:context=".ScannerActivity">
    <ImageView
        android:layout_width="wrap_content"
        android:layout_height="200dp"
        android:layout_centerHorizontal="true"
        android:layout_marginTop="50dp"
        android:src="@drawable/logo"
        android:id="@+id/idIVCaptureImage"/>
    <TextView
        android:id="@+id/idTVDetectedText"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_below="@id/idIVCaptureImage"
        android:layout_marginStart="20dp"
        android:layout_marginTop="20dp"
        android:layout_marginEnd="20dp"
```

```
        android:layout_marginBottom="20dp"
        android:gravity="center"
        android:padding="4dp"
        android:text="Your Result will appear here"
        android:textAlignment="center"
        android:textColor="@color/teal_700"
        android:textSize="18sp"
        android:textStyle="bold" />
<Button
    android:id="@+id/idBtnSnap"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_below="@+id/idTVDetectedText"
    android:layout_marginStart="20dp"
    android:layout_marginTop="30dp"
    android:layout_marginEnd="20dp"
    android:background="@drawable/button_back"
    android:text="Take Meter Photo"
    android:textAllCaps="false"
    app:backgroundTint="@color/teal" />
<Button
    android:id="@+id/idBtnDetect"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_below="@+id/idBtnSnap"
    android:layout_marginStart="20dp"
    android:layout_marginTop="30dp"
    android:layout_marginEnd="20dp"
    android:background="@drawable/button_back"
    android:text="Detect"
    android:textAllCaps="false"
    app:backgroundTint="@color/teal" />
<Button
    android:id="@+id/BtnNext2"
```

```
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_below="@+id/idBtnDetect"
        android:layout_marginStart="20dp"
        android:layout_marginTop="30dp"
        android:layout_marginEnd="20dp"
        android:background="@drawable/button_back"
        android:text="Proceed"
        android:textAllCaps="false"
        app:backgroundTint="@color/teal" />
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:id="@+id/s1"/>
</RelativeLayout>
```

---

### 6.1.5 Bill Calculation Page

---

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout 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=".UserDetails"
    android:orientation="vertical">
    <ImageView
        android:layout_width="wrap_content"
        android:layout_height="200dp"
        android:layout_centerHorizontal="true"
        android:layout_marginTop="50dp"
        android:layout_gravity="center"
        android:src="@drawable/logo"
```

---

```
    android:id="@+id/logo"/>
    <TextView
        android:id="@+id/previous"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginLeft="100dp"
        android:layout_marginTop="20dp"
        android:text="Previous Reading"/>
    <EditText
        android:id="@+id/prevreading"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginLeft="100dp"
        android:editable="false"
        android:ems="10"
        tools:ignore="Deprecated" />
    <View
        android:layout_width="match_parent"
        android:layout_height="20dp"
        android:background="#ffffff"/>
    <TextView
        android:id="@+id/present"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Present Reading"
        android:layout_marginLeft="100dp" />
    <EditText
        android:id="@+id/presreading"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginLeft="100dp"
        android:inputType="number"
        android:ems="10" />
    <View
```

```
        android:layout_width="wrap_content"
        android:layout_height="20dp"
        android:background="#ffffff"/>
<RelativeLayout
    android:layout_width="wrap_content"
    android:layout_height="wrap_content">
    <TextView
        android:id="@+id/rate"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Minimum Charge   :"
        android:layout_marginLeft="100dp" />
    <TextView
        android:id="@+id/ratevalue"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginLeft="250dp"
        android:ems="10"
        android:text="Rs. 21.0" />
</RelativeLayout>
<View
    android:layout_width="match_parent"
    android:layout_height="20dp"
    android:background="#ffffff"/>
<TextView
    android:id="@+id/total"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Total Amount"
    android:layout_marginLeft="100dp" />
<TextView
    android:id="@+id/totalvalue"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
```



```
        android:layout_marginLeft="100dp"
        android:ems="10" />
<View
    android:layout_width="match_parent"
    android:layout_height="5dp"
    android:background="#ffffff"/>
<Button
    android:id="@+id/BtnNext3"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_below="@+id/idBtnDetect"
    android:layout_marginStart="20dp"
    android:layout_marginTop="30dp"
    android:layout_marginEnd="20dp"
    android:background="@drawable/button_back"
    android:text="Bill"
    android:textAllCaps="false"
    app:backgroundTint="@color/teal" />
<Button
    android:id="@+id/BtnPay"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_below="@+id/BtnNext3"
    android:layout_marginStart="20dp"
    android:layout_marginTop="5dp"
    android:layout_marginEnd="20dp"
    android:background="@drawable/button_back"
    android:text="Payment"
    android:textAllCaps="false"
    app:backgroundTint="@color/teal" />
</LinearLayout>
```

