

Development of Smart Pharmacy App for Binary-IT

A Practicum Report Submitted By

Md. Rifat Ahmed Rony

ID: 18103088

In Partial Fulfillment of the Requirement for the Award of

Bachelor of Computer Science and Engineering



Department of Computer Science and Engineering

College of Engineering and Technology

IUBAT – International University of Business Agriculture and Technology

Summer 2021

Development of Student Mentorship Application for Binary-IT

Md. Rifat Ahmed Rony

ID: 18103088

A practicum report submitted in partial fulfillment of the requirements for the degree of
Bachelor of Computer Science and Engineering (BCSE)

The practicum has been examined and approved,

Prof Dr Utpal Kanti Das

Chairman

Dept. of Computer Science and Engineering

Dr Hasibur Rashid Chayon

Co-supervisor, Coordinator

Associate Professor

Dept. of Computer Science and Engineering

Toyeer-E-Ferdous

Supervisor

Assistant Professor

Dept. of Computer Science and Engineering

Fall 2021

Letter of Transmittal

4 December 2021

The Chair

Practicum Defense Committee

Department of Computer Science and Engineering

IUBAT–International University of Business Agriculture and Technology

4 Embankment Drive Road, Sector 10, Uttara Model Town

Dhaka 1230, Bangladesh

Subject: Letter of Transmittal.

Dear Sir,

It is my pleasure to submit here with my dissertation, which has been prepared under the sound and dynamic leadership of a personality like you.

This paper is about “Smart Pharmacy App” for “Binary-IT”. I have tried my level best to collect the relative information as possible in preparing the report. During preparation of the report I have experienced practically a lot that will help me a great in my career. It has enlightened my practical knowledge. I will be able to explain anything for more clarification if necessary.

I would like to thank you, for giving me the opportunity to do a report on the above mentioned topic.

Yours sincerely,

.....

Md. Rifat Ahmed Rony

ID: 18103088

Dept. of Computer Science and Engineering

Letter of Authorization

4 December 2021

Md. Rifat Ahmed Rony

ID: 18103088

Dear Student,

I hereby authorized you to develop “Smart Pharmacy App” for “Binary-IT” to fulfill the requirement of the project of the Practicum Course.

I wish you to complete the project successfully as part of the completion of the course.

.....

Toyeer-E-Ferdous

Supervisor

Senior Lecturer,

Dept. of Computer Science and Engineering

IUBAT- International University of Business Agriculture and Technology.

Student's Declaration

I am Md. Rifat Ahmed Rony, bearing ID# 18103088, student of BCSE - Bachelor of Computer Science and Engineering program, under the College of Engineering and Technology (CEAT) of IUBAT- International University of Business Agriculture and Technology declaring that, this report on the topic of “Smart Pharmacy App for Binary-IT” has been prepared for the fulfillment of the internship CSC 490, Practicum as well as the partial requirement of BCSE-Bachelor of Computer Science and Engineering degree.

The report and the project on — “Smart Pharmacy App for Binary-IT” were originally prepared by me. All modules and procedures of this project are being made after proper inspection and internet information.

It has not been prepared for any other purposes, rewards or presentations.

Yours sincerely,

.....

Md. Rifat Ahmed Rony

ID: 18103088

Dept. of Computer Science and Engineering

Acknowledgement

First, I would like to thank the Almighty for His blessing. Without His concern nothing can be possible.

I would like to express my heartiest gratitude to my honorable supervisor, Toyeer-E-Ferdous, Senior Lecturer, Department of Computer Science and Engineering, International University of Business Agriculture and Technology, also my honorable co-supervisor Dr Hasibur Rashid Chayon, Coordinator and Associate Professor, Department of Computer Science and Engineering, International University of Business Agriculture and Technology, for his guidance, encouragement, motivation and support to prepare this report by spending his valuable time to review and evaluate this work.

I am very grateful to the Department of Computer Science and Engineering (CSE) of IUBAT—International University of Business Agriculture and Technology for providing their all-out support during the work. I would like to thank our Professor & Chair Dr. Utpal Kanti Das, Department of Computer Science and Engineering.

Finally, I express my gratitude to my parents and classmates for always being motivating and supportive.

Supervisor's Certification

This is to certify that the Practicum report on “**Smart Pharmacy App for Binary-IT**” has been carried out by Md. Rifat Ahmed Rony ID# 1810308 of IUBAT – International University of Business Agriculture and Technology as a partial fulfillment of the requirement of practicum defense course. The report has been prepared under my guidance and is a record of the accomplished work carried out successfully. To the best of my knowledge and as per his declaration, no parts of this report have been submitted anywhere for any degree, diploma or certification.

Now he is permitted to submit the report. I wish him success in all his future endeavors.

Supervisor

.....

Toyeer-E-Ferdous

Senior Lecturer,

Department of Computer Science and Engineering

IUBAT- International University of Business Agriculture and Technology

Departmental Certification

On behalf of the Department of Computer Science and Engineering of International University of Business Agriculture and Technology (IUBAT) we, the undersigned, certify that this practicum report “**Smart Pharmacy App for Binary-IT**” for the award of Bachelor of Computer Science and Engineering (BCSE) degree was duly presented by Md. Rifat Ahmed Rony (ID No. 18103088) and accepted by the department.

Prof Dr Utpal Kanti Das

Chairman

Dept. of Computer Science and Engineering

Dr Hasibur Rashid Chayon

Co-supervisor, Coordinator

Associate Professor

Dept. of Computer Science and Engineering

Toyeer- E-Ferdous

Supervisor

Senior Lecturer,

Dept. of Computer Science and Engineering

Organization Certificate

Table of Contents

Letter of Transmittal	3
Letter of Authorization	4
Student's Declaration	5
Acknowledgement	6
Supervisor's Certification	7
Departmental Certification	8
Organization Certificate	9
Table of Contents	10
List of Figure	Error! Bookmark not defined.
List of Table	14
Chapter 1: Project Introduction	16
1.1 Introduction	16
1.2 Background of Study	17
1.3 Objectives	17
1.3.1 Broad Objectives	17
1.3.2 Specific Objectives	17
1.4 Proposed System's Benefits	18
1.5 Methodology	18
1.5.1 Data Sources	18
1.6 Limitation	18
1.7 Process Model	19
1.7.1 Reason for Choosing	19
	10

1.8 Feasibility Study	20
1.8.1 Technical Feasibility	20
1.8.2 Economic Feasibility	21
1.8.3 Operational Feasibility	21
Chapter 2: The Organization	22
2.1 Organizational Overview	22
2.2 Organization Services	23
2.3 Organization Vision	23
2.4 Organization Mission	23
2.5 Organization Clients	23
2.6 My position in the Organization	24
2.7 Organization Location	24
2.8 Organization Structure	24
Chapter 3: Requirement Engineering	26
3.1 Requirement Analysis	27
3.2 Requirement Engineering	27
3.2.1 User Requirement	28
3.2.2 System Requirement	29
3.2.3 Functional Requirement	31
3.2.4 Non-Functional Requirement	32
3.2.5 Hardware Requirement	32
3.2.6 Software Requirement	32
3.3 Use Case Diagram of the System	33
3.3.1 Use case symbol	33
3.3.2 Use case diagram	35
3.3.3. Use Case text	35
Chapter 4: System planning	36
4.1 Scope of Project	37
4.2 Function of Proposed System	37
4.2.1 Function description	38
4.3 System project Planning	40

4.3.1 System Project Estimation	41
4.3.2 Function Oriented Metrics	41
4.4 Identifying Complexity	45
4.4.1 Identifying Complexity of transition function	45
4.4.2 Identifying Complexity of Data function	45
4.4.3 Unadjusted function for transition function	46
4.4.4 Unadjusted function for data function	47
4.4.5 Performance and Environmental impact	48
4.4.6 Calculating Function Point	48
4.5 Process Based Estimation	49
4.5.1 Effort Distribution	49
4.5.2 Task Schedule	49
4.6 Cost Estimation	50
4.6.1 Personal Cost	50
4.6.2 Hardware cost	51
4.6.3 Software Cost	52
4.6.4 Others cost	52
4.6.5 Account Table	53
Chapter 5: Risk Engineering	54
5.1 Risk Management	55
5.1.1 Stage of Risk	55
5.1.2 Types of Risk	56
5.2 The RMMM plan	57
Chapter 6: Analysis Modeling	61
6.1 Software Analysis Pattern	62
6.2 Activity Diagram	62
Chapter 7: Designing	67
7.1 Data flow diagram	68
7.1.1 Context Level Diagram	70
7.1.2 Level 1 Data Flow Diagram	71
7.1.3 Level 2 DFD of Process 1(Login)	72
	12

