



An Undergraduate Internship/Project on Accomium Accounting Mobile App

By

Mohammad Asaduzzaman

Student ID: **1721599**

Summer, 2021

Supervisor:

Sheikh Abujar

Lecturer

Department of Computer Science & Engineering

Independent University, Bangladesh

September 12, 2021

Dissertation submitted in partial fulfillment for the degree of Bachelor of
Science in Computer Science & Engineering

Department of Computer Science & Engineering

Independent University, Bangladesh

Attestation

This is to certify that the report is completed by me, Mohammad Asaduzzaman (ID:1721599), submitted in partial fulfillment of the requirement for the Degree of Computer Science and Engineering from Independent University, Bangladesh (IUB). It has been completed under the guidance of Sheikh Abujar, Lecturer, Department of Computer Science and Engineering, Independent University, Bangladesh. I also certify that all my work is genuine which I have learned during my Internship. All the sources of information used in this project and report has been duly acknowledged in it.

Asaduzzaman

08 - 08 - 2021

Signature

Date

Mohammad Asaduzzaman

Name

Acknowledgement

First of all, I would like to express my gratitude to Almighty Allah to enabling me to complete this report on Internship. Successfully completion of any type of project

requires help from a number of persons. I have also taken help from different people for the preparation of this report. Now, there is a little effort to show my deep gratitude to that helpful person. I convey my sincere gratitude to my honorable faculty and supervisor

Sheikh Abujar, Lecturer, Department of Computer Science and Engineering, Independent University, Bangladesh. Without his kind direction and proper guidance this study would have been a little success. In every phase of report his supervision and guidance shaped this report to be completed perfectly. Then I would like to express my gratitude to MR. Rajib Hossain Pavel, CTO of New England Automation, for giving me the opportunity to complete my internship at Accoumium and my Android Application Developer team members for their guidance and support in this three months internship program.

Letter of Transmittal

07, August 2021

To,

Sheikh Abujar Lecturer Department of Computer Science and Engineering, Independent University, Bangladesh Subject: Submission of Internship Report Summer, 2021 Dear Sir, With due honor and respect, I am Mohammad Asaduzzaman, from section – 07, Summer 2021 would like to inform you that I want to submit my internship report.

I tried my best to work sincerely to cover all aspects regarding the matter. I have thoroughly enjoyed in preparing this internship report which has contributed significantly to my understanding on the essentials and importance of practical knowledge.

I therefore looking forward to you that my report will be achieve your approval. If you need any kind of clarification I will ready to do so.

Yours sincerely,

Mohammad Asaduzzaman

Student of Computer Science and Engineering

School of Engineering, Technology and Sciences, IUB

ID: 1721599.

Email: 1721599@iub.edu.bd

Evaluation Committee

.....
Signature

.....
Name

.....
Supervisor

.....
Signature

.....
Name

.....
Internal Examiner

.....
Signature

.....
Name

.....
External Examiner

.....
Signature

.....
Name

.....
Convener

Abstract

This report is based on the internship project. Categorized into eleven different chapters. In every chapter here discussed detail. introduction part contain the background of the work, objectives of this project and give some idea about the scopes. literature review part contain how i implement my under graduation knowledge how this project is related with this project. In project management part contain about the Work Break-down structure, Process/Activity wise time distribution, Gant Chart, Process/Activity wise Resource Allocation and Estimated Costing of this project. Methodology contain about the project methodology. Body of the project describe the system design part with some diagram and rest of the part of this project about some analyse, result and concluding summary. Now large number of people are using smart phone and it is now very impotent part of life. Many work now can be easily done because of smartphone. Accomium Accounting app also developing for to make easy the accounting calculation by a smart phone and also to syncs all financial data.

Contents

| | |
|---|------------|
| Attestation | i |
| Acknowledgement | ii |
| Letter of Transmittal | iii |
| Evaluation Committee | iv |
| Abstract | v |
| 1 Introduction | 1 |
| 1.1 Overview/Background of the Work | 1 |
| 1.2 Objectives | 1 |
| 1.3 Scopes | 2 |
| 2 Literature Review | 3 |
| 2.1 Relationship with Undergraduate Studies | 3 |
| 2.2 Related works | 4 |
| 3 Project Management & Financing | 5 |
| 3.1 Work Breakdown Structure | 5 |
| 3.2 Process/Activity wise Time Distribution | 6 |
| 3.3 Gantt Chart | 6 |
| 3.4 Process/Activity wise Resource Allocation | 7 |
| 3.5 Estimated Costing | 7 |
| 4 Methodology | 8 |
| 5 Body of the Project | 11 |
| 5.1 Work Description | 11 |
| 5.2 System Analysis | 11 |
| 5.2.1 Six Element Analysis | 11 |
| 5.2.2 Feasibility Analysis | 12 |

| | | |
|----------|---|-----------|
| 5.2.3 | Problem Solution Analysis | 12 |
| 5.2.4 | Effect and Constraints Analysis | 13 |
| 5.3 | System Design | 14 |
| 5.3.1 | Rich Picture | 14 |
| 5.3.2 | UML Diagrams | 15 |
| 5.3.3 | Usecase Diagrams | 16 |
| 5.3.4 | Functional and Non-Functional Requirements | 16 |
| 5.4 | Product Features | 18 |
| 5.4.1 | Input | 19 |
| 5.4.2 | Output | 24 |
| 5.4.3 | Architecture | 27 |
| 6 | Results & Analysis | 28 |
| 7 | Project as Engineering Problem Analysis | 33 |
| 7.1 | Sustainability of the Project/Work | 33 |
| 7.2 | Social and Environmental Effects and Analysis | 33 |
| 7.3 | Addressing Ethics and Ethical Issues | 34 |
| 8 | Lesson Learned | 35 |
| 8.1 | Problems Faced During this Period | 35 |
| 8.2 | Solution of those Problems | 35 |
| 9 | Future Work & Conclusion | 36 |
| 9.1 | Future Works | 36 |
| 9.2 | Conclusion | 36 |
| | Bibliography | 37 |

List of Figures

| | | |
|------|---|----|
| 3.1 | Work Breakdown Structure Figure | 5 |
| 3.2 | Gantt Chart | 6 |
| 4.1 | Methodology | 10 |
| 5.1 | Rich Picture Of Accomium Accounting Application | 14 |
| 5.2 | UML Diagram | 15 |
| 5.3 | Usecase Diagram | 16 |
| 5.4 | Sign up and Show & Hide Fields | 19 |
| 5.5 | Add Accounts & Customer/Supplier | 20 |
| 5.6 | Add Product | 21 |
| 5.7 | Add Sale Details | 22 |
| 5.8 | Setting option List (Settings Repository) | 23 |
| 5.9 | Notification Option List (Notifications Repository) | 23 |
| 5.10 | Added Accounts & Customer/Supplier List | 24 |
| 5.11 | Added Product List | 25 |
| 5.12 | Setting & Notification Option List | 26 |
| 5.13 | Android Clean Architecture | 27 |
| 6.1 | Dependency Libraries | 28 |
| 6.2 | Customer & Supplier List | 29 |
| 6.3 | Accounts List | 30 |
| 6.4 | Product List | 31 |
| 6.5 | Add Sale | 32 |

List of Tables

| | | |
|-----|---|----|
| 3.1 | Process/Activity wise Time Distribution | 6 |
| 3.2 | Process/Activity wise Resource Allocation | 7 |
| 3.3 | Estimated Costing | 7 |
| 5.1 | Six Element Analysis | 11 |

Chapter 1

Introduction

1.1 Overview/Background of the Work

Mobile app development is the act or process by which a mobile app is developed for mobile devices, such as personal digital assistants, enterprise digital assistants or mobile phones. It is becoming more and more popular now a days because smartphones are becoming affordable and available for almost everyone. And internet is also available almost everywhere. In this project I have worked on a mobile application for Accomium, under the parent company name of the New England Automation. This is the Introduction chapter that includes the outlook and the features of the project.

Accomium is a startup company in Bangladesh. They work on web application and Mobile application development. They developing a free accounting system mobile application for their marketing purpose. This Android App is Kotlin based. On the frontend build with heavy on Compose and backend follow MVI Architecture.

1.2 Objectives

In general, a project objective describes the desired outcomes of a project. Objective of a project is specific, measurable and must meet time, budget and most importantly meet the client's requirements. The main objectives of this application are described below: Time efficiency: Using the application user do not have to use any browser to visit their application, all updates and information are organized here, so user can easily check their required update in a short time.

User friendly interface: The user interface is very easy to use, those who are not used to in browsing internet will also be able to check anything from the application easily.

Make task easier: The application will make their every task easier to complete. For example, now they can easily check accounting related thing with this accounting system app, they can easily check any notice or get alert by only one click on the application.

Cost Free: User can use all of the service of this application without any cost.

Online Platform: The application will work as an online accounting system for all user who will use this application. They can complete all their accounting related online tasks from one application. Purchasing order, Sales order, Bank transection, financial information's etc.

Linked with website: The application is linked with their website; user can easily visit required page of their website by the application. They will also can login to their account and complete any kind of task.

1.3 **Scopes**

- Notification: User can check notice board directly.
- Cost Free: User can use this app freely they have to login this app for use this system.
- Record: Automatic save Financial / Accounting report and transactions.
- Bank Reconciliation: Combines major financial accounting functions into one application.
- Invoicing: Make easy to send customer their invoices to accounting system. To make easy Small business accounting.
- Helpline: All kind of help line numbers are given with dial button.
- Forecast: For better sales forecast.
- Generate report: It will generate report about all interaction.
- Alert: It will alert about Know timely submission of SST and income tax.
- Calendar: User can check year calendar directly.

Chapter 2

Literature Review

2.1 Relationship with Undergraduate Studies

An internship is professional training in a safe and organized work environment that fosters mentor relationships. The general job training program provides internships with a mentor or one of the experts in the field from which you can learn. Counselors train and give advice to their experts on how to go about work ethic in a particular industry. The lessons are designed to help you develop work habits that will lead to your success. It is the practical experience of the courses a student has done in his undergraduate period. My project is directly and indirectly related with few courses I have done. Here I am describing the relation of the project with my undergraduate courses: Mobile Application

Development (CSE464/CEN464/CSC464): I have done Mobile Application Development course under Lecturer Mr. Sanzar Adnan Alam in Summer 2020. The basic of mobile application I learned here. In this course I have completed a few projects. Those projects are listed here:

- Calculator Application
- 7 Days Weather Forecast Application
- Resume Builder Application
- Tutor Finder System

These projects helped me a lot in my internship period. The project is fully based on the knowledge I gained in the course. I learned layout designing in Android Studio in this course, I also learned many features used in the project from the course.