---

### 6.1.6 Payment Page

---

```
<?xml version="1.0" encoding="utf-8"?>

<LinearLayout 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=".UserDetails"
    android:orientation="vertical">

    <TextView
        android:id="@+id/tv1"
        android:layout_width="233dp"
        android:layout_height="200dp"
        android:layout_centerHorizontal="true"
        android:layout_gravity="center"
        android:layout_marginTop="50dp"
        android:gravity="center"
        android:textSize="30dp"
        android:textColor="@color/teal"
        android:textStyle="bold"
        android:text="Payment" />

    <TextView
        android:id="@+id/carnotv"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginLeft="100dp"
        android:layout_marginTop="20dp"
        android:text="Card Number"/>

    <EditText
        android:id="@+id/cardno"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
```

```
        android:layout_marginLeft="100dp"
        android:inputType="number"
        android:ems="10" />
<View
    android:layout_width="match_parent"
    android:layout_height="20dp"
    android:background="#ffffff"/>
<TextView
    android:id="@+id/ExpiryDate"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Expiry Date"
    android:layout_marginLeft="100dp" />
<EditText
    android:id="@+id/date"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginLeft="100dp"
    android:inputType="date"
    android:ems="2" />
<TextView
    android:id="@+id/slash"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginLeft="100dp"
    android:text="/" />
<EditText
    android:id="@+id/year"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:inputType="date"
    android:layout_marginLeft="100dp"
    android:ems="4" />
<View
```

```
        android:layout_width="wrap_content"
        android:layout_height="20dp"
        android:background="#ffffff"/>
    <TextView
        android:id="@+id/cvctv"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="CVC"
        android:layout_marginLeft="100dp" />
    <EditText
        android:id="@+id/cvc"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginLeft="100dp"
        android:inputType="numberPassword"
        android:ems="3" />
    <View
        android:layout_width="match_parent"
        android:layout_height="20dp"
        android:background="#ffffff"/>
    <Button
        android:id="@+id/PayBill"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_below="@+id/payBill"
        android:layout_marginStart="20dp"
        android:layout_marginTop="10dp"
        android:layout_marginEnd="20dp"
        android:background="@drawable/button_back"
        android:text="Rs. ---"
        android:textAllCaps="false"
        app:backgroundTint="@color/teal" />
</LinearLayout>
```

---

## 6.2 Java Pages

### 6.2.1 First Page

---

```
package com.example.awmrapp;

import androidx.appcompat.app.AppCompatActivity;
import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.ImageView;
import android.widget.TextView;

public class Login_page extends AppCompatActivity {
    ImageView idIVLogo1;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_login_page);
        idIVLogo1=(ImageView) findViewById(R.id.idIVLogo);
        idIVLogo1.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                Intent i=new Intent(Login_page.this,MainActivity.class);
                startActivity(i);
            }
        });
    }
}
```