7.1.4 Level 2 DFD of Process 2(Register)	72
7.1.5 Level 2 DFD of Process 3	73
7.1.6 Level 2 DFD of Process 4	73
7.1.7 Level 2 DFD of Process 5	74
7.1.8 Level 2 DFD of Process 6	74
7.1.9 Level 2 DFD of Process 7	75
7.1.10 Level 2 DFD of Process 8	75
7.1.11 Level 2 DFD of Process 9	76
7.1.12 Level 2 DFD of Process 10	77
7.1.13 Level 2 DFD of Process 11	78
7.1.15 Level 2 DFD of Process 12	78
7.1.16 Level 2 DFD of Process 13	79
7.1.17 Level 2 DFD of Process 14	79
7.2 Database Design	80
7.2.1 Entity Relationship Diagram	80
7.2.2 Database Design	81
7.3 Interface Design	87
Chapter 8: System Quality and Testing	115
Chapter 9: Conclusion	119

No table of figures entries found.

List of Table

Chapter 1: Project Introduction

1.1 Introduction

Smart Pharmacy App (SPA) is a management system app that is used to improve the quality for managing any type of pharmacy shop. The main aim of the project is the management of the database of the pharmaceutical shop. This project is an insight into the design and implementation

of a Pharmacy Management System. This is done by creating a database of the available medicines in the shop. It is a mobile (Android) based system which helps the Pharmacist to improve inventory management, cost, medical safety etc.

1.2 Background of Study

Smart Pharmacy Application is an Android Application Project which can be used by any Pharmacy shop. In my project there will be only one User (Admin) for an app. Here in my project the Pharmacy Shop owner (Admin) can Add Medicine, can Purchase Medicine after purchase it will be available in stock, user can sell medicine and at the same time the stock will decrease. While the admin will sell medicine there will be a customer panel where users can see to whom he sells medicine. Users can keep any type of note if needed. Admin can also the list of medicine which will expire soon.

1.3 Objectives

1.3.1 Broad Objectives

The broad objective of this project is to use my academic knowledge in the real life working environment by developing a Smart Pharmacy App.

1.3.2 Specific Objectives

- Online based Android Application.
- Authentication for Admin through Firebase Authentication.
- Admin can Purchase medicine.
- Admin can sell medicine.
- Admin can see the stock medicine list.

- Admin can manage accounts.
- Admin can keep note.
- Admin can see the list of expired soon medicine.
- Admin can manage the customer

1.4 Proposed System's Benefits

- Here the user can add the list of medicine that he want to store on his shop.
- He can manage the stock.
- He can see the daily sale.
- Can manage the list of medicine that will expire soon.
- All the information is safe and protected.

1.5 Methodology

1.5.1 Data Sources

- **Primary Data Sources** – Primary data has been generated within the organization.
- **Secondary Data Sources** – Secondary Data has been collected through the real-life experience, studying some article and information are collected through the internet. Sources provide better understanding about the project. For that I Search for some online system.

1.6 Limitation

- **Data backup** - There is no data backup way. Once a storage device is crashed, all the data will be lost.
- **Security**–The system is not protected for CSRF attack. But if we use HTTPS connection then we can mitigate this issue.

1.7 Process Model

In my project I am using the Incremental Model. Incremental Model is a process of software development where requirements are broken down into multiple standalone modules of the software development cycle. Each iteration passes through the **requirements, design, coding and testing phases**. And each subsequent release of the system adds function to the previous release until all designed functionality has been implemented. That is why I have chosen this type of process Model.

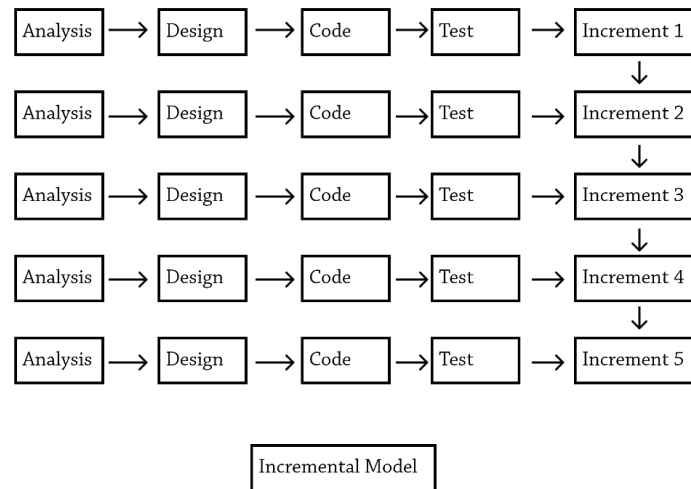


Figure 2. 1: Incremental Process Model

1.7.1 Reason for Choosing

I have chosen to follow the Incremental Process Model for the following reason,

- Errors are easy to identify.
- In this model users can respond to each build.
- Lowers initial delivery cost.
- Easier to test and debug.

- Simply manage risk because it is handled during its iteration.
- We can generate software quickly during the software development life cycle.
- This is flexible and it is less expensive to change the requirements and scope.
- The client gets important functionality early.
- Lower risk of project failure.
- Though the development stages changes can be done.

1.8 Feasibility Study

Feasibility studies determine whether this system will be feasible or achievable for the organization or not.

There are three major areas of feasibility study

- Technical feasibility
- Economic feasibility
- Operational feasibility

1.8.1 Technical Feasibility

The technical feasibility focused on gaining an understanding of the present technical resources of the organization and their applicability to the expected needs of the proposed system. I find out that this system is technically feasible. Because in this project I need a high level programming language and in my case I use JAVA, Android Studio as code editor, I used firebase real-time database for store data and use firebase authentication for Authorized any use. In my project I used my phone to run the app but the condition is Android Version should be Android 6 (Marshmallow) or more then 6.

1.8.2 Economic Feasibility

Economic feasibility determines to what extent a new system, is cost effective. My software is economically feasible. For my project I just need a pc, IDE, and android device for checking. So here the cost is less compare with others. So I can say that my system is economically feasible.

1.8.3 Operational Feasibility

The proposed system is designed from only admin point of view. The proposed system is exactly similar to the manual system. The system is developed in a user-friendly way so that the admin can easily use it. User doesn't requires any special training for operating the system. Anyone who have a knowledge about Android phone he can use this system. As this is also support English and Bangla Both Language so the user can easily operate the system. During the implementation of the system, it was found that users are very much interested and eager to this software. So, the system is operationally feasible.

Chapter 2: The Organization

2.1 Organizational Overview

Binary-IT is a multi-layered Bangladesh based software company that focuses highly qualitative, timely delivered and cost-effective Software development.

2009 February.

2.2 Organization Services

Our services consist of-

- Software development
- Mobile Application

2.3 Organization Vision

To become customer-oriented best ITES Company in Bangladesh.

2.4 Organization Mission

Mission of the organization is to provide IT products and services focusing on customer requirement and exceed the expectation always. Organizations main aim is to adopt the latest technology and blend with innovation to develop exceptional software. Organization is committed to providing astonishing service to our customers.

2.5 Organization Clients

1. Islamic Foundation.
2. Dhaka North City Corporation- DNCC.
3. Mohammadi Steel Works Limited.
4. Water development board.
5. Naori Adarsha College.



Figure 2.1: Organization Client

2.6 My position in the Organization

I am an Intern of project management section of this organization. I am doing my internship by a supervisor in this organization. For me it's a new experience to maintain office time. I am also maintaining the other rules and regulations of this organization. I am sure it'll create great impact in my future career.