Object Oriented Programming (CSE213/CEN305/CSC305): I have done Object Oriented Programming course under Lecturer Mr. Subrata Kumar Dey in Spring 2019. In Android Studio tool, there are two programming language option, Kotlin and Java. I used kotlin. I learned Java in my Object-Oriented Programming course but Kotlin is

very similar language like java so it's not that much hard to switch between Java to Kotlin. In this course I have completed a few projects. Our final project was Hotel Booking system application using Java. This project helped me a lot to learn Java.

Web Applications and Internet (CSE309/CSC454): I have done Web Applications and Internet course under Lecturer Sanzar Adnan Alam in Spring 2020. Web application and internet is a basic course for an intern in computer science and engineering student. The project is not directly related to the project, but in this project for user data we have to link this application with the website. And for this project basic knowledge of web application is needed.

Database Management (CSE303/CEN401/CSC401): I have done Database Management course under Mahadi Hasan, PhD (Associate Professor) in Summer 2019. From this course we learned basic about Database Management without this knowledge work in a project containing real time data is not possible. Entity Relationship Diagram, Rich Picture, Process Diagram, Rich Picture, SQL and many other key parts of database management were taught here. These concepts helped me a lot to complete the project.

The project is also connected with few other courses. It is like, I learned one thing from any course and realizing the use after years while working on it in practical project. The mobile application project is directly related to Mobile Application Development and Object-Oriented Programming course. Web Applications and Internet, Database Management courses are also related with the project. The project is also indirectly related with few courses or topics not described here.

2.2 Related works

Xero is a New Zealand domiciled public technology company, listed on the Australian Securities Exchange. Xero is a cloud based accounting software platform for small- and medium-sized businesses.

There are various type of mobile apps are available in the app stores such as: • Mobile Banking Application.

- Finance App
- Calendar diary App
- Fees management and payment App.
- Business Forecast App

Chapter 3

Project Management & Financing

3.1 Work Breakdown Structure

A work-breakdown structure in project management and systems engineering, is a deliverable-oriented breakdown of a project into smaller components but it is very important document for maintain the project management. Our Accomium accounting system application project we used a top down approaches. It help to maintain all of the tasks of project to complete properly and also help to set the estimation budget and schedule.

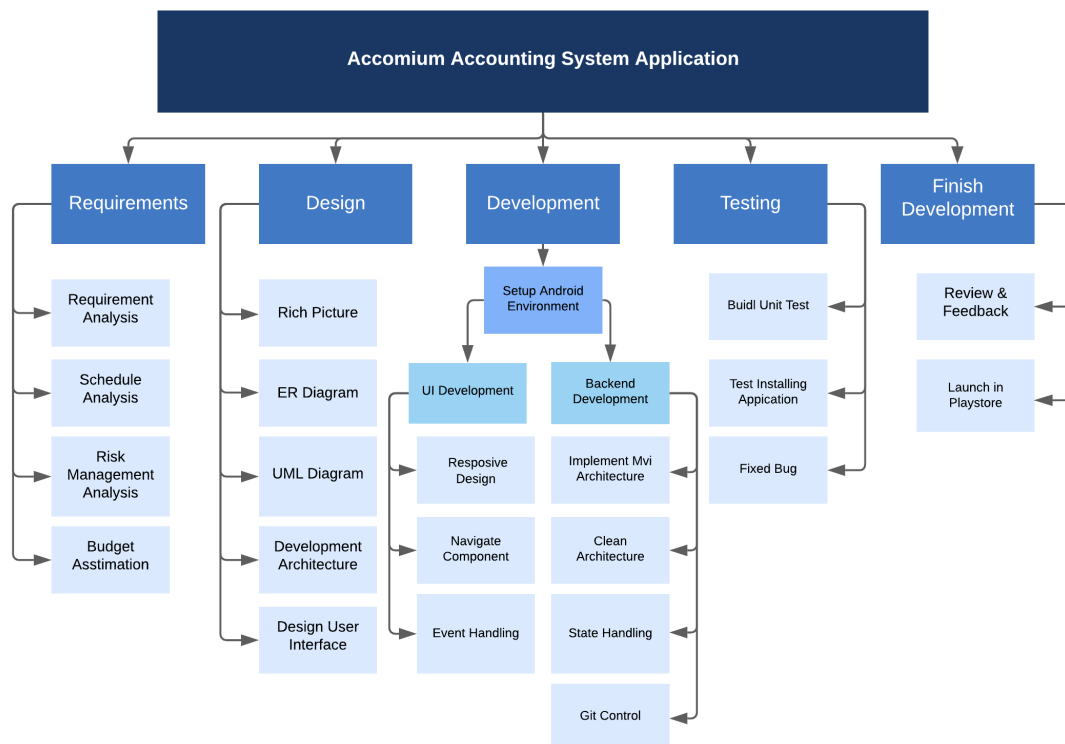


Figure 3.1: Work Breakdown Structure Figure

3.2 Process/Activity wise Time Distribution

For our Accounting system application we set a time schedule of each main section of Work Breakdown Structure. Time set depending on the task. For every task time duration is not same because depending on important and also different task take different for complete. Sometime changes happened when necessary depending on the situation.

| No | Activity | Start | Duration |
|----|-----------------------|--------------|----------|
| 1 | Requirement Analysis | 20 June | 10 Days |
| 2 | Design | 1 July | 20 days |
| 3 | Development | 26 July | 45 days |
| 4 | Testing | 12 September | 10 days |
| 5 | Finishing Development | 22 September | 5 days |

Table 3.1: Process/Activity wise Time Distribution

3.3 Gantt Chart

This Gantt chart used to describe our plan and schedule all the activities which are needed to complete the project. This chart Describe step by step with required time.

Gantt Cart



Figure 3.2: Gantt Chart

3.4 Process/Activity wise Resource Allocation

For completing our project successfully each step allocate with each department to complete and each department need some tools to complete their duty. The Table below shows them in details.

| Process/Activity | Resource Allocation |
|----------------------|--|
| Requirement | System Analyst & Designer |
| Design | System Analyst & Designer , UI/UX Designer |
| Development | Mobile Application Developer |
| Testing | Q/A Tester |
| Finishing & Feedback | Team Leader |

Table 3.2: Process/Activity wise Resource Allocation

3.5 Estimated Costing

The cost based on need to complete each step of accounting system app. Sometime cost can be changed depending on the project.

| No | Activity | Estimated Costing |
|-------|-----------------------|-------------------|
| 1 | Research | 500 \$ |
| 2 | Design | 500 \$ |
| 3 | Android Development | 1500 & |
| 4 | Testing | 250 \$ |
| 5 | Finishing Development | 100 \$ |
| Total | | 2850 \$ |

Table 3.3: Estimated Costing

Chapter 4

Methodology

Mobile applications represent a new application model being introduced to information systems in the recent time. For that reason, it represents a good challenge to expand research area of information systems development methodologies with research on mobile applications development methodology. The first part of the chapter introduces classical and a context-aware mobile application model. Based on that, the second part explores the role of mobile applications in information systems with the emphasis on showing the semantic contribution of the use of mobile applications in information systems. The core part of the chapter introduces mobile applications development methodology. The methodology is introduced through development phases and tasks which have to be performed within phases. The emphasis of methodology introduction is on phases of strategy and analysis. [1]

Planning: Planning is the first thing of start a project. For our accounting app our team set the planning before the project start. Planning about component requirement. How we will complete our project step by step. Set up the responsibility among the team. Set the goal about project. User will use this app without any cost but this project planning is to produce to user premium service.

Requirement: User requirements are typically written when discussing the use cases for a project. The requirements definition is done with the customer or product managers that know how the embedded system will be used by the user. Many user requirements deal with how a user will interact with a system and what that user expects. If there is a screen or human machine interface aspect to the system, a user requirement may be based on what happens when the user selects an action on the screen. Maybe with a button press not only does a process start, but it also switches to another screen and provides an audible notification. When user requirements such as these are written down, they can often break into multiple system requirements later due to switching of screens, the maximum delays in starting the process, and finally what the next screen

should look like. One pitfall is starting to try to write the system requirements during a user requirement meeting. This often detracts from gaining insight into the requirements of the user, and key functionality pieces could be missed. In fact, as mentioned to earlier, it is often better to keep user requirements and system requirements separate in their tracking and reporting. The user requirements are often more readable, understandable and provide a better sense of how the system will operate. Even though user requirements may lack specifics on what really needs to occur in the system, they are still valuable in that they can provide the overarching system functionality expectations. As a developer I observed user process and planned my project module considering to make user process easier.[2]

Design: Android users expect your app to look and behave in a way that's consistent with the platform. Not only should you follow material design guidelines for visual and navigation patterns, but you should also follow quality guidelines for compatibility, performance, security, and more. Jetpack Compose is Android's modern toolkit for building native functional UI. It simplifies and accelerates UI development on Android. Quickly bring your app to life with less code, powerful tools, and intuitive Kotlin APIs. Jetpack is a suite of libraries to help us follow best practices, reduce boilerplate code, and write code that works consistently across Android versions and devices so that we developers can focus on the code they care about.

Implementation in Code: Android Studio is the official integrated development environment for Google's Android operating system, built on JetBrains' IntelliJ IDEA software and designed specifically for Android development. To create an android app android implement those planning, requirement and business logic. For our project we use MVI architecture with kotlin language. We will use database locally and remotely. To connect our app remotely we will use GraphQL API.[3]

Testing: Testing your app is an integral part of the app development process. By running tests against your app consistently, you can verify your app's correctness, functional behavior, and usability before you release it publicly. After test our code if we found any changes need then we will solve those code again and again until get success.

Maintenance: Maintenance is one of the most crucial stages of the mobile app development lifecycle. That's why you should set aside a separate budget just for the app maintenance. This project is marketing purpose of this company so this project have to maintain properly to give customer good experience and also need update add new features in project as customer demand for better service.