---

### 6.2.2 Login Page

---

```
package com.example.awmrapp;

import androidx.annotation.NonNull;
```

---

```
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.app.ActivityCompat;
import com.google.android.gms.location.LocationRequest;
import android.Manifest;
import android.annotation.SuppressLint;
import android.content.Context;
import android.content.Intent;
import android.content.pm.PackageManager;
import android.location.Location;
import android.location.LocationManager;
//import android.location.LocationRequest;
import android.os.Bundle;
import android.os.Looper;
import android.provider.Settings;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.ImageView;
import android.widget.TextView;
import android.widget.Toast;
import com.google.android.gms.location.FusedLocationProviderClient;
import com.google.android.gms.location.LocationCallback;
import com.google.android.gms.location.LocationResult;
import com.google.android.gms.location.LocationServices;
import com.google.android.gms.tasks.OnCompleteListener;
import com.google.android.gms.tasks.OnSuccessListener;
import com.google.android.gms.tasks.Task;
import com.google.firebase.auth.FirebaseAuthException;
import com.google.firebase.database.DataSnapshot;
import com.google.firebase.database.DatabaseError;
import com.google.firebase.database.DatabaseReference;
import com.google.firebase.database.FirebaseDatabase;
import com.google.firebase.database.Query;
import com.google.firebase.database.ValueEventListener;
```

```
import java.security.acl.Owner;

public class MainActivity extends AppCompatActivity {

    //Initialize variable

    Button btLocation,btnnext;

    TextView tvLatitude,tvLongitude;

    FirebaseDatabase firebaseDatabase;

    DatabaseReference reference2;

    User user;

    ImageView img;

    FusedLocationProviderClient fusedLocationProviderClient;

    String lati;

    String lat,lon;

    float flat,flon;

    EditText cuId;

    @Override

    protected void onCreate(Bundle savedInstanceState) {

        super.onCreate(savedInstanceState);

        setContentView(R.layout.activity_main);

        //Assign Variable

        btLocation=findViewById(R.id.bt_location);

        tvLatitude=findViewById(R.id.tv_latitude);

        tvLongitude=findViewById(R.id.tv_longitude);

        btnnext=findViewById(R.id.BtnNext1);

        img=(ImageView)findViewById(R.id.idfirstLogo);

        cuId=(EditText) findViewById(R.id.cuid);

        user=new User();

        reference2=FirebaseDatabase.getInstance().getReference().child("User");

        img.setOnClickListener(new View.OnClickListener() {

            @Override

            public void onClick(View view) {

                // Intent i=new Intent(getApplicationContext(),user_register.class);

                // startActivity(i);

                Toast.makeText(getApplicationContext(), "App logo",

                    Toast.LENGTH_SHORT).show();

            }

        });

    }

}
```

```
    }  
});  
btnnext.setOnClickListener(new View.OnClickListener() {  
    @Override  
    public void onClick(View v) {  
        String slat=tvLatitude.getText().toString();  
        Double dlat=Double.parseDouble(slat);  
        String slong=tvLongitude.getText().toString();  
        Double dlong=Double.parseDouble(slong);  
        String scuid=cuId.getText().toString();  
        if (scuid.isEmpty()) {  
            Toast.makeText(getApplicationContext(), "enter customerId",  
                Toast.LENGTH_SHORT).show();  
        }  
        else if (scuid.equals("admin123"))  
        {  
            Toast.makeText(getApplicationContext(), "going to adminPage",  
                Toast.LENGTH_SHORT).show();  
            Intent i=new Intent(getApplicationContext(),user_register.class);  
            startActivity(i);  
        }  
        else  
        {  
            Query query =  
                reference2.orderByChild("customer_id").equalTo(scuid);  
            query.addListenerForSingleValueEvent(new ValueEventListener() {  
                @Override  
                public void onDataChange(@NonNull DataSnapshot dataSnapshot) {  
                    if (dataSnapshot.exists()) {  
                        for (DataSnapshot snapshot :  
                            dataSnapshot.getChildren()) {  
                            user = snapshot.getValue(User.class);  
                        }  
                    }  
                }  
            });  
        }  
    }  
});
```



```
        Double
            dUserLat=Double.parseDouble(user.getLat().toString());
        if (dlat<(dUserLat+10.0)&&dlat>(dUserLat-10.0)) {
            Toast.makeText(getApplicationContext(), "logged
            successfully", Toast.LENGTH_SHORT).show();
            Intent i=new
                Intent(MainActivity.this,MainActivity_ocr1.class);
            i.putExtra("id",user.getCustomer_id());
            i.putExtra("name",user.getName());
            i.putExtra("previous",user.getPrev_data());
            startActivity(i);} else {
                Toast.makeText(getApplicationContext(),
                    "location mismatch",
                    Toast.LENGTH_SHORT).show();
            }
        }
    } else {
        Toast.makeText(getApplicationContext(), "no account in
        this Id", Toast.LENGTH_SHORT).show();
    }
}

@Override
public void onCancelled(@NonNull DatabaseError databaseError) {
    Toast.makeText(getApplicationContext(), "error",
        Toast.LENGTH_SHORT).show();
}

});
}

}

});

fusedLocationProviderClient= LocationServices.getFusedLocationProviderClient(
    MainActivity.this
);

btLocation.setOnClickListener(new View.OnClickListener() {
```

```
@Override
public void onClick(View view) {
    //Check condition
    if(ActivityCompat.checkSelfPermission(MainActivity.this
        , Manifest.permission.ACCESS_FINE_LOCATION) ==
        PackageManager.PERMISSION_GRANTED
        &&
        ActivityCompat.checkSelfPermission(MainActivity.this,Manifest.permission.AC
        == PackageManager.PERMISSION_GRANTED){
        //When both permission are granted
        //Call method
        getCurrentLocation();
    }else {
        //When permission is not granted
        //Request permission
        ActivityCompat.requestPermissions(MainActivity.this
            ,new String[]{Manifest.permission.ACCESS_FINE_LOCATION
            ,Manifest.permission.ACCESS_COARSE_LOCATION}
            ,100);
    }
}

});

}

@SuppressLint("MissingSuperCall")
@Override
public void onRequestPermissionsResult(int requestCode, @NonNull String[]
    permissions, @NonNull int[] grantResults) {
    //Check condition
    if (requestCode == 100 && grantResults.length > 0 && (grantResults[0] +
        grantResults[1]
        == PackageManager.PERMISSION_GRANTED)) {
        //When permission granted
        //Call method
        getCurrentLocation();
    }
}
```

```
    }else
    {
        //When permission are denied
        //Display toast
        Toast.makeText(getApplicationContext(), "Permission Denied !!!",
            Toast.LENGTH_SHORT).show();
    }
}

@SuppressLint("MissingPermission")
private void getCurrentLocation()
{
    //Initialize location manager
    LocationManager locationManager=(LocationManager) getSystemService(
        Context.LOCATION_SERVICE
    );
    //Check condition
    if(locationManager.isProviderEnabled(LocationManager.GPS_PROVIDER)
    || locationManager.isProviderEnabled(LocationManager.NETWORK_PROVIDER)){
        //When location service is enabled
        //Get last location
        fusedLocationProviderClient.getLastLocation().addOnCompleteListener(new
            OnCompleteListener<Location>() {
                @Override
                public void onComplete(@NonNull Task<Location> task) {
                    //Initialize location
                    Location location = task.getResult();
                    //Check condition
                    if (location!= null){
                        //When location result is not null
                        //Set latitude
                        tvLatitude.setText(String.valueOf(location.getLatitude()));
                        lati=String.valueOf(location.getLatitude());
                        //Set Longitude
                        tvLongitude.setText(String.valueOf(location.getLongitude()));
```

```
lon=String.valueOf(location.getLongitude());
}else {
    //When Location result is null
    //Initialize Location request
    LocationRequest locationRequest=new LocationRequest()
        .setPriority(LocationRequest.PRIORITY_HIGH_ACCURACY)
        .setInterval(10000)
        .setFastestInterval(1000)
        .setNumUpdates(1);
    //Initialize location call back
    LocationCallback locationCallback=new LocationCallback() {
        @Override
        public void onLocationResult(LocationResult locationResult)
        {
            //Initialize location
            Location location1=locationResult.getLastLocation();
            //super.onLocationResult(locationResult);
            //set latitude
            tvLatitude.setText(String.valueOf(location1.getLatitude()));
            //set longitude
            tvLongitude.setText(String.valueOf(location1.getLongitude()));
        }
    };
    // Request location updates
    fusedLocationProviderClient.requestLocationUpdates(locationRequest
        ,locationCallback, Looper.myLooper());
}

});
}else {
    //when location service is not enabled
    //Open location setting
    startActivity(new Intent(Settings.ACTION_LOCATION_SOURCE_SETTINGS)
        .setFlags(Intent.FLAG_ACTIVITY_NEW_TASK));
}
```

```
    }  
}  
}
```