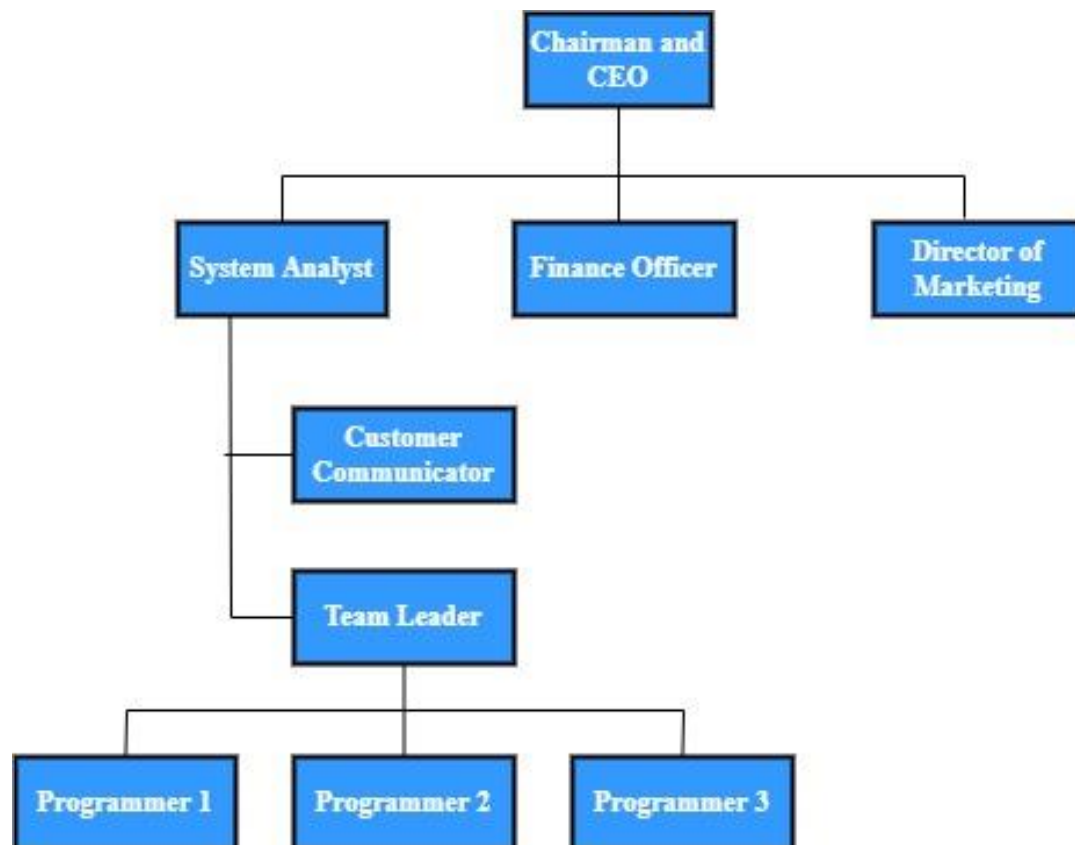
2.7 Organization Location

Tropical Alauddin Tower

Room # 11E, Sector 3, Uttara

Dhaka-1230

2.8 Organization Structure



Chapter 3: Requirement Engineering

Chapter 3 combines all the requirements for developing the project. In this chapter detail of each of the requirements is discussed along with a use case diagram for better understanding of the

project and matching the various requirements of the projects to be implemented.

3.1 Requirement Analysis

Requirements analysis is a software engineering task that bridges the gap between system engineering and system design. Requirements analysis allows the software engineer to define the software allocation and build the module of the data, functional and behavioral domains that will be treated by software. Requirement analysis provides the software designer with a representation of information, function and behavior of the system.

Requirements analysis is the first stage in the software development process. It encompasses those tasks that go into determining the needs or conditions to meet for a new or altered product taking account of the possibly conflicting requirements of the various stakeholders such as beneficiaries or users. Analysis of the requirement is critical to the success of a development project. Requirements must be actionable, measurable, testable, related to identified business needs or opportunities, and defined to a level of detail sufficient for system design.

3.2 Requirement Engineering

Requirement engineering is the process of defining, documenting, and maintaining requirements in the engineering design process. There are many definitions of Requirements Engineering; but they all have the same idea that requirements involve finding out what people want from a computer system and understanding what their needs mean in terms of design. Requirement engineering is closely related to software engineering, which focuses more on the process of designing the system that users want.

- User requirements

- System requirements
- Functional requirements
- Non-functional requirements
- Hardware Requirements
- Software Requirements

3.2.1 User Requirement

1. Admin can Sign Up and Sign In.
2. Admin can add medicine.
3. Admin can update medicine.
4. Admin can delete medicine.
5. Admin can view medicine.
6. Admin can add, view, update, and delete Medicine Manufacture.
7. Admin can add, view, update, and delete Generic Name.
8. Admin can add, view, update, and delete Medicine Type.
9. Admin can purchase medicine.
10. Admin can sell medicine.
11. Admin can manage customers.
12. Admin can see the list of upcoming expired medicine.
13. Admin can view note.
14. Admin can add note.

3.2.2 System Requirement

1. Admin can Sign up and Sign In in the system

- 1.1 First admin will go to login panel.
- 1.2 Admin will sign up using their email, password.
- 1.3 Through firebase authentication he can create a new account.
- 1.4 They can login using email and password into system.

2. Admin can add medicine

- 2.1 First admin will have to login into the system
- 2.2 Admin can add medicine after giving all the requirement for adding medicine.
- 2.3 By clicking “Add Medicine” admin can add medicine

3. Admin can view medicine

- 3.1 First admin will have to login into the system.
- 3.2 Admin can view medicine list by clicking View Medicine button.

4. Admin can Update medicine

- 4.1 First admin will have to login into the system.
- 4.2 Then admin have to view the medicine list by clicking View Medicine button.
- 4.3 In view medicine there is an option for edit the medicine list.
- 4.4 After clicking in Edit there will be popup window for edit medicine

5. Admin can Delete medicine

- 5.1 First admin will have to login into the system.
- 5.2 Then admin have to view the medicine list by clicking View Medicine button.
- 5.3 In view medicine there is an option for delete the medicine list.
- 5.4 After clicking in Delete button there will be an alert dialog for delete medicine and for delete medicine user will have to confirm by clicking yes button.

6. Admin can add medicine manufacture

- 6.1 First admin will have to login into the system.

- 6.2 There will be an option of medicine.
- 6.3 Inside the medicine there will be an option of manufacture.
- 6.4 Here the admin can add medicine manufacture.

7. Admin can medicine type

- 7.1 First admin will have to login into the system.
- 7.2 There will be an option of medicine.
- 7.3 Inside the medicine there will be an option of Medicine type.
- 7.4 Here the admin can add medicine type.

8. Admin can add generic name

- 8.1 First admin will have to login into the system.
- 8.2 There will be an option of medicine.
- 8.3 Inside the medicine there will be an option of generic name.
- 8.4 Here the admin can add generic name.

9. Admin can purchase medicine

- 9.1 First admin will have to login into the system.
- 9.2 There will be an option of Purchase Medicine.
- 9.3 Here admin will have to give all the details to Purchase a medicine
- 9.4 Then simply click by Save to Purchase the medicine.

10. Admin can Sell medicine

- 10.1 First admin will have to login into the system.
- 10.2 There will be an option of Sell Medicine.
- 10.3 Here admin will have to give all the details to sell a medicine
- 10.4 Then simply click by Sell button to sell the medicine.

11. Admin can manage customer

- 11.1 First admin will have to login into the system.
- 11.2 There will be an option of Customer Medicine.

- 11.3 Admin can see the details of the customer whatever the customer buy.
- 11.4 Admin can edit or delete the customer details.

12. Admin can see the upcoming expire medicine list

- 12.1 First admin will have to login into the system.
- 12.2 There will be an option of Expire Medicine.
- 12.3 Admin can see the list of medicine which will be expire soon.

13. Admin can View note

- 13.1 First admin will have to login into the system.
- 13.2 There will be an option of Note.
- 13.3 Here admin will have to give title and description.
- 13.4 Then simply click by Save to Purchase the medicine.

14. Admin can add note

- 14.1 First admin will have to login into the system.
- 14.2 There will be an option of Note.
- 14.3 First he will see the note then there will be a floating action button for add note.
- 14.4 Here admin will have to give the title and description.
- 14.5 Then simply click on Save to save note.