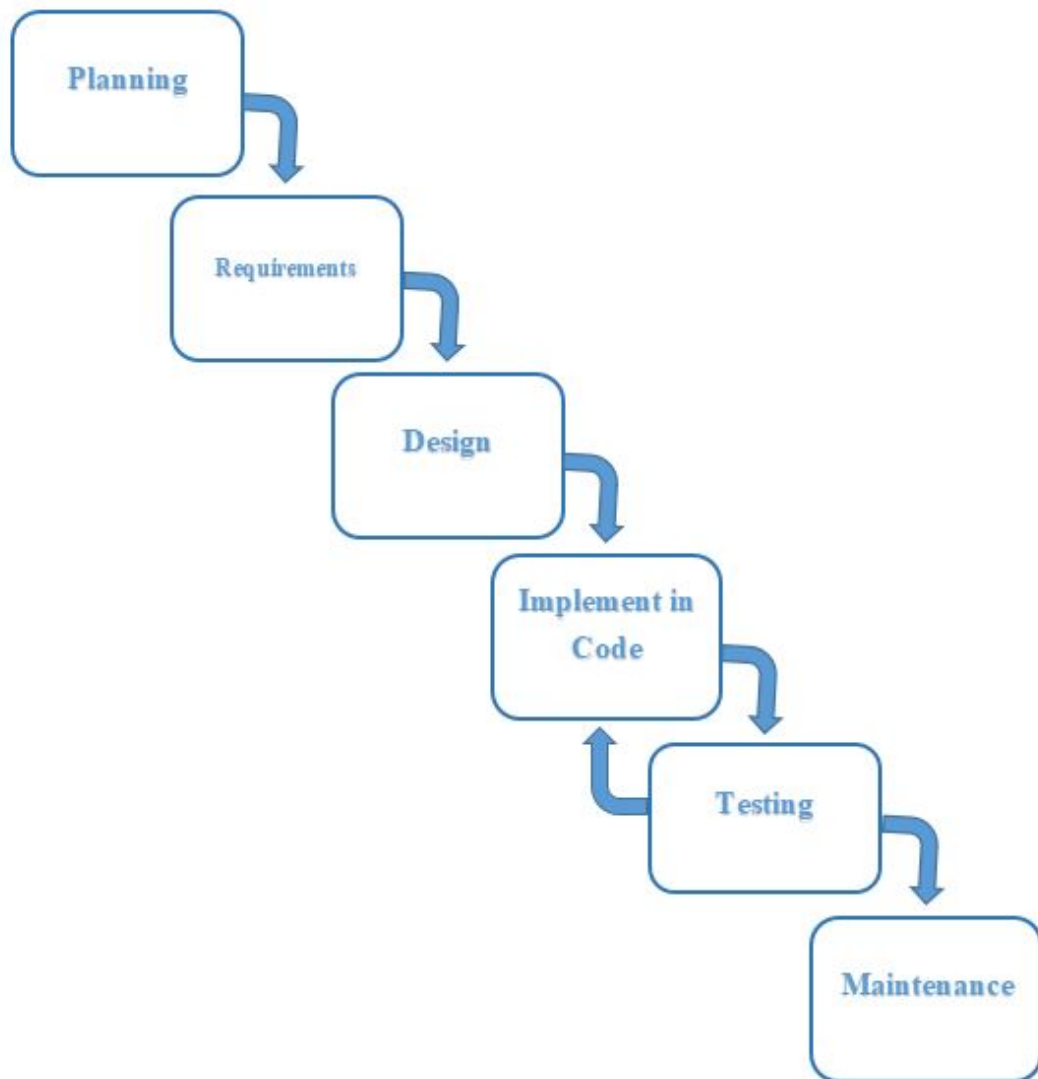


Figure 4.1: Methodology

Chapter 5

Body of the Project

5.1 Work Description

Accomium Accounting system is an accounting app which is help full for small business accounting. This is a web base application it will run depending on website. This application is free for user but there will be some premium feature which will need cost for using. My work in this project is mainly help the developing team by setup the environment with injecting dependency in the project, build some user interface and event handling work. Navigating one page to another page. Convert the project with MVI architecture and creating MVI code with list component which will usable for whole project in every feature when it will connect with website using API.

5.2 System Analysis

5.2.1 Six Element Analysis

| System Role | | | | | |
|--------------------------|--------------------|----------------------|----------------------------------|----------------------|----------------------------|
| Human | Computer Hardware | Non-computer Hardwar | Software | Database | Communication & Networking |
| User, Company & Employee | PC, Tablet & Phone | NA | Accoumium Accounting Application | Cloud based database | Website, mail & Internet |

Table 5.1: Six Element Analysis

5.2.2 Feasibility Analysis

Resources: Before or after building a project company need time and money. Company have to maintain team to support the system and that team have to experts. For maintaining those employee company have to give right amount of salary and working environment. Sometime support team can be take some time to adjust with the environment. If company maintain those thing properly then it is very helpful and better in project long run.

Technical: Technical support team in long run of this project before and after launching project play very impotent role they maintain the system. Company need to ensure the capable person in the technical team. Technical team ensure the user use application the application smooth experience. And they find the lacking's of the system and try to solve those. If they not able to solve those problem then they have to hire expertise for solve the problem.

Schedule: Project scheduling is a key tool used for project planning, efficient execution, reporting and ultimately project claims. A good project schedule provides some confidence in the project plan and the dates for important project milestones. It helps manage resources. Finally, a good critical path method schedule is an essential prerequisite to a successful schedule risk analysis.[4]

5.2.3 Problem Solution Analysis

Set Project Environment: for building a project as a developer in android studio the first challenge is to set up the environment of the project by adding dependence library. For building a project it required lot of library to inject but problem is there are multiple version for each library and all are not working properly. Without right library version building project is very tough. For solution choice right library version by help of documentation or other project.

New environment: This project is building with new UI tool kit google jetpack compose, because the number of resource are not available that much so it is very tough to working with it. MVI architecture also new for thing to learn. How to code with MVI and mainly maintaining the flow of data also tough thing for newbie. By practicing and by help from documentation from android developer documentation it can be solved.

5.2.4 Effect and Constraints Analysis

This application is building for those who have small business to track their Sale, Purchase, Payments, Expenses, and Taxes etc. this app will help to maximum survive of business which is free. This app has some beneficial also have some backward. User have to update each and every data for their activities. If user missed any record or anything to record in the system the system will fail to generate right accounting result. This app depend on the website and the website also on progress. In this pandemic situation it's also affected our team member and delay some time. But we had to get used with work from home to continue our development which is a big advantage during the ongoing pandemic.

5.3 System Design

5.3.1 Rich Picture

Rich picture of Accomium Accounting Application.

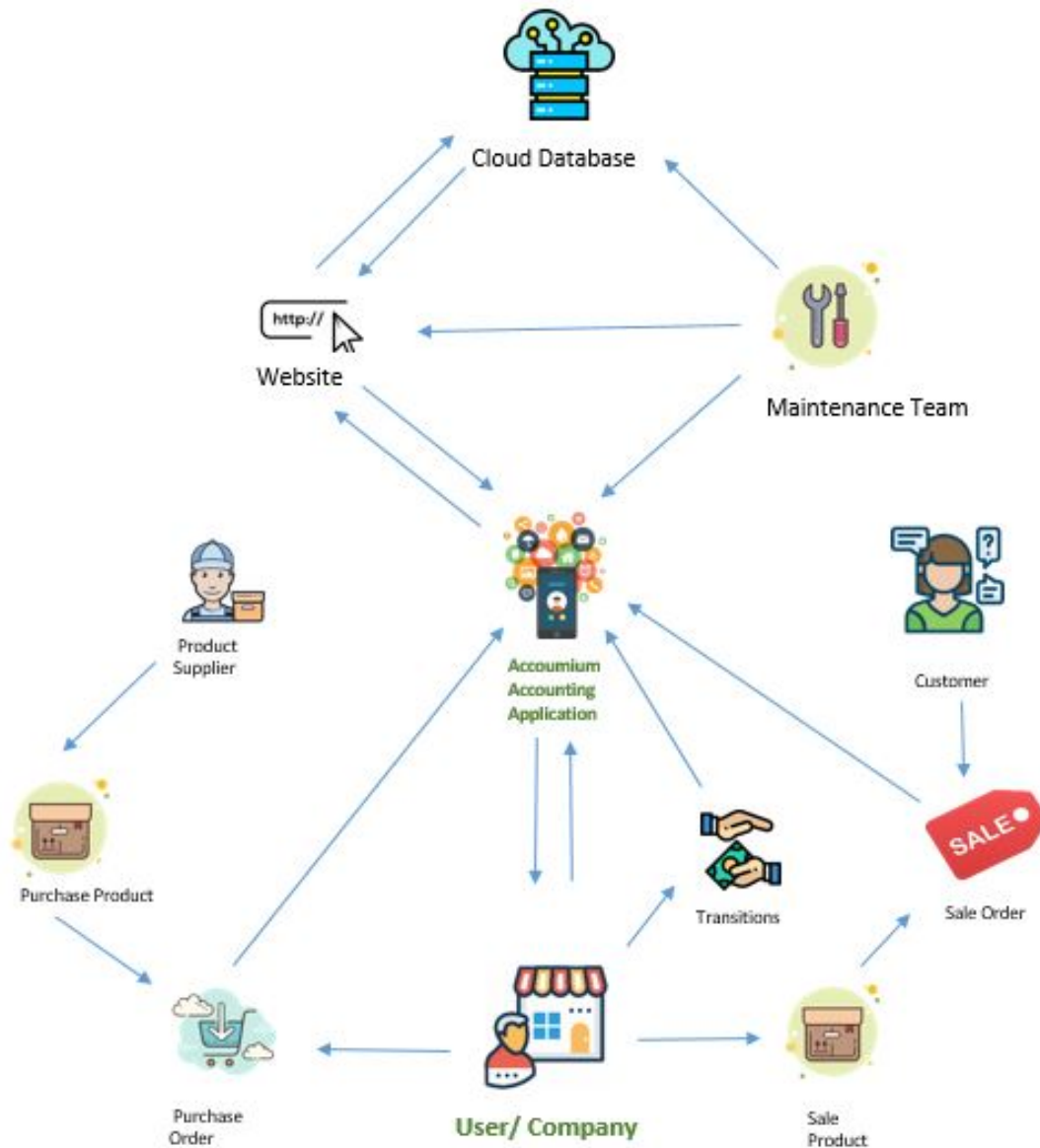


Figure 5.1: Rich Picture Of Accomium Accounting Application

5.3.2 UML Diagrams

Here is the process diagram for unlock premium features:

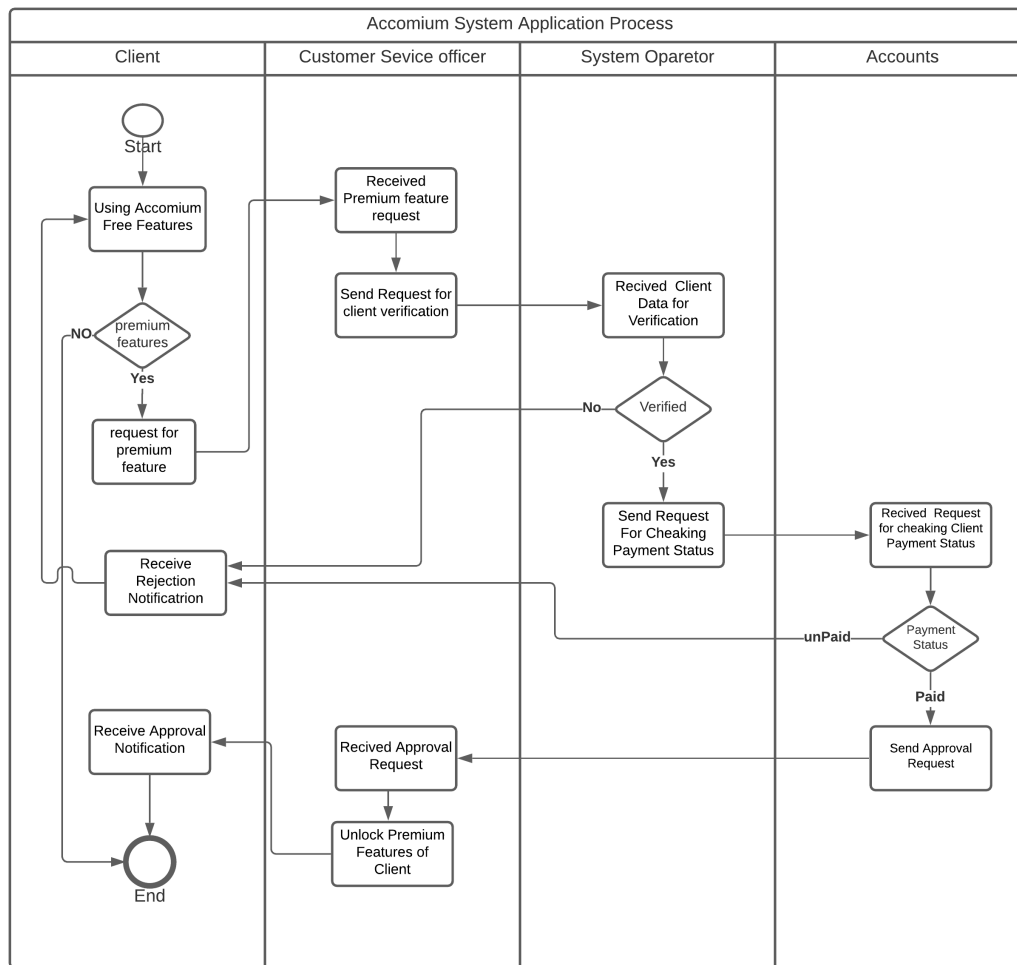


Figure 5.2: UML Diagram

5.3.3 Usecase Diagrams

The use case diagram represents the the system. It shows the actors, cases, communication links, system and relationship.

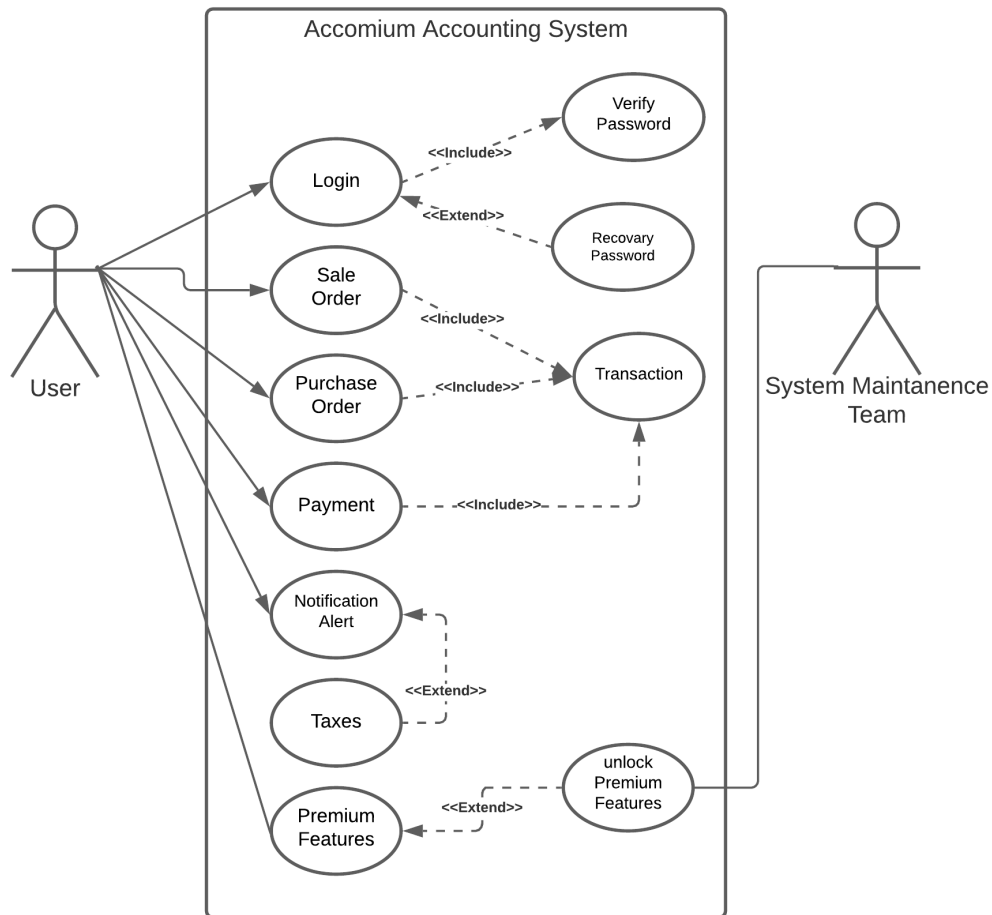


Figure 5.3: Usecase Diagram

5.3.4 Functional and Non-Functional Requirements

Functional Requirements:

1. Responsive and user friendly user interface.
2. User have to register with a unique email, password with information system will verified those data.

3. User can logged in with using email and password if they registered user.
4. User can use account from multiple device this system will use cloud base database so that each and every data will store in data base.
5. User can edit their information any time.
6. User can recovery password by email if they forget their password.
7. Remote data will synchronized with local data.
8. All time Updated data will be showing in the user interface.
9. This app will helps to reorder multiple users Sale, Purchase, Payments, Expenses, and Taxes etc. with the detail information of those.
10. This app will reorder all transection of users Sale, Purchase, Payments, Expenses, and Taxes etc.
11. User will be notify by notification alert when deadline for payment or collecting money from other. Example: taxes payment deadline.
12. Auto invoice and report will be generated by this system.
13. System will generate payment facilities for user.
14. This app also connect with a website if user update anything from website then the data will automatically update in the application.
15. User can track the amount of money usage in any feature.
16. User can use free feature like maximum survive and with low cost user can use some premium feature Growth report – guidelines, Digital marketing and financial consultancy's etc.

Non-Functional Requirements:

1. Usability: The system is uses a website. The system user friendly and online help makes using the system easy. The system is designed for a user friendly environment so that user can perform the various tasks easily and in an effective way.
2. Availability: The system is available 100% for the user and used 24 hour a day and also 365 days in a year.

3. Efficiency: Even if the system fails, the system will recovered backup within an hour or less. User data will safely backup for disaster recovery.
4. Economic: This system will deploy to a cloud hosting provider, So we don't need a data center or server. We will activate the pay-as-you-use package from the cloud provider. So we will only pay as we use. No extra cost need for maintain data center and place.
5. Performance: The information is refreshed at regular intervals depending upon whether some updates have occurred or not. The system will respond to the user in very short time. Performance should always be smooth and easy to understand,
6. Reliability: The system has to 100% reliable due to the importance of data and the damages that can be caused by incorrect or complete data. Safe backup, no data can be lost. System update or any kind of testing will not affect the running system.
7. Maintainability and Portability: This system will maintain user privacy. It will maintain by adding new features and also it will maintain the security check for system security. it runs smoothly and does not get slow or lag. Any bugs or problems can be fixed in short time.

5.4 Product Features

- Customer
- Supplier
- Product
- Sale
- Purchase
- Sale order
- Purchase order
- Payment
- Tran section
- Expense
- Financial consultancy

5.4.1 Input

Our Project is now in under constructions i am showing some input here.

Sign up and Show & Hide Fields

The figure displays two mobile application screens side-by-side. The left screen is a sign-up page with the heading 'Welcome, Sign in to continue,'. It features a 'Password' input field, a 'Sign in with Google' button, an 'Or' separator, and a 'Sign in with Email' button. At the bottom, there is a link: 'Haven't an account ? Log in'. The right screen is titled 'Show/Hide Fields' and contains a list of fields with toggle buttons: 'Terms and condition', 'Header/Footer, Signature', 'Image Attachment', 'Add Ref No.', and 'PD Number/Date'. Each field has a blue 'Show' button and a grey 'Hide' button. At the bottom of this screen are 'Save' and 'Cancel' buttons. Both screens have a black Android navigation bar at the very bottom.