---

### 6.2.3 User Details Fetching Page

---

```
package com.example.awmrapp;  
  
import androidx.appcompat.app.AppCompatActivity;  
import android.content.Intent;  
import android.os.Bundle;  
import android.view.View;  
import android.widget.Button;  
import android.widget.TextView;  
  
public class MainActivity_ocr1 extends AppCompatActivity {  
    private Button captureBtn;  
    TextView tid,tname,tprevious;  
    String sprevious,sid;  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_main_ocr1);  
        captureBtn=findViewById(R.id.idBtnCapture);  
        tid=(TextView)findViewById(R.id.idcustomer);  
        tname=(TextView)findViewById(R.id.idname);  
        tprevious=(TextView)findViewById(R.id.idprevious) ;  
        Intent iin= getIntent();  
        Bundle b = iin.getExtras();  
        if(b!=null)  
        {  
            sid =(String) b.get("id");  
            String sname =(String) b.get("name");  
            sprevious =(String) b.get("previous");  
            tid.setText("Customer Id : "+sid);  
        }  
    }  
}
```

```
        tname.setText("Name : "+sname);
        tprevious.setText("Previous Data : "+sprevious);
    }
    captureBtn.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            Intent i=new Intent(MainActivity_ocr1.this,ScannerActivity.class);
            i.putExtra("previous",sprevious);
            i.putExtra("cid",sid);
            startActivity(i);
        }
    });
}
```