3.2.3 Functional Requirement

- Login
- View all medicine
- Add medicine
- Edit medicine
- Delete medicine
- Add manufacture
- Add medicine type

- Add generic name
- Can purchase medicine
- Sell medicine
- Manage customer
- See list of upcoming expire medicine
- Can view note
- Can add note

3.2.4 Non-Functional Requirement

- Each user has his own account.
- User will have to login in the system by his email and password.
- User email and password will be saved in database.

3.2.5 Hardware Requirement

The hardware listed by no means a minimum requirement to run the system, but rather a base limit for running the system smoothly and comfortably. This is also considering the potential amount of traffic that may go through the server.

- Intel Core i5 Processor.
- 8 GB RAM.
- 256 GB SSD.
- Internet connection.
- Android Mobile Phone
- Android version 6.0(Marshmallow)

3.2.6 Software Requirement

1. Text Editor (Android Studio)

2. Frontend- XML (Extensible Markup Language)
3. Backend-Java
4. Designing Tools-Draw.io
5. Database-Firebase

3.3 Use Case Diagram of the System

3.3.1 Use case symbol

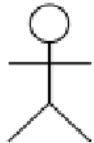

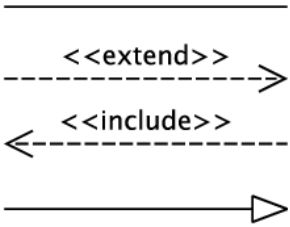
Symbol	Reference Name
	Actor
	Use case
	Relationship

Figure 3. 1: Use Case Diagram Symbols

Actor: An actor represents a set of roles that users of use case play when interacting with these use cases. Actors can be human or automated systems.



Use case: A use case represents a user goal that can be achieved by accessing the system or software application. A use case is the specification of a set of actions performed by a system, which yields an observable result that is typically of value for one or more actors of the system.



Association: Actor and use case can be associated to indicate that the actor participates in that use case. Therefore, an association corresponds to a sequence of actions between the actors and use case in achieving the use case.



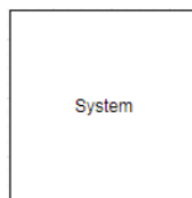
Include: An include relationship specifies how the behavior for the inclusion use case is inserted into the behavior defined for the base use case.



Extend: An extend relationship specifies how the behavior of the extension use case can be inserted into the behavior defined for the base use case.



System: The use cases of the system are placed inside the system shape, while the actor who interact with the system are put outside the system. The use cases in the system make up the total requirements of the system.



3.3.2 Use case diagram

3.3.3. Use Case text

Chapter 4: System planning

In this chapter I am going to discuss the purpose of the project, other functional and logical discussion and the scopes also. The major functional purpose is given there so that the ultimate purpose can be justified by this. The Counting Adjusted Function Point, Transaction Functions-Data Functions calculation are also here.

4.1 Scope of Project

A project scope is the clear identification of the work required to successfully complete or deliver a given project. Project scope control and management is often overlooked in the management ranks. Software scope determination is the first and most important activity in software project planning. The statement of software scope must bind. The software should have ability to manage the members, operators and package information's and the database. The project has given the power of re-usability. Present developers are hard to strict towards giving the system a fully dynamic come across.

4.2 Function of Proposed System

Function Name	Function Number
User Registration	F1
Login into the system	F2
Manage Medicine	F3
Manage Stock	F4
Purchase Medicine	F5
Sell Medicine	F6

Manage Customer	F7
Manage Account	F8
View Expire Soon Medicine	F9

4.2.1 Function description

Function description descriptive the function in details. It concerns on three factors: what is the possible input, possible output for a particular function and which table of the database uses by the function.

- **User Registration**

Input: E-mail address password.

Output: If user input email does not follow the syntax of email it will show an error message in email filed, if password is not correct or less than 6 number it will show an error. If user input email is already registered it will show **User is already registered**. Here will generate an UID automatically, and we will store this UID in firebase real-time database.

Used Database: Data will store in firebase Authentication.

- **Login into the system**

Input: E-mail & password

Output: If user give valid email address system will take him in Dashboard. If email and password do not match it will show error.

Used Database: Data will store in firebase Authentication.

- **Manage Medicine**

Input: medicine_name, manufacture_name, box_pattern, medicine_category, medicine_type, medicine_unit, generic_name, sell_price, manufacture_price, shelf_no.

Output: Admin will add the list of medicine which will he store in his medicine store.

Used Database: Under the firebase authentication UID there is a field of medicine and inside this medicine field there is field for **Medicine List**.

- **Manage Stock**

Input:

Output:

Used Database field:

- **Purchase Medicine**

Input: manufacture_name, medicine_name, date, payment_type, batch_id, expire_date, stock_quantity, purchse_quantity, manufacture_price, total_price, paid_amount, due_amount.

Output: Admin will be able to purchase medicine and stock will updated automatic.

Used database field: Purchase Medicine

- **Sell Medicine**

Input: customer_name, medicine_name, date, payment_type, stock_quantity, sell-quantity, unit_sell_price, total_price, paid_amount, due_amount.

Output: Admin can sell medicine to a customer and after sell stock will decrease automatic.

Used database field: Sell Medicine

- **Manage Customer**

Input: customer_name, medicine_name, total_medicine_price, total_paid, total_due;

Output: Admin can see the list of customer who buy medicine and if any due to pay.

Used database field: Customer Details

- **Manage Accounts**

Input:

Output:

Used database field: Accounts

- **View expire soon medicine list**

Input:

Output: Admin can see the medicine stock list which will expire in next 30 days.

Used database field: Stock Medicine.

4.3 System project Planning

Before starting any project, it is compulsory to estimate the work to be done, the resources that will be required, the time that will elapse from start to finish and to analyze the project to determine whether it is feasible or not. Software project planning is the second activity of CPF. Software project management commences with a set of activities that collectively called software project planning. Through the software project planning I estimate the work to be 33 done, the resources that will be required, and the time that will elapse from start to finish and finally I analyze the project to determine whether it is feasible or not. The following activities of software project planning that have followed in this project are:

- Function Point Estimation of the project
- Function Oriented Metrics
- Process Based Estimation
- Task Scheduling
- Project Schedule Chart
- Cost Estimation

4.3.1 System Project Estimation

The accuracy of a software project estimate predicated based on a number of things:

- Properly estimated the size of the product to build.
- The ability to translate the size estimation into human effort, calendar time and money.
- The degree to which the project plan reflects the abilities of the software team or engineer.
- The stability of the product requirements and the environment that supports the software engineering effort.

Software size estimation is the most important matter that I have to consider during the software project. If the software size not calculate properly, then this will cause various problems such as scheduling problems, budget problem etc. As the project goes on, before estimating the software size, I have to confirm that software scope is bounded.

4.3.2 Function Oriented Metrics

Function point-based estimation focuses on information domain values rather than software values. Function points are computed by comparing five information domain characteristics.

Transaction Functions:

- External Inputs [EI]
- External Output [EO]
- External Inquiries [EQ]

Data Functions:

- Internal Logical Files [ILF]
- External Logical Files [EIF]

Number of external inputs (EI) – Each user input that provides distinct application-oriented data to the software is counted inputs should be distinguished from inquiries.

Number of external outputs (EO) – Each user output that provides application-oriented

information to the user is counted.

Number of external inquiries (EQ) – An inquiry defined as an on-line input those results in the generation of some immediate software response in the form of an on-line output. Each distinct inquiry counted. **Number of Internal logical files (ILF)** – Each logical internal file is a logical grouping of data that resides within the application's boundary and is maintained via external inputs.

Numbers of external interfaces files (EIF) – All machine-readable interfaces that used to transmit information to another system counted. The weights of the domains are fixes, which are provided in appropriate table location. Weights can be divided into three categories according to the functionality of the system. They are simple, average and complex. The total system is a complex system but the part of the total system. Once these data have collected, a complexity value is associated with each count.