Figure 5.4: Sign up and Show & Hide Fields

Add Accounts & Customer/Supplier

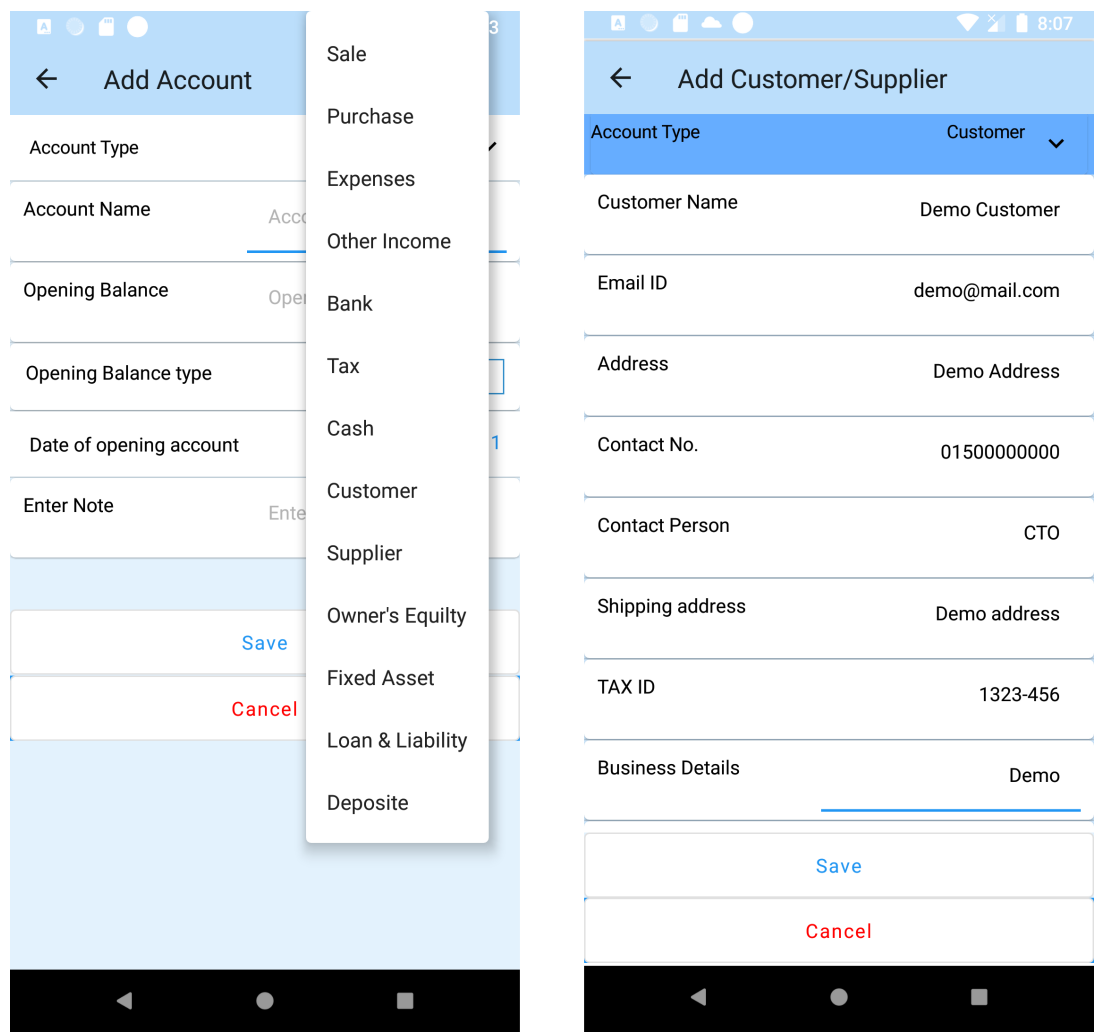
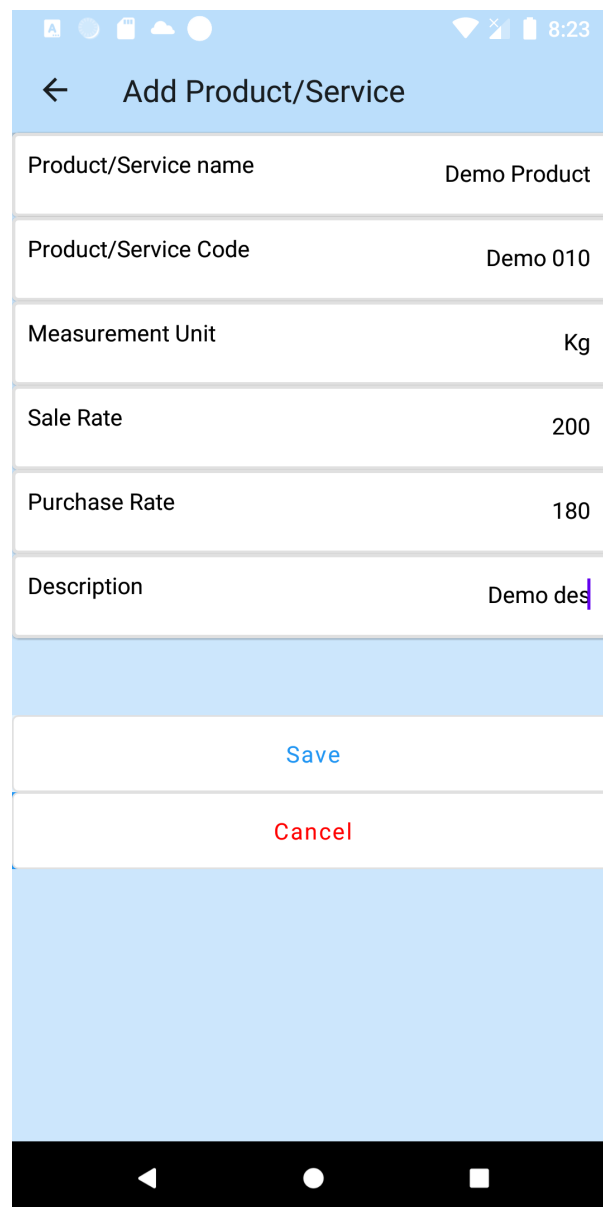


Figure 5.5: Add Accounts & Customer/Supplier

Add Product



The screenshot shows a mobile application interface for adding a product or service. The title bar at the top is light blue and contains a back arrow and the text 'Add Product/Service'. The status bar at the very top shows icons for signal, Wi-Fi, battery, and the time 8:23. The form itself is a table with the following data:

| | |
|----------------------|--------------|
| Product/Service name | Demo Product |
| Product/Service Code | Demo 010 |
| Measurement Unit | Kg |
| Sale Rate | 200 |
| Purchase Rate | 180 |
| Description | Demo des |

Below the form table, there is a light blue horizontal bar. Underneath that bar are two buttons: a blue 'Save' button and a red 'Cancel' button. At the bottom of the screen is a black navigation bar with standard Android icons (back, home, recent apps).