---

## 6.2.4 Capturing and Detecting Image Page

---

```
package com.example.awmrapp;

import static android.Manifest.permission.CAMERA;
import androidx.annotation.NonNull;
import androidx.annotation.Nullable;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.app.ActivityCompat;
import androidx.core.content.ContextCompat;
import android.content.Intent;
import android.content.pm.PackageManager;
import android.graphics.Bitmap;
import android.graphics.Point;
import android.graphics.Rect;
import android.os.Bundle;
import android.provider.MediaStore;
import android.view.View;
import android.widget.Button;
```

---

```
import android.widget.ImageView;
import android.widget.TextView;
import android.widget.Toast;
import com.google.android.gms.tasks.OnFailureListener;
import com.google.android.gms.tasks.OnSuccessListener;
import com.google.android.gms.tasks.Task;
import com.google.mlkit.vision.common.InputImage;
import com.google.mlkit.vision.text.Text;
import com.google.mlkit.vision.text.TextRecognition;
import com.google.mlkit.vision.text.TextRecognizer;
import com.google.mlkit.vision.text.TextRecognizerOptions;
public class ScannerActivity extends AppCompatActivity {
    private ImageView captureIV;
    private TextView resultTV;
    private Button snapBtn,detectBtn,btnnext2;
    private Bitmap imageBitmap;
    static final int REQUEST_IMAGE_CAPTURE=1;
    String sprevious,cid;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_scanner);
        captureIV=findViewById(R.id.idIVCaptureImage);
        resultTV=findViewById(R.id.idTVDetectedText);
        snapBtn=findViewById(R.id.idBtnSnap);
        detectBtn=findViewById(R.id.idBtnDetect);
        btnnext2=findViewById(R.id.BtnNext2);
        Intent iin= getIntent();
        Bundle b = iin.getExtras();
        if(b!=null)
        {
            sprevious =(String) b.get("previous");
            cid=b.getString("cid");
        }
    }
}
```

```
btnnext2.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        Intent i=new Intent(ScannerActivity.this,UserDetails.class);
        i.putExtra("previous",sprevious);
        i.putExtra("cid",cid);
        i.putExtra("scanned",resultTV.getText().toString().replaceAll("2000","/"));
        startActivity(i);
    }
});
detectBtn.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        detectText();
    }
});

snapBtn.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        if (checkPermission())
        {
            captureImage();
        }else{
            requestPermission();
        }
    }
});

private boolean checkPermission(){
    //check whether permission granted or not
    int camerPermission=
        ContextCompat.checkSelfPermission(getApplicationContext(),CAMERA);
    return camerPermission== PackageManager.PERMISSION_GRANTED;
}
```



```
}

private void requestPermission(){
    //Request for permission
    int PERMISSION_CODE= 200;
    ActivityCompat.requestPermissions(this,new String[]{CAMERA},PERMISSION_CODE);
}

private void captureImage(){
    Intent takePicture=new Intent(MediaStore.ACTION_IMAGE_CAPTURE);
    if(takePicture.resolveActivity(getPackageManager())!=null){
        startActivityForResult(takePicture,REQUEST_IMAGE_CAPTURE);
    }
}

@Override
public void onRequestPermissionsResult(int requestCode, @NonNull String[]
permissions, @NonNull int[] grantResults) {
    super.onRequestPermissionsResult(requestCode, permissions, grantResults);
    if (grantResults.length>0)
    {
        boolean
            cameraPermission=grantResults[0]==PackageManager.PERMISSION_GRANTED;
        if (cameraPermission){
            Toast.makeText(this, "Permission Granted", Toast.LENGTH_SHORT).show();
            captureImage();
        }else
        {
            Toast.makeText(this, "Permission Denied", Toast.LENGTH_SHORT).show();
        }
    }
}

@Override
protected void onActivityResult(int requestCode, int resultCode, @Nullable Intent
data) {
```

```
super.onActivityResult(requestCode, resultCode, data);
if(requestCode==REQUEST_IMAGE_CAPTURE && resultCode==RESULT_OK){
    Bundle extras=data.getExtras();
    imageBitmap=(Bitmap) extras.get("data");
    captureIV.setImageBitmap(imageBitmap);
}
}

private void detectText(){
    InputImage image=InputImage.fromBitmap(imageBitmap,0);
    TextRecognizer recognizer=
        TextRecognition.getClient(TextRecognizerOptions.DEFAULT_OPTIONS);
    Task<Text> result=recognizer.process(image).addOnSuccessListener(new
        OnSuccessListener<Text>() {
            @Override
            public void onSuccess(@NonNull Text text) {
                StringBuilder result=new StringBuilder();
                for (Text.TextBlock block: text.getTextBlocks()){
                    String blockText=block.getText();
                    Point[] blockCornerPoint=block.getCornerPoints();
                    Rect blockFrame=block.getBoundingBox();
                    for (Text.Line line:block.getLines()){
                        String lineText=line.getText();
                        Point[] lineCornerPoint=line.getCornerPoints();
                        Rect linRect=line.getBoundingBox();
                        for (Text.Element element: line.getElements()){
                            String elementText=element.getText();
                            result.append(elementText);
                        }
                        resultTV.setText(blockText);
                    }
                }
            }
        }).addOnFailureListener(new OnFailureListener() {
```

```
        @Override
        public void onFailure(@NonNull Exception e) {
            Toast.makeText(ScannerActivity.this, "Failed To Detect text from
                image"+e.getMessage(), Toast.LENGTH_SHORT).show();
        }
    });
}
```