To find out the FP count the following formula is used,

Value Adjustment Factor (VAF) = $(0.65 + (.01 \times \text{TDI}))$

UFP = UFP (Data Fn) + UFP (Transaction Fn)

Adjusted Function Point Count (AFP) = $\text{UFP} \times \text{VAF}$

Effort for JavaScript = $\text{AFP} \times \text{Productivity}$

Complexity Matrix:

El Table

FTR's	DATA ELEMENTS		
	1-4	5-15	> 15
0-1	Low	Low	Ave
2	Low	Ave	High
3 or more	Ave	High	High

Shared EO and EQ Table

FTR's	DATA ELEMENTS		
	1-5	6-19	> 19
0-1	Low	Low	Ave
2-3	Low	Ave	High
> 3	Ave	High	High

Values for transactions

Rating	VALUES		
	EO	EQ	EI
Low	4	3	3
Average	5	4	4
High	7	6	6

RET's	DATA ELEMENTS		
	1-19	20 - 50	> 50
1	Low	Low	Ave
2-5	Low	Ave	High
> 5	Ave	High	High

Rating	Values	
	ILF	EIF
Low	7	5
Average	10	7
High	15	10

4.4 Identifying Complexity

4.4.1 Identifying Complexity of transition function

sl	Transaction Functions	TF	Fields/File involvement	FTR	DET
1	User need to login using valid email address and pasword	EI	Fields: Email, Password File: password	1	2
2	View daily sale and total sale, total balance, total number of medicine	EO	Fields: Click button File: User_dashboard	1	1
3	View date wise all transaction	EO	Fields: Click button File: Transaction_history	1	1
4	View medicine list	EO	Fields: Click File: medicine_list	1	1
5	Admin can add new medicine category	EI	Fields: Click button File: medicine_category	1	1
6	Admin can add new medicine type	EI	Fields: Click button File: medicine_type	1	1
7	Admin can add medicine unit	EI	Fields: Click button File: medicine_unit	1	1
8	Admin can add Generic name	EI	Fields: Click button File: generic_name	1	1
9	Admin can add manufacture	EI	Fields: Click button File: manufacture	1	1
10	Can can add medicine	EI	Fields: m_name, manufacture, medicine_category, medicine_un	1	10
11	Admin Can add supplier	EI	Fields: supplier_name, phone File: supplier_list	1	2
12	Admin can add Customer	EI	Fields: customer_name, address, phone File: customer_list	1	3
13	Admin can add account	EI	Fields: bank_name, account_name, account_number, account_t	1	6
14	Admin can purchase medicine	EI	Fields: supplier, manufacture, medicne, date, expire_date, quan	1	10
15	Admin can Sell medicine	EI	Fields: customer, medicine, date, payment_type, quantity, sell_price, to	1	9
16	Admin can add note	EI	Fields: title, description File : note	1	2
17	View All report	EO	Filed: purchase_details, sell_details File: Details	1	2
18					
19					
20					

4.4.2 Identifying Complexity of Data function

SL	Data Functions	DF	Fields/File involvement	RET	DET
1	Accounts	ILF	Fields: account_name, account_number, account_type, bank_name, branch, opening_balance, uid	1	7
2	Customer	ILF	Fields: customer_name, customer_contact, customer_address, total_due, uid	1	5
3	Generic Name	ILF	Fields: generic_name	1	1
4	Manufacture Name	ILF	Fields: manufacture_name	1	1
5	Medicine Category	ILF	Fields: category, uid	1	2
6	Medicine List	ILF	Fields: box_pattern, m_category, m_name, m_type, m_unit, manufacture_price, s_generic_name, s_manufacture, sell_p	1	11
7	Medicine Type	ILF	Fields: medicie_type	1	1
8	Medicine Unit	ILF	Fields: medicine_unit, uid	1	2
9	Note Details	ILF	Fields: title, description, uid	1	3
10	Purchase Medicine	ILF	Fields: account_balance, batch_id, buy_date, expire_date, manufacture_price, payment_type, purchase_Uid, paid, due, q	1	14
11	Sell Medicine	ILF	Fields: customer_name, due_amount, medicine_name, paid_amount, payment_type, sell_date, sell_quantity, stock_quan	1	10
12	Stock medicine	ILF	Fields: manufacture, medicine_name, manufacture_price, medicine_unit, sell_price, stock_quantity, uid	1	7
13	Sipplier list	ILF	Fields: supplier_name, supplier_contact, supplier_uid, total_due	1	4
14	Transaction	ILF	Fields: amount, bank, date, name, reason, uid	1	6
15					
16					
17					
18					
19					
20					

4.4.3 Unadjusted function for transition function

Sl	Transaction Functions	TF	FTR	DET	Complexity	UFP
1	User need to login using valid email address and pasword	EI	1	2	LOW	4
2	View daily sale and total sale, total balance, total number of medicine	E0	1	1	LOW	3
3	View date wise all transaction	E0	1	1	LOW	4
4	View medicine list	E0	1	1	LOW	4
5	Admin can add new medicine category	EI	1	1	LOW	3
6	Admin can add new medicine type	EI	1	1	LOW	3
7	Admin can add medicine unit	EI	1	1	LOW	3
8	Admin can add Generic name	EI	1	1	LOW	4
9	Admin can add manufacture	EI	1	1	LOW	3
10	Can can add medicine	EI	1	10	AVE	7
11	Admin Can add supplier	EI	1	2	LOW	4
12	Admin can add Customer	EI	1	3	LOW	3
13	Admin can add account	EI	1	6	LOW	5
14	Admin can purchase medicine	EI	1	10	AVE	7
15	Admin can Sell medicine	EI	1	9	AVE	6
16	Admin can add note	EI	1	2	LOW	2
17	View All report	E0	1	2	LOW	3
18	0	0	0	0		
19	0	0	0	0		
20	0	0	0	0		

4.4.4 Unadjusted function for data function

SL	DataFunctions	DF	RET	DET	Complexity	UFP
1	Accounts	ILF	1	7	LOW	7
2	Customer	ILF	1	5	LOW	7
3	Generic Name	ILF	1	1	LOW	7
4	Manufacture Name	ILF	1	1	LOW	7
5	Medicine Category	ILF	1	2	LOW	7
6	Medicine List	ILF	1	11	LOW	8
7	Medicine Type	ILF	1	1	LOW	7
8	Medicine Unit	ILF	1	2	LOW	7
9	Note Details	ILF	1	3	LOW	7
10	Purchase Medicine	ILF	1	14	LOW	9
11	Sell Medicine	ILF	1	10	LOW	7
12	Stock medicine	ILF	1	7	LOW	7
13	Supplier list	ILF	1	4	LOW	7
14	Transaction	ILF	1	6	LOW	7
15	0	0	0	0		
16	0	0	0	0		
17	0	0	0	0		
18	0	0	0	0		
19	0	0	0	0		
20	0	0	0	0		

4.4.5 Performance and Environmental impact

General System Characteristic	Brief Description	DI
1 Data communications	How many communication facilities are there to aid in the transfer or exchange of information with the application or system?	5
2 Distributed data processing	How are distributed data and processing functions handled?	2
3 Performance	Was response time or throughput required by the user?	3
4 Heavily used configuration	How heavily used is the current hardware platform where the application will be executed?	5
5 Transaction rate	How frequently are transactions executed daily, weekly, monthly, etc.?	
6 On-Line data entry	What percentage of the information is entered On-Line?	5
7 End-user efficiency	Was the application designed for end-user efficiency?	2
8 On-Line update	How many ILP's are updated by On-Line transaction?	0
9 Complex processing	Does the application have extensive logical or mathematical processing?	5
10 Reusability	Was the application developed to meet one or many user's needs?	0
11 Installation ease	How difficult is conversion and installation?	0
12 Operational ease	How effective and/or automated are start-up, back-up, and recovery procedures?	5
13 Multiple sites	Was the application specifically designed, developed, and supported to be installed at multiple sites for multiple organizations?	0
14 Facilitate change	Was the application specifically designed, developed, and supported to facilitate change?	2
Total Degree of Influence (TDI)		34
Value Adjustment Factor (VAF) = $65 + (0.01 \times \text{TDI})$		
		0.99

4.4.6 Calculating Function Point

UFP =	UFP (TF) + UFP (DF)
UFP(Total) =	169
Adjusted Function Point Estimation (AFP) =	UFP (Total) * TDI
Adjusted Function Point Estimation (AFP) =	167.31

Effort for Java =	AFP * Productivity for Java						
Effort for Java =	167.31	17					
Effort for Java =	2844.27		Person Hours				
Effort for Java =	355.53375		Person Days (8 Hours Per Day)				
Effort for Java =	15.45798913		Person Months (23 Days Per Month)				
Effort for Java =	3.864497283	4	Months (1 person for developing the project)				
		Number of team members					
		1 person need 3.86 months of completing the work					

4.6 Cost Estimation

Cost analysis represents the total cost to complete any project. In this project, there are four factors to analyze and calculate the cost. The factors are personnel cost, software cost, hardware cost and other cost.