Figure 5.6: Add Product

Add Sale Details

←

Add Sale Details

Invoice

01-09-2021

Due Date

23-09-2021

▼

☐ Do Not Provide Line Item

Record Sale In

Sale Account

▶

Po Number

PO Date

DD-MM-YYYY

Customer Name

👤

Demo Customer 2

🛒

Select Line Items

USD 0

▼

Product List

Demo Product

×

Demo Product 2

Book

500

Hello

Demo Des 2

☰

Discount

USD 0.00

▼

🏷️

Other Charges

USD 0

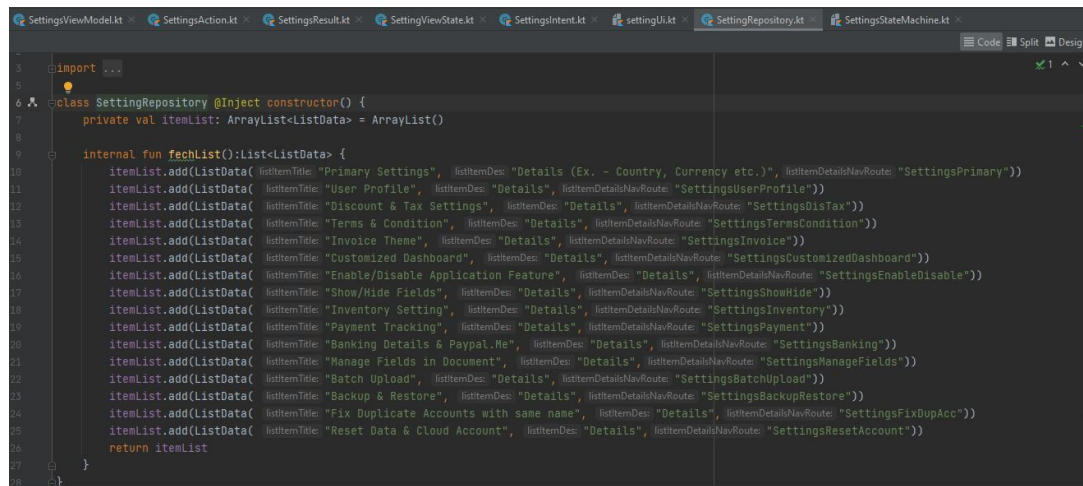
▼

Figure 5.7: Add Sale Details

Page 22 of 37

Back end Input

Setting option List



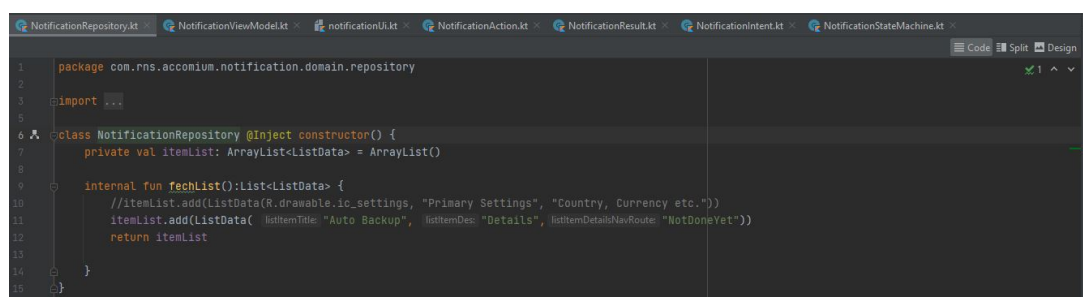
```

1  import ...
2
3  class SettingRepository @Inject constructor() {
4      private val itemList: ArrayList<ListData> = ArrayList()
5
6      internal fun fetchList(): List<ListData> {
7          itemList.add(ListData( itemListTitle: "Primary Settings", itemListDes: "Details (Ex. - Country, Currency etc.)", itemListDetailsNavRoute: "SettingsPrimary"))
8          itemList.add(ListData( itemListTitle: "User Profile", itemListDes: "Details", itemListDetailsNavRoute: "SettingsUserProfile"))
9          itemList.add(ListData( itemListTitle: "Discount & Tax Settings", itemListDes: "Details", itemListDetailsNavRoute: "SettingsDisTax"))
10         itemList.add(ListData( itemListTitle: "Terms & Condition", itemListDes: "Details", itemListDetailsNavRoute: "SettingsTermsCondition"))
11         itemList.add(ListData( itemListTitle: "Invoice Theme", itemListDes: "Details", itemListDetailsNavRoute: "SettingsInvoice"))
12         itemList.add(ListData( itemListTitle: "Customized Dashboard", itemListDes: "Details", itemListDetailsNavRoute: "SettingsCustomizedDashboard"))
13         itemList.add(ListData( itemListTitle: "Enable/Disable Application Feature", itemListDes: "Details", itemListDetailsNavRoute: "SettingsEnabledisable"))
14         itemList.add(ListData( itemListTitle: "Show/Hide Fields", itemListDes: "Details", itemListDetailsNavRoute: "SettingsShowHide"))
15         itemList.add(ListData( itemListTitle: "Inventory Setting", itemListDes: "Details", itemListDetailsNavRoute: "SettingsInventory"))
16         itemList.add(ListData( itemListTitle: "Payment Tracking", itemListDes: "Details", itemListDetailsNavRoute: "SettingsPayment"))
17         itemList.add(ListData( itemListTitle: "Banking Details & Paypal.Me", itemListDes: "Details", itemListDetailsNavRoute: "SettingsBanking"))
18         itemList.add(ListData( itemListTitle: "Manage Fields in Document", itemListDes: "Details", itemListDetailsNavRoute: "SettingsManageFields"))
19         itemList.add(ListData( itemListTitle: "Batch Upload", itemListDes: "Details", itemListDetailsNavRoute: "SettingsBatchUpload"))
20         itemList.add(ListData( itemListTitle: "Backup & Restore", itemListDes: "Details", itemListDetailsNavRoute: "SettingsBackupRestore"))
21         itemList.add(ListData( itemListTitle: "Fix Duplicate Accounts with same name", itemListDes: "Details", itemListDetailsNavRoute: "SettingsFixDupAcc"))
22         itemList.add(ListData( itemListTitle: "Reset Data & Cloud Account", itemListDes: "Details", itemListDetailsNavRoute: "SettingsResetAccount"))
23         return itemList
24     }
25 }

```

Figure 5.8: Setting option List (Settings Repository)

Notification Option List



```

1  package com.rns.accomium.notification.domain.repository
2
3  import ...
4
5  class NotificationRepository @Inject constructor() {
6      private val itemList: ArrayList<ListData> = ArrayList()
7
8      internal fun fetchList(): List<ListData> {
9          //itemList.add(ListData(R.drawable.ic_settings, "Primary Settings", "Country, Currency etc.))
10         itemList.add(ListData( itemListTitle: "Auto Backup", itemListDes: "Details", itemListDetailsNavRoute: "NotDoneYet"))
11         return itemList
12     }
13 }

```

Figure 5.9: Notification Option List (Notifications Repository)

5.4.2 Output

Our Project is now in under constructions. i am showing some output here.