---

### 6.2.5 Bill Calculation Page

---

```
package com.example.awmrapp;
import androidx.appcompat.app.AppCompatActivity;
import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;
import android.widget.Toast;
public class UserDetails extends AppCompatActivity {
    private EditText presreading1,prevreading1;
    private TextView totalvalue1;//scannedvalue;
    private Button btnnext3;
    String sprevious,cid;
    double scanned;double pres,prev;
    Button BtnPay1;
    double t;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_user_details);
        totalvalue1=findViewById(R.id.totalvalue);
    }
}
```

```
presreading1=findViewById(R.id.presreading);
prevreading1=findViewById(R.id.prevreading);
btnnext3=findViewById(R.id.BtnNext3);
BtnPay1=findViewById(R.id.BtnPay);
Intent iin= getIntent();
Bundle b = iin.getExtras();
if(b!=null)
{
    sprevious =(String) b.get("previous");
    scanned = Double.parseDouble((String) b.get("scanned"));
    prevreading1.setText(sprevious);
    cid=b.getString("cid");
}
btnnext3.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        prev=Double.parseDouble(prevreading1.getText().toString());
        pres =Double.parseDouble(presreading1.getText().toString());
        if (pres==scanned && prev<pres) {
            double c = pres - prev;
            if (c <= 5000)
                t = 21;
            else if (c <= 10000)
                t = 21 + ((c - 5000) / 1000) * 4.2;
            else if (c <= 15000)
                t = 42 + ((c - 5000) / 1000) * 5.25;
            else if (c <= 20000)
                t = (c / 1000) * 6.3;
            else if (c <= 25000)
                t = (c / 1000) * 7.35;
            else if (c <= 30000)
                t = (c / 1000) * 9.45;
            else if (c <= 40000)
                t = (c / 1000) * 12.6;
```

```
        else if (c <= 50000)
            t = (c / 1000) * 14.7;
        else
            t = 735 + ((c - 50000) / 1000) * 42;
        totalvalue1.setText(String.valueOf(t));
    }
    else
    {
        Toast.makeText(UserDetails.this, "Invalid Readings !!!",
            Toast.LENGTH_SHORT).show();
    }
}
});

BtnPay1.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        if(pres==scanned && pres>prev){
            Intent pay=new Intent(UserDetails.this,Payment.class);
            pay.putExtra("cid",cid);
            pay.putExtra("present",""+pres);
            pay.putExtra("rs",""+t);
            startActivity(pay);
        }
        else
        {
            Intent img=new Intent(UserDetails.this,ImageUpload.class);
            startActivity(img);
        }
    }
});
}
```