- Personnel cost: Personnel cost is the salary of the customer communicator, system analyst and designer, coder and tester. For estimating the cost the analyzer used the minimum industrial average.
- Software cost: It is the cost of the software which is used in this project
- Hardware cost: cost of the computer that used to complete the project.
- Other cost: Other cost includes the cost of the house rent, telephone bill, electricity bill, convenience and so on.

4.6.1 Personal Cost

- Number of days in a year = 365
- Number of government holidays in a year = 20
- Number of weekly holidays in a year = 52 [Friday only]
- Total number of working days to develop the project = $365 - (52 + 20) = 293$ days
- Total number of working days per months to develop the project = $293/12 = 24.417$ days
- Organization working hours per day = 8 hours
- Organization working hours per month = $24.417 * 8 = 195.33$ hours
- Total working hour in 4 months = $24 * 8 * 4 = 768$ hours

Designation	Person	Working Hours	Salary/ Month	Salary/ Hour	Individual Salary	Total Salary
System Analyst	One	768	9766.5	50	38400	140560 BDT
Coder	One	768	8789.85	45	34560	
Designer	One	768	7813.2	40	30720	
Tester	One	768	6836.55	35	36880	

4.6.2 Hardware cost

The first step is to sum the digits or number starting with the life and going back to one. For example, an asset with a life of 5 would have a sum of digits follows: $5+4+3+2+1 = 15$. To find the percentage for each year divide the year's digit by the sum. In the example above the 49 percentage would be calculated as follows:

Year	Service Percentage
1 st Year	$5/15 = 0.3334 * 100\% = 33.34\%$
2 nd Year	$4/15 = 0.2667 * 100\% = 26.67\%$
3 rd Year	$3/15 = 0.2 * 100\% = 20\%$
4 th Year	$2/15 = 0.1333 * 100\% = 13.33\%$
5 th Year	$1/15 = 0.0667 * 100\% = 6.67\%$

Table 4. 1 Hardware Service Percentage

Hardware Name	Quantity	Purchase Price(PP)	Depreciation Rate(DR)	Depreciation Cost= PP*DR	Depreciation Expense ((PP-SV)/12)*4	Total Cost
Computer	1	40000	33.34%	13336	8888	10776.7 BDT
Printer	1	8500	33.34%	2833.9	1888.7	

4.6.3 Software Cost

Name	Quantity	Price per year	Price for 4 months	Total Cost
Windows 10 Home	1 PC	Free	Free	1600 BDT
Microsoft office	1 user	4800	1600.00	
Android Studio	N/A	Free	Free	
Firebase	N/A	Free	Free	
Emulator	N/A	Free	Free	

4.6.4 Others cost

Purpose	Amount	Total Cost
Transport	2000	10500 BDT
House Rent	2500	
Electric Bill	5500	
Other Minors	1000	

Table 4. 2 Other Cost Estimation

4.6.5 Account Table

Cost Sector	Cost
Personnel Cost	140560
Hardware Cost	10776.7
Software Cost	1600
Other Cost	10500
Total =	163436.7

Chapter 5: Risk Engineering

5.1 Risk Management

A risk is a serious problem that might or might not happen. It is necessary to analyze the potential risks in a project. If the risks of a software project are not properly analyzed and estimated, many problems can plague the software project. Anyone developing any type of system encounter with it and it has to be managed.

5.1.1 Stage of Risk

- **Risk Identification:** Risk identification is the process of detecting potential risks or hazards through data collection. A range of data collection and manipulation tools and techniques exists. The team is using both automated and manual techniques to collect data and begin to characterize potential risks to Web resources. Web crawling is one effective way to collect information about the state of Web pages and sites.
- **Risk Classification:** Risk classification is the process of developing a structured model to categorize risk and fitting observable risk attributes and events into the model. The team combines quantitative and qualitative methods to characterize and classify the risks to Web pages, Web sites, and the hosting servers.
- **Risk Assessment:** Risk assessment is the process of defining relevant risk scenarios or sequences of events that could result in damage or loss and the probability of these events. Rosenthal describe the characteristics of a generic standard for risk assessment as "transparent, coherent, consistent, complete, comprehensive, impartial, uniform, 52 balanced, defensible, sustainable, flexible, and accompanied by suitable and sufficient guidance.
- **Risk Analysis:** Risk analysis determines the potential impact of risk patterns or

scenarios, the possible extent of loss, and the direct and indirect costs of recovery. This step identifies vulnerabilities considers the willingness of the organization to accept risk given potential consequences, and develops mitigation responses.

- **Risk Management Implementation:** Risk management implementation defines policies, procedures, and mechanism to manage and respond to identifiable risks. The implemented program should balance the value of assets and the direct and indirect costs of preventing or recovering from damage or loss.

5.1.2 Types of Risk

There are different categories of risks that should be considered in any software project. The following categories of risks have been considered in this software project.

- **Project risks:** These risks threaten the project plan. If these risks become real, it is likely that the project schedule will slip and that costs will increase. Project risks identify potential budgetary, schedule, personnel, resource, customer and requirement problems and their impact on the software project.
- **Technical risks:** These risks threaten the quality and timeliness of the software to be produced. If a technical risk becomes a reality, implementation may become difficult or impossible. Technical risks identify potential design, implementation, interface, verification and maintenance problems. Moreover, specification ambiguity, technical uncertainty, technical obsolescence are also risk factors.
- **Business risks:** These risks threaten the viability of the software to be built. The business

risks can be market risks, building a system that no one really wants. Strategic risks, building a system that no longer fits into the overall business strategy for the company. Management risks, losing the support of senior management due to a change in focus or a change in people. Budget risks, losing budgetary or personnel commitment.

5.2 The RMMM plan

- **Risk Mitigation:** Proactive planning for risk avoidance.
- **Risk Monitoring:** Assessing whether predicted risks occur or not, ensuring preventive steps are being properly applied, collect information for future risk analysis, attempt to determine which risks caused which problem.
- **Risk Management:** Actions to be taken in the event that mitigation steps have failed and the risk has become a live problem.
- **Type of Impact:**
 1. Catastrophic
 2. Marginal
 3. Tolerable
 4. Critical
- **Type of Probability:**
 1. Very low (<10%),
 2. Low (10–25%),
 3. Moderate (25–50%),
 4. High (50–75%),
 5. Very high (>75%).

- **Project Risks:** Threaten the project plan. In my system, the bellow mentioned projects risks I needed manage.

Project risk 1	
Name	Change in requirement
Probability	Low
Impact	Marginal
Description	Customer may change their requirement
Mitigation and Monitoring	Requirements are redefined by the company due to time or business needs. Meeting will be held with the company regularly. This ensures that the product we are producing solves a problem.
Management	Emergency meeting between both parties to identify new project requirement and goals.
Status	Not Occur

Project risk 2	
Name	Poor Quality of documentation
Impact	Catastrophic
Description	Documentation quality may poor
Mitigation and Monitoring	Routine Meeting will be held to offer documentation suggestions and topics. The progress on documentation will also have a monitor in each meeting.
Management	Meeting will call and discuss about quality improvement of documentation. New things will assigned to responsible person.
Status	Monitoring it

Project risk 3	
Name	Computer crash
Impact	Catastrophic
Description	Computer may crash due to several reason
Mitigation and Monitoring	We should take proper follow up of computers. Need to use them carefully.
Management	If computer has been crashed then we will restore backup.
Status	Managed it.

Chapter 6: Analysis Modeling

Analysis Model could be a technical illustration of the system. It acts as a link between system description and style model. In Analysis Modelling, data, behavior, and functions of the system area unit outlined and translated into the design, component, and interface level style within the style modeling.