Added Accounts & Customer/Supplier List

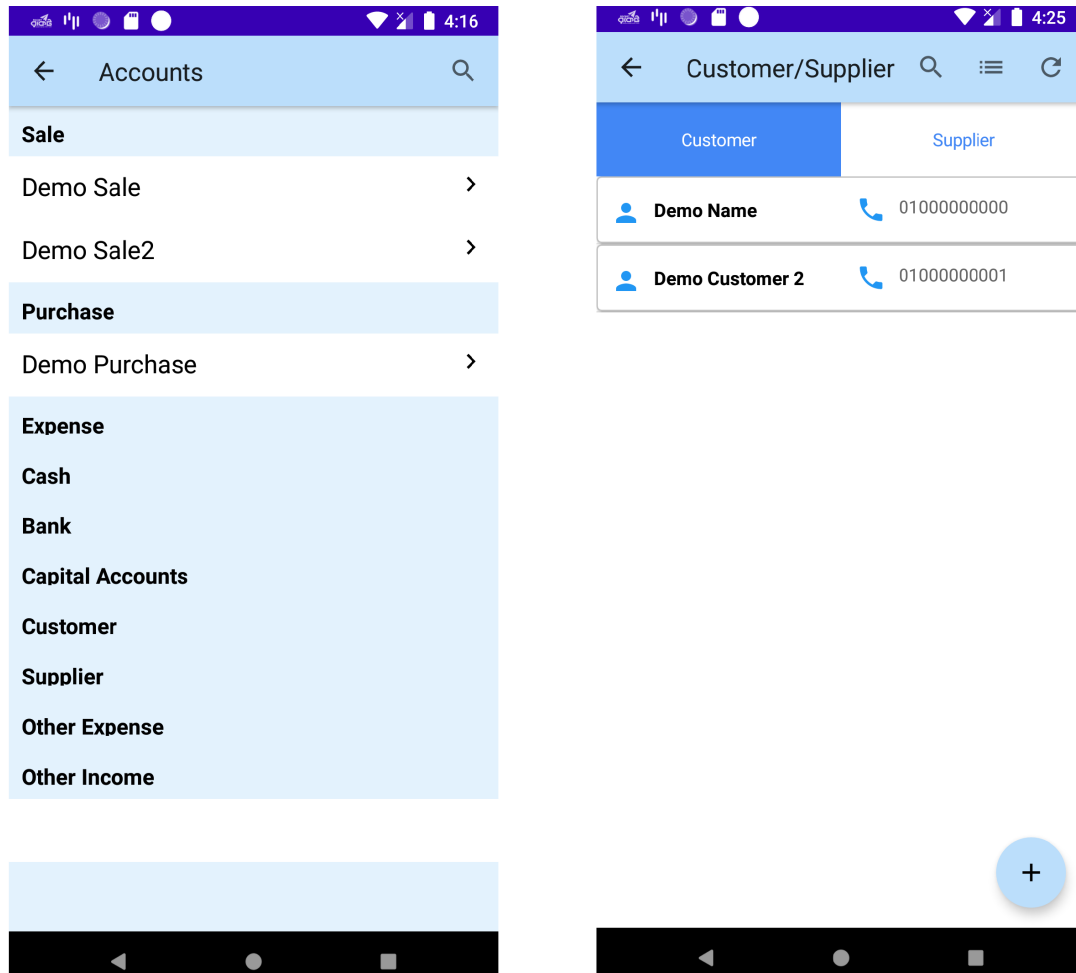


Figure 5.10: Added Accounts & Customer/Supplier List

Added Product List

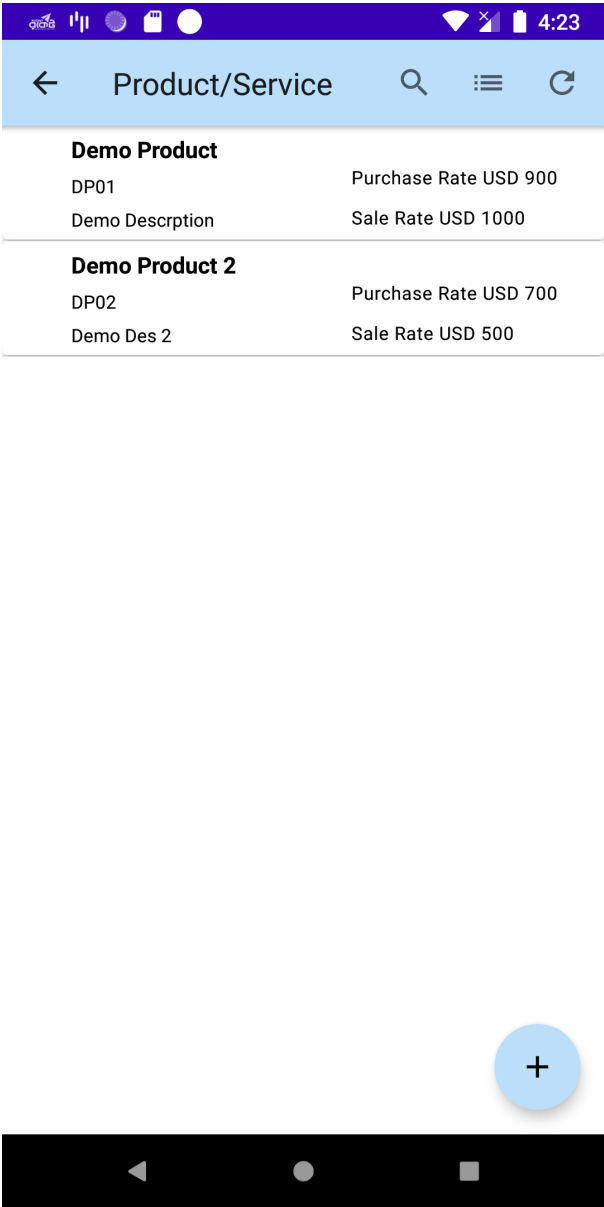


Figure 5.11: Added Product List

Added Product List Setting & Notification Option List

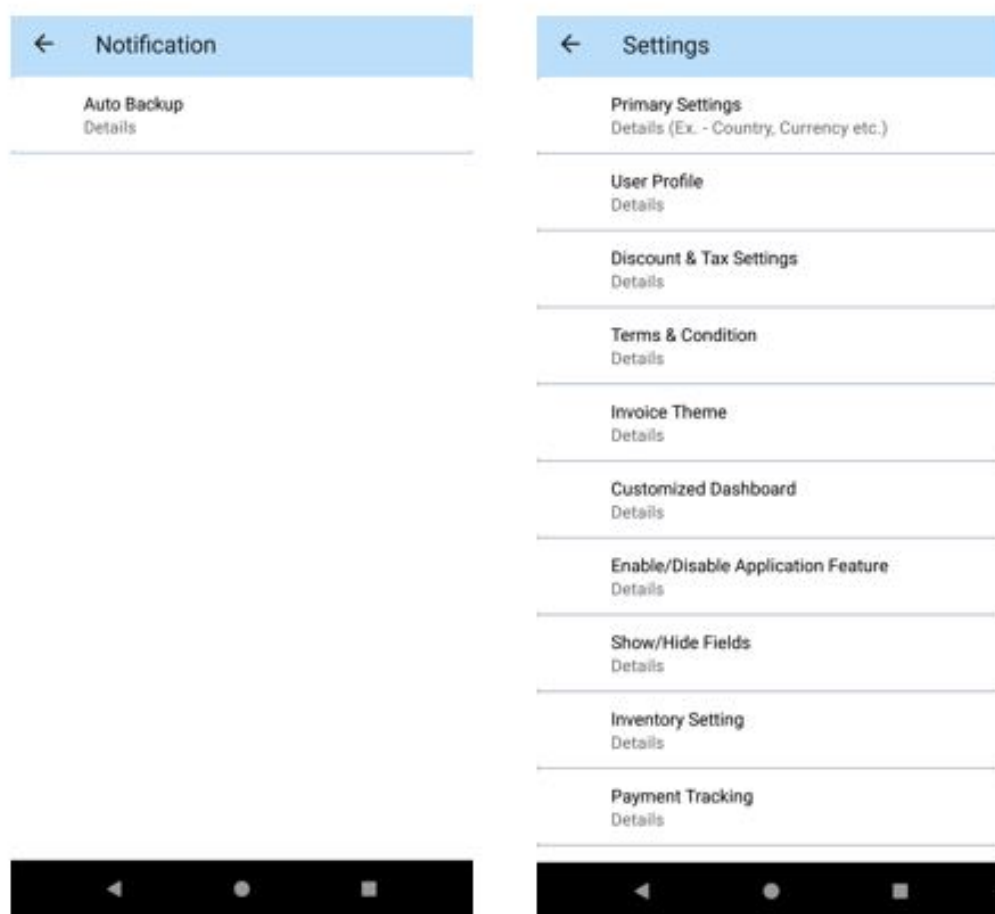


Figure 5.12: Setting & Notification Option List

5.4.3 Architecture

For building this project we are using Android famous mvi (Model – View – Intent) Architecture for front end and Clean Architecture for back end .

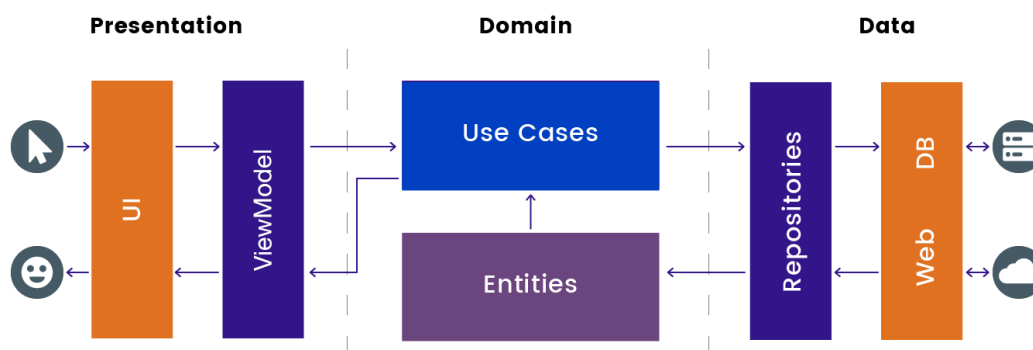


Figure 5.13: Android Clean Architecture

MVI

MVI stands for Model-View-Intent. This pattern has been introduced recently in Android. It works based on the principle of unidirectional and cylindrical flow.

Model: Unlike other patterns, In MVI Model represents the state of the UI. UI might have different states like Data Loading, Loaded, Change in UI with user Actions, Errors, User current screen position states. Each state is stored as similar to the object in the model.

View: The View in the MVI is our Interfaces which can be implemented in Activities and fragments. It means to have a container which can accept the different model states and display it as a UI.

Intent: The result of the user actions is passed as an input value to Intents. In turn, we can say we will be sending models as inputs to the Intents which can load it through Views.[5]

Clean Architecture

Presentation: which includes the UI, Presenter, and ViewModels

Domain: which includes Entities and Interactors

Data: which includes Databases and Rest clients[6]

Chapter 6

Results & Analysis

Dependency Library Injection

We injected dependency libraries successfully to create our project working environment. The below Figure shows those dependency libraries:

```
dependencies {  
  
    implementation "androidx.navigation:navigation-compose:2.4.0-alpha05"  
  
    implementation("androidx.compose.ui:ui:1.0.1")  
    // Tooling support (Previews, etc.)  
    implementation("androidx.compose.ui:ui-tooling:1.0.1")  
    // Foundation (Border, Background, Box, Image, Scroll, shapes, animations, etc.)  
    implementation("androidx.compose.foundation:foundation:1.0.1")  
    // Material Design  
    implementation "androidx.compose.material:material:1.0.1"  
    // Material design icons  
    implementation("androidx.compose.material:material-icons-core:1.0.1")  
    implementation("androidx.compose.material:material-icons-extended:1.0.1")  
  
    implementation 'androidx.core:core-ktx:1.5.0'  
    implementation 'androidx.appcompat:appcompat:1.3.1'  
    implementation 'com.google.android.material:material:1.4.0'  
    implementation 'androidx.lifecycle:lifecycle-runtime-ktx:2.3.1'  
    implementation 'androidx.activity:activity-compose:1.3.1'  
    testImplementation 'junit:junit:4.+'  
    androidTestImplementation 'androidx.test.ext:junit:1.1.2'  
    androidTestImplementation 'androidx.test.espresso:espresso-core:3.3.0'  
    androidTestImplementation "androidx.compose.ui:ui-test-junit4:$compose_version"  
  
    //login  
  
    def arrow_version = "0.10.4"  
    implementation "io.arrow-kt:arrow-core:$arrow_version"  
    implementation "io.arrow-kt:arrow-syntax:$arrow_version"  
    implementation "io.arrow-kt:arrow-meta:$arrow_version"  
  
    //hilt  
    implementation 'com.google.dagger:hilt-android:2.37'  
    kapt 'com.google.dagger:hilt-android-compiler:2.37'  
  
    // When using Kotlin.  
    implementation "androidx.hilt:hilt-lifecycle-viewmodel:1.0.0-alpha03"  
    kapt "androidx.hilt:hilt-compiler:1.0.0"  
    implementation "androidx.hilt:hilt-navigation-compose:1.0.0-alpha03"  
  
    def lifecycle_version = "2.3.1"  
    // ViewModel  
    implementation("androidx.lifecycle:lifecycle-viewmodel-ktx:$lifecycle_version")  
    implementation("androidx.lifecycle:lifecycle-viewmodel-compose:1.0.0-alpha07")  
    implementation("androidx.lifecycle:lifecycle-runtime-ktx:$lifecycle_version")  
  
    //coroutine  
    implementation("org.jetbrains.kotlinx:kotlinx-coroutines-android:1.3.9")  
    implementation "androidx.lifecycle:lifecycle-extensions:2.2.0"  
  
    //Room  
    def room_version = "2.3.0"  
  
    implementation 'androidx.room:room-ktx:2.3.0'  
    implementation "androidx.room:room-runtime:$room_version"  
    annotationProcessor "androidx.room:room-compiler:$room_version"  
    kapt "androidx.room:room-compiler:$room_version"  
  
    // optional - Guava support for Room, including Optional and ListenableFuture  
    implementation "androidx.room:room-guava:$room_version"  
  
    // optional - Test helpers  
    testImplementation("androidx.room:room-testing:$room_version")  
  
    // optional - Paging 3 Integration  
    implementation("androidx.room:room-paging:2.4.0-alpha04")  
}
```

Figure 6.1: Dependency Libraries

Navigation & MVI

All of the views are navigated by Navigation host and controller. Also MVI (MODEL - View - Intent) implemented for all view.

Customer & Supplier List

Here user can add his customer and supplier by press save with information. Then they can see their customer and supplier list. This data will store in remote and local database in a synchronized way. They can use those customer and supplier data when they sale or purchase any product or service. But our website for this project is under contraction so we used here only local database.

The figure displays two screenshots of a mobile application interface for managing customers and suppliers.

Left Screenshot: Add Customer/Supplier Form

- Header:** Add Customer/Supplier
- Account Type:** Customer (dropdown menu)
- Customer Name:** Demo Customer
- Email ID:** demo@mail.com
- Address:** Demo Address
- Contact No.:** 01500000000
- Contact Person:** CTO
- Shipping address:** Demo address
- TAX ID:** 1323-456
- Business Details:** Demo
- Buttons:** Save (blue), Cancel (red)

Right Screenshot: Customer/Supplier List

- Header:** Customer/Supplier (with search, menu, and refresh icons)
- Filter:** Customer (selected), Supplier
- List Items:**
 - Demo Name:** 01000000000
 - Demo Customer 2:** 01000000001
- Bottom Right:** Plus (+) button for adding new entries

Figure 6.2: Customer & Supplier List

Accounts List

Here user can add Account with some required user data by press save. Then they can see their account which will separated by the account type. User can create and use different Account for different activity which show them details. This data will use for user further activities