---

### 6.2.6 Payment Page

---

```
package com.example.awmrapp;

import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;
import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.Toast;

import com.google.firebase.database.DataSnapshot;
import com.google.firebase.database.DatabaseError;
import com.google.firebase.database.DatabaseReference;
import com.google.firebase.database.FirebaseDatabase;
import com.google.firebase.database.Query;
import com.google.firebase.database.ValueEventListener;

public class Payment extends AppCompatActivity {

    Button PayBill;

    String cid,present,rs;

    @Override

    protected void onCreate(Bundle savedInstanceState) {

        super.onCreate(savedInstanceState);

        setContentView(R.layout.activity_payment);

        PayBill=(Button)findViewById(R.id.PayBill);

        Bundle b=getIntent().getExtras();

        if(b!=null)

        {

            present=b.getString("present");

            cid=b.getString("cid");

            rs=b.getString("rs");

            PayBill.setText("Rs. "+rs);

        }

        PayBill.setOnClickListener(new View.OnClickListener() {

            @Override
```

```
public void onClick(View view) {  
    //update firebase  
    DatabaseReference reference=  
        FirebaseDatabase.getInstance().getReference().child("User");  
    Query query2=reference.orderByChild("customer_id").equalTo(cid);  
    query2.addListenerForSingleValueEvent(new ValueEventListener() {  
        @Override  
        public void onDataChange(@NonNull DataSnapshot dataSnapshot) {  
            for (DataSnapshot snapshot : dataSnapshot.getChildren())  
            { snapshot.getRef().child("prev_data").setValue(present);  
                Toast.makeText(getApplicationContext(),"value  
                    updated",Toast.LENGTH_SHORT).show();  
            }  
        }  
        @Override  
        public void onCancelled(@NonNull DatabaseError databaseError) {  
        }  
    });  
    Intent i=new Intent(Payment.this,FinalPage.class);  
    startActivity(i);  
}  
});  
}
```

---

# Chapter 7

## Testing and Implementation

### 7.1 Testing and Types of Testing Used

Once a software is developed, the major activity is to test whether the actual results match with the experimental results. This process is called testing. It's used to make sure that the developed system is defect free. The main aim of testing is to find the errors and missing operations by executing the program. It also ensure that all of the objectives of the project are met by the developer. The objective of testing is not only to evaluate the bugs in the created software but also finding the ways to improve the efficiency, usability and accuracy of it. It aims to measure the functionality, specification and performance of a software program. Tests are performed on the created software and their results are compared with the expected documentation. When there are too much errors occurred, debugging is performed. And the result after debugging is tested again to make sure that the software is error free. The major testing processes applied to this project are unit testing, integration testing and system testing. In unit testing, our aim is to test all individual units of the software. It makes sure that all the units of the software work as intended. In integration testing, the combined individual units are tested to check whether it meets the intended function or not. It helps us to find out the faults that may arise when the units are combined. In system testing the entire software is tested to make sure that it satisfies all of the requirements. The tables shown below describe the testing processes conducted during the development of the project 'AquaMeter-Automated Water Meter Reading'. This defines the various steps taken to make the project error free.



### 7.1.1 Unit Testing

It involves the basic testing of a piece of code, the size of which is often undefined in practice. During the unit testing it is tested to know whether that particular unit in the proper manner as expecting, if not appropriate modifications are applied to get proper outputs

### 7.1.2 Integration Testing

Integration testing (sometimes called integration and testing, abbreviated IT) is the phase in software testing in which individual software modules are combined and tested as a group. Integration testing is conducted to evaluate the compliance of a system or component with specified functional requirements.

### 7.1.3 System Testing

System Testing is a type of software testing that is performed on a complete integrated system to evaluate the compliance of the system with the corresponding requirements. In system testing, integration testing passed components are taken as input.