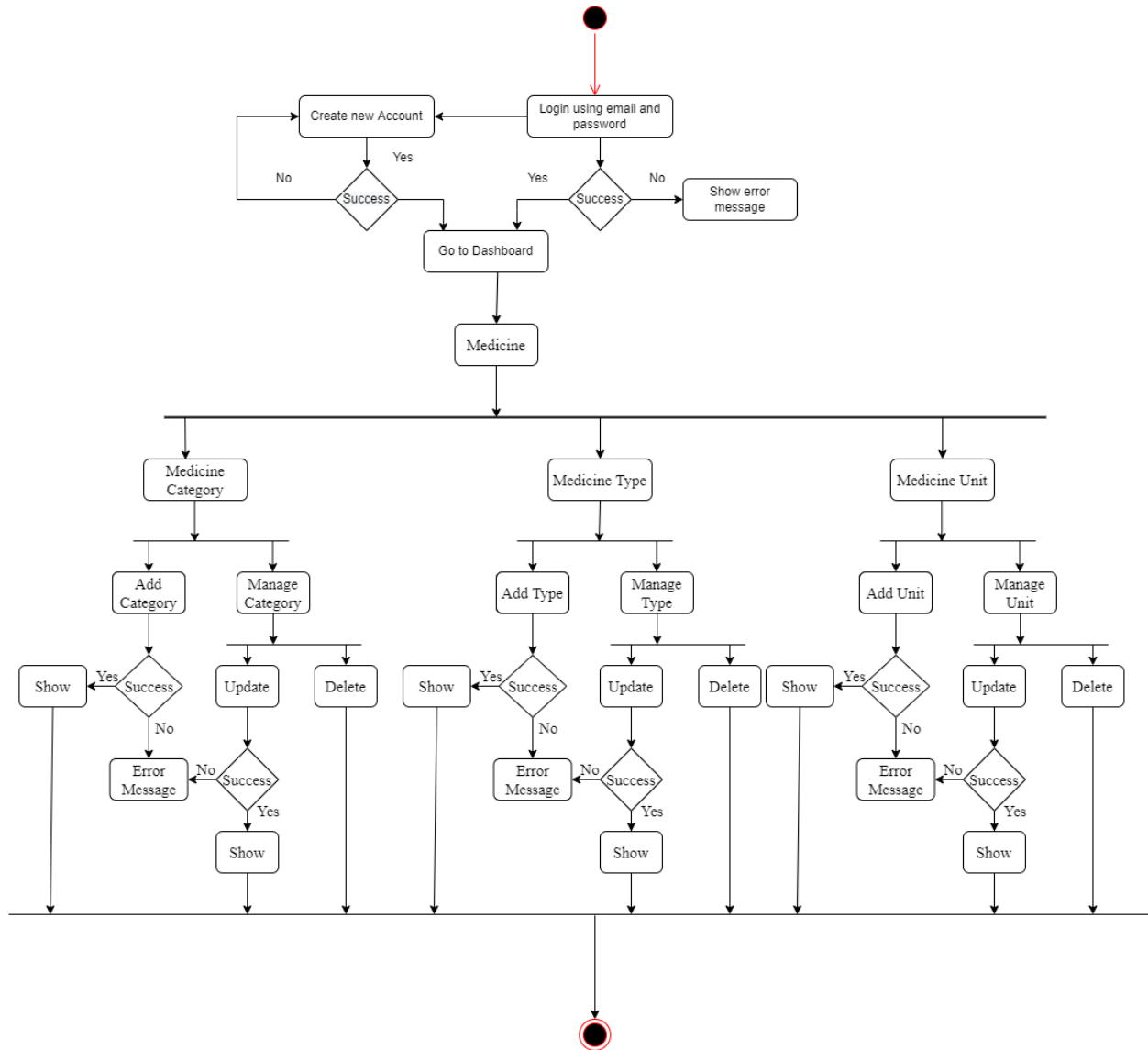
6.1 Software Analysis Pattern

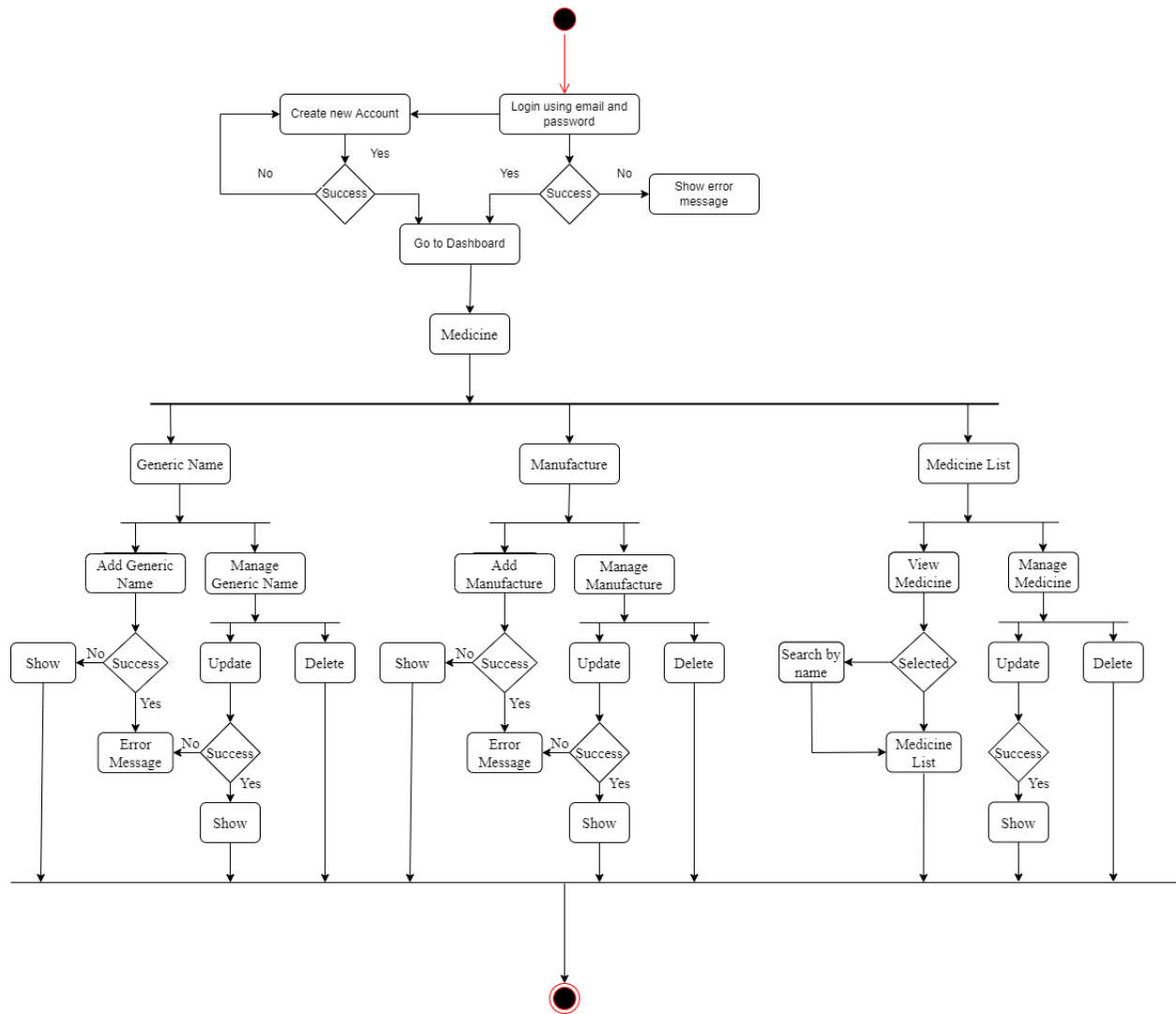
Software analysis patterns or analysis patterns in software engineering are conceptual models, which capture an abstraction of a situation that can often be encountered in modeling. An analysis pattern can be represented as a group of related, generic objects (meta-classes) with stereotypical attributes (data definitions), behaviors (method signatures), and expected interactions defined in a domain-neutral manner. By analysis modeling developer can finalize the specifications of the system. It is mandatory –