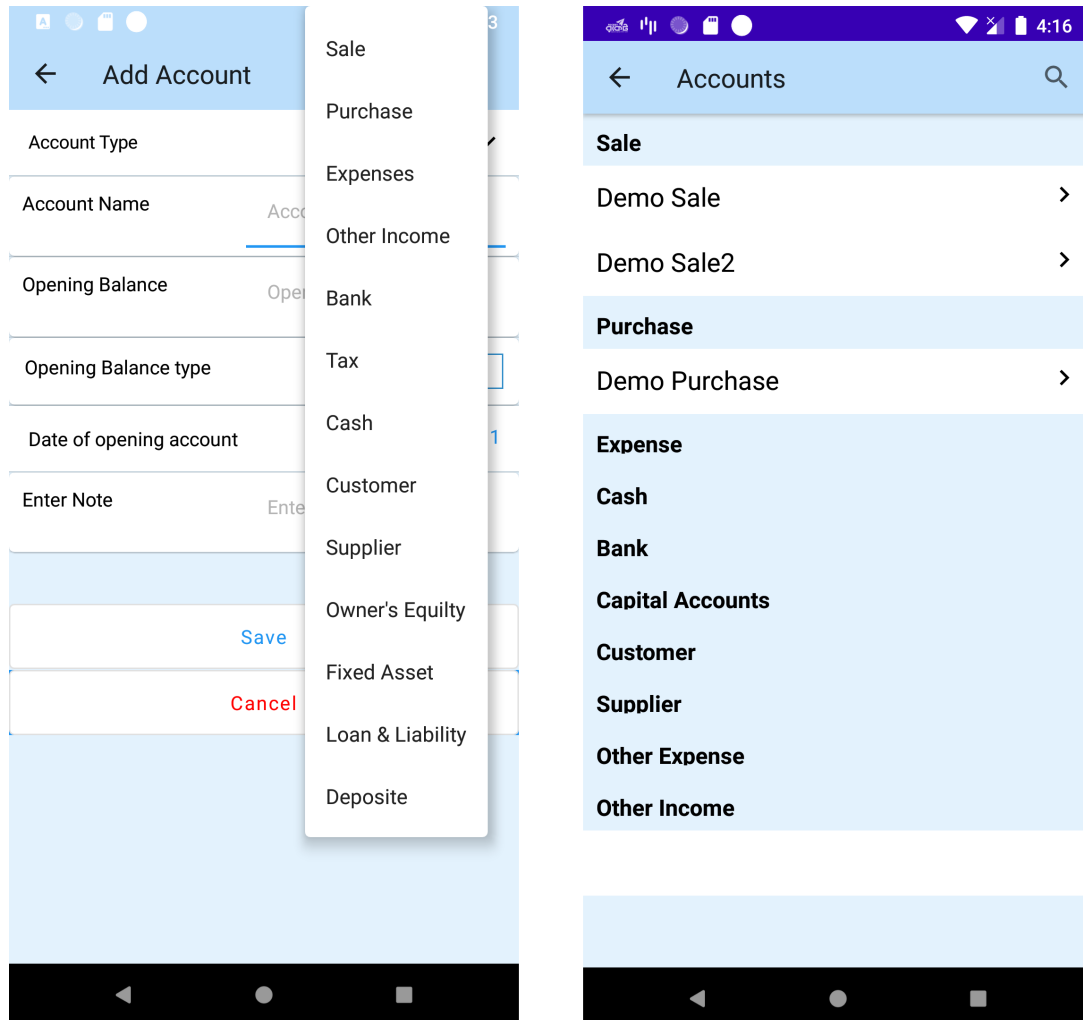


Figure 6.3: Accounts List

Product List

Here user can add product with product details by press save which product user will use for sale. Then they can see their product list. This data will use for users further activities (example : Sale product).

The figure consists of two side-by-side screenshots of a mobile application interface.

The left screenshot shows the 'Add Product' screen. It has a light blue header with a back arrow and the title 'Add Product'. Below the header is a form with six rows, each with a label and a value:

| | |
|----------------------|--------------------------------------|
| Product/Service name | Product/Service name Demo Product |
| Product/Service Code | Product/Service Code DP01 |
| Measurement Unit | Measurement Unit KG |
| Sale Rate | Sale Rate 1000 |
| Purchase Rate | Purchase Rate 900 |
| Description | Description Demo Description |

Below the form are two buttons: 'Save' (blue text) and 'Cancel' (red text). At the bottom right, there is a large blue circle with a white plus sign.

The right screenshot shows the 'Product/Service' list screen. It has a light blue header with a back arrow, the title 'Product/Service', and icons for search, menu, and refresh. Below the header is a list of products:

| | |
|-----------------------|-----------------------|
| Demo Product | |
| DP01 | Purchase Rate USD 900 |
| Demo Description | Sale Rate USD 1000 |
| Demo Product 2 | |
| DP02 | Purchase Rate USD 700 |
| Demo Des 2 | Sale Rate USD 500 |

At the bottom right, there is a large blue circle with a white plus sign.

Figure 6.4: Product List

Add Sale List

Here user can add Sale detail by selecting Customer, product and Account . This page will calculate the total price of sale with provided extra charges and discount amount.

The screenshot displays the 'Add Sale Details' screen in a mobile application. At the top, there's a blue header with a back arrow and the title 'Add Sale Details'. Below this, the form is organized into several sections. The first section contains 'Invoice' (01-09-2021) and 'Due Date' (23-09-2021). The second section has a checkbox for 'Do Not Provide Line Item' and a dropdown for 'Record Sale In' (Sale Account). The third section includes 'Po Number' and 'PO Date' (DD-MM-YYYY). Below these is a 'Customer Name' field with 'Demo Customer 2'. A 'Select Line Items' section shows a product list with 'Demo Product 2' selected, a quantity of 500, and a price of USD 0. A 'Discount' field shows USD 0.00 and an 'Other Charges' field shows USD 0. The bottom navigation bar is visible.

Figure 6.5: Add Sale

Chapter 7

Project as Engineering Problem Analysis

7.1 Sustainability of the Project/Work

One of the important part of a project is sustainability. Sustainability of project mean maintain the application or software or monitoring the project when get any problem try to fixed those problem. This project Accomium accounting system is now under developing but after launching the app the app will be maintain by company regularly. Also the team will find the new features new technologies to implement those thing in this project. The team will also monitor the customer review and depend on the customer expectation team fixed if any bug report found or update those features.

7.2 Social and Environmental Effects and Analysis

Social

vspace.5cm Main goal of this project to make easy of a small business accounting. It help to maintain the accounting calculation, syncs all financial data and also It instantly generates key financial reports. Sometime paper lost with the accounting calculation, transaction money history etc. Maintaining tax and payment due date manually is very difficult and people have to pay fine or it will harmed relation between people. Using this app user can easily maintain their accounting calculation, it will alert user when they have to pay or have to collect their due payment also tax. This app will help to startup company because user can use most of the feature of this app freely so it will help to their business grow.

Environmental

We know in small business most of the time paper used for accounting calculation and store information for long time. So it is bad for environment use of paper. By this application it will reduce the use of paper because every calculation will done by this app and user can easily store their desire data with this app . This app will eco-friendly. This app will not affect the environment.

7.3 Addressing Ethics and Ethical Issues

User privacy and private information is most valuable thing. Now the cybercrime rate increasing day by day. So before developing any app company have to think about the security for protect user data and maintenance.

All of user data for this app will store in a cloud database using Maria DB. Cloud database is one of the best for storing data safely. Hacking or chance something unethical thing happened is very low because cloud database has their own security to protect stored data and always concerned about those thing.

Because of user data privacy back end developer work using a demo server. After finishing the work only senior leader only accessed for cloud server. benefits of using demo server is their is no chance of user data lost and also no need to worry about security.

Chapter 8

Lesson Learned

8.1 Problems Faced During this Period

Work from home in during covid'19 pandemic is very difficult thing. In my internship period I faced lots of problem and its make more difficult when I work with some new thing. We can't maintain the schedule of project properly for the lockdown because of our one teammate stuck because of lockdown and it is caused for some delay . Sometime communication gap happened which also causes delay. Jetpack compose, Android Architecture and room database is very new thing for me so I faced very trouble with this project . Online resource is very less because those are very new.

8.2 Solution of those Problems

During internship I gained many new knowledge, skills. I learned about googles new modern UI toolkit Jetpack compose . Learned kotlin language which is similar to Java. i also gained some knowledge about various kind of android architecture and how to build an app by using those architecture . I analysis many project from github repository and watch many YouTube tutorial to complete my task in my project when I faced some trouble I talk with my supervisor who is our team leader of project. He gave me very much support and also helped me a lot to sort out my problems. The best part during my internship period my supervisor always enforce me to learn new thing. Our project will contact with a dedicated website by using GraphQL API although I didn't work with it but i took some idea about it . I complete some back end database part with android sqlite room database. After connecting with website the local database will synchronized with remote database.

Chapter 9

Future Work & Conclusion

9.1 Future Works

This project is now in under development so company have plan to connect this app with a dedicate website with graphQL api and adding many new features. company building this app for marketing purpose so they have plan to add new thing and making app smooth for user to get some positive impressions. Also Application can be updated in near future as and when requirement for the same arises, as it is very flexible in terms of expansion. Company will maintain and update this app regularly.

9.2 Conclusion

It has been a great opportunity for me to work as a mobile app developer during internship at the startup software solution company Accomium Software Company. This three-month internship program was like a step stone to my carrier in the job market. During internship I learned many important knowledges about IT sector. The internship program helped me to gain important knowledge about how the IT sector shaping in our country. Internship at Accomium taught me how to handle, recent software related technologies, frequently used technologies and best used of this technology. This program gave me a clear idea about professional life as a mobile application developer, what I must face and how to handle those situations. During internship, I tried to cover my weakness about mobile app development and tried to become a better mobile app developer.

Bibliography

- [1] R. Rupnik, “Mobile Application Development Methodology..” <https://www.igi-global.com/chapter/handbook-research-mobile-business-second/19540>, 2009. Accessed: 2021-08-06.
- [2] L. T. Mark Kraeling, “User Requirement..” <https://www.sciencedirect.com/topics/engineering/user-requirement>, 2019. Accessed: 2021-08-06.
- [3] A. for Developers, “Designs for Android..” <https://developer.android.com/design?hl=AR>. Accessed: 2021-08-06.
- [4] D. Hulett, “Project Schedule Assessment Using Best Scheduling Practices.” https://www.projectrisk.com/project_schedule_assessment.html. Accessed: 2021-08-21.
- [5] S. Vasudev, “MVI Architecture - Android Tutorial for Beginners - Step By Step Guide..” <https://blog.mindorks.com/mvi-architecture-android-tutorial-for-beginners-step-by-step-guide>. Accessed: 2021-08-19.
- [6] D. M. Anastasia D, “Clean Architecture of Android Apps with Practical Examples..” <https://rubygarage.org/blog/clean-android-architecture>. Accessed: 2021-08-19.