### 7.1.4 Test Case Description

Table 7.1: Test Cases and Results

Sl No	Procedures	Expected result	Actual result	Pass or Fail
1	User-Login	The user logs into the application using his or her login credentials along with GPS location	Same as expected	Pass
2	User Details View	User can see their details	Same as expected	Pass

*Continued on next page*

Table 7.1 – *Continued from previous page*

Sl No	Procedures	Expected result	Actual result	Pass or Fail
3	User capturing the image	User capture and detect the image	Same as expected	Pass
4	Bill Calculation	Bill is calculated according to the detected value and entered value	Same as expected	Pass
5	Payment	After the bill is generated the payment is done and database is updated	Same as expected	Pass

# Chapter 8

## Results and Discussion

The main aim of the project was to capture the image of meter reading and bill generation. It also aims to produce a secure application with more security functions with the use of GPS. By tracing the GPS location the meter can access the details of registered consumers. Reduces physical effort of employees. And it is observed that the system performs all the functionalities as expected. By using this application the customer itself can capture the image ,instant bill generation and payment.

### 8.1 Advantages and Limitations

The proposed system posses more advantages over the existing system. The proposed system save a huge amount of time. Like every other system, this system also have it's own disadvantages. But they are negligible while comparing with the advantages and they can be overcame in future

#### 8.1.1 Advantages

- 1) Main advantage of our system is we can be a part of "save paper initiative" Because we are getting our bills on application itself this gives lot of paper saving in printed bills.
- 2) Quicker processing, no moving or storage of questionnaires near operator:Application is used to get the readings from the meter automatically by simply capturing the image of the meter and then performing the OCR technique which is nothing but —optical character recognition. The OCR technique is used to identify the character from an image and used this character to get the meter readings.

- 3) Savings in costs and efficiencies by not having the paper questionnaires
- 4) Scanning and recognition allowed efficient management and planning the rest of the processing workload. All that needs to be done is to capture the image then android app will perform operation of extracting the meter reading text from image and send to the server. Then the application does the calculation and proceed bills are shown to the relevant consumers at the same instance.
- 5) Reduced long term storage requirements, bills could be managed after the initial scanning, recognition and repair.
- 6) Quick retrieval for editing and reprocessing.
- 7) Minimizes errors associated with physical handling of the bill.
- 8) Easy to use features

### **8.1.2 Limitations**

- Lack of payment integration
- Lack of Automatic notifications by message to the customer about bill

# Chapter 9

## Conclusion and Future Scope

### 9.1 Conclusion

AUTOMATED WATER METER READING - AquaMeter , simpler, accurate and efficient application is developed for reading meter using OCR technique. The automated system of meter reading will reduce the burden on the employees, avoid human errors and incorrectness in bill generation. With the use of automated system most of the manual processes and calculations are eliminated so that the meter readings can easily be collected more accurately to be updated to the server.

This gives the solutions to address the problems related to manual water billing process. The customers also complaint about incorrectness of bill, this is because the assumption of reading when not available and leads to major problem of current technology. Most of the time bill calculation and system updating are done manually. There can be some resulting human errors to frustrated customers who are not satisfied about the service of the Water Authority at all. The suggested system has come up with solutions which address all the above mentioned problems. Most of the manual processes and calculations are eliminated so that the meter readings can easily be collected more accurately to be updated to the system. On the other hand, the Android customers are also given a mobile solution so that they can view their latest bills and make payments and other important tasks efficiently.

The development of this application demonstrates the concept and implementation of new, it has a low infrastructure cost, low operating costs, more data security and less man power required. Therefore it not only solves the problem of conventional meter reading but also provides additional

feature bill generation on mobile. The inputted image is preprocessed with help of OCR. The result is updated in Database .With help of result of OCR technique the bill is generated and payment is done by the customer.All is needed is a mobile with good internet connectivity.

## 9.2 Future Scope

- Admin App Development for verification of the image uploaded and updating the customer about any news regarding the authority etc...
- Payment Integration for the system.
- Designed for water meter reading, but can be extended to other type of meter reading.

# Bibliography

- [1] Madhubala Talwar, Pooja Nimbalkar, Sima Bodage, Bharati Gavit, Prof. Shreelekha Mankhair , “Android based Meter Reader using OCR Technique” , April 2016 , Department of Computer Engineering, Modern Education Society’s College of Engineering, Pune, India, Journal of Emerging Technologies and Innovative Research (JETIR)
  
- [2] Riddhi Gor, Priyanka Karkate, Vidyalaxmi Selvaraj, Prof. Harsh N. Bhor B.E IT Student, KJSIEIT, Sion, Mumbai, Asst Professor, KJSIEIT, Sion, Mumbai, Maharashtra, India , “Android Application for Meter Reading Using OCR” , International Journal for Research in Engineering Application Management (IJREAM) Vol-01, Issue 03, June 2015.