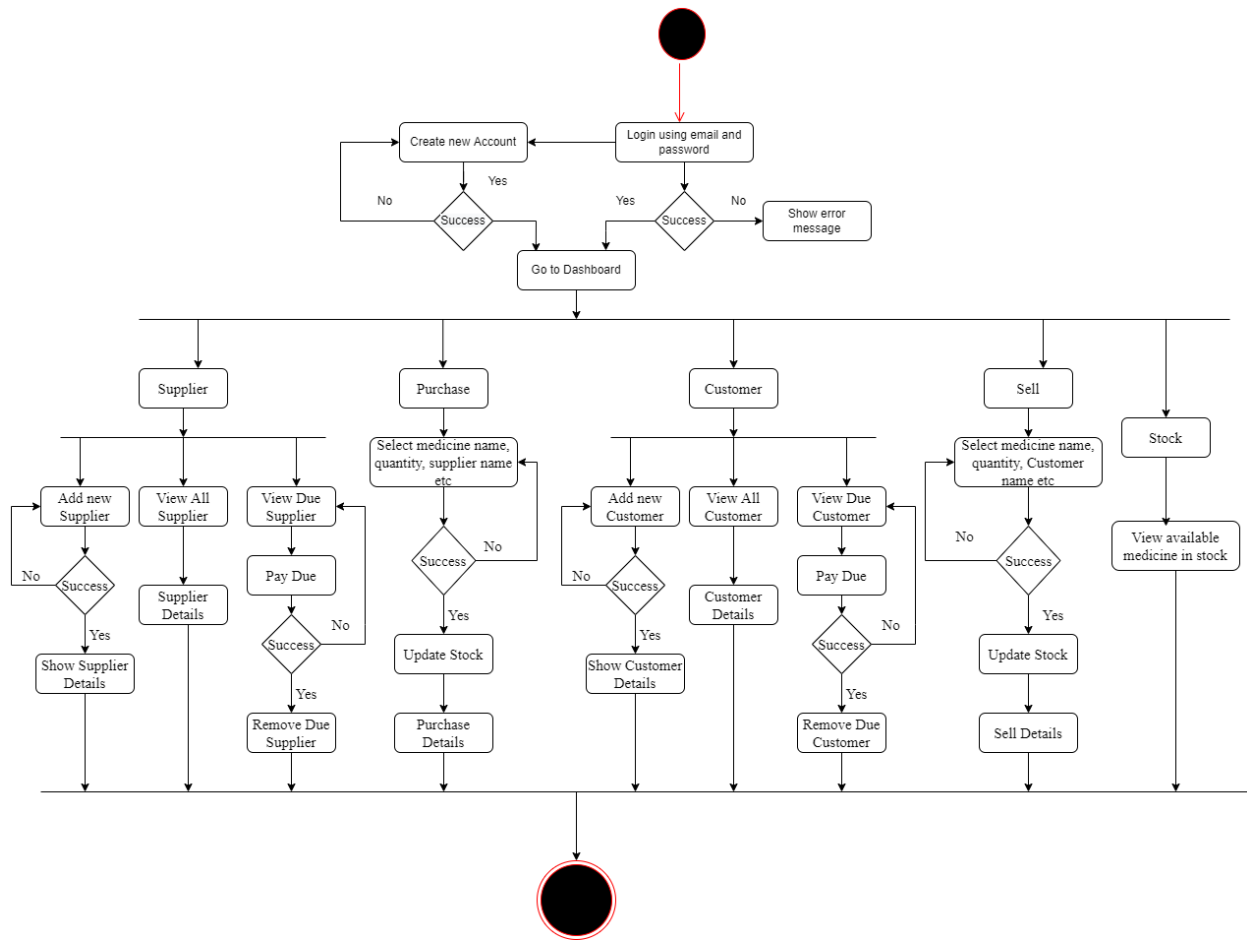
- To describe what the customer require
- To establish a basis for the creation of a software design
- To define a set of requirements that can be validated once the software is built.

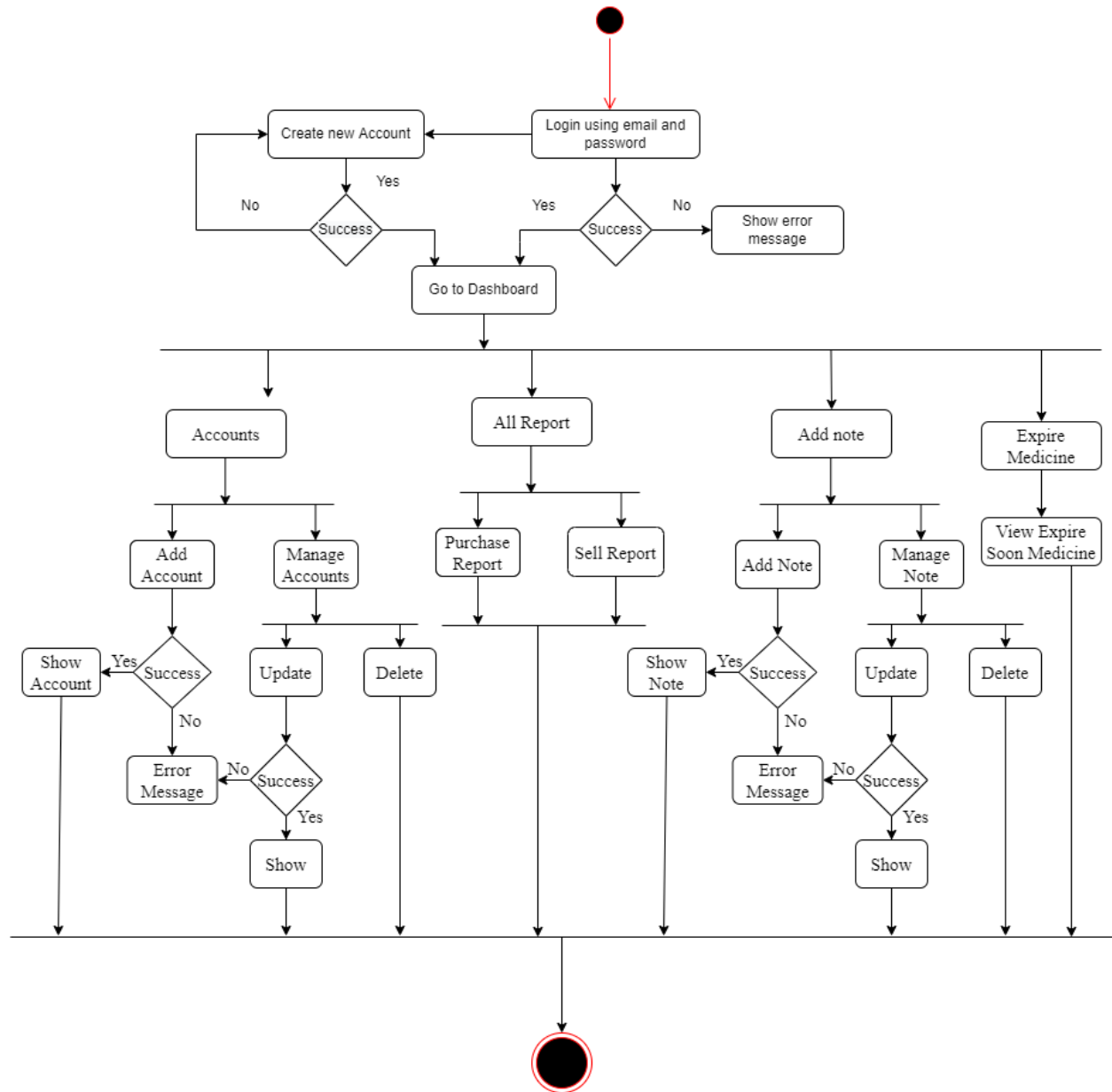
6.2 Activity Diagram

Activity diagram is an important behavioral diagram in UML diagram to describe dynamic aspects of the system. Activity diagram is essentially an advanced version of flow chart that modeling the flow from one activity to another activity. The diagrams describe the state of activities by showing the sequence of activities performed.









Chapter 7: Designing

7.1 Data flow diagram

A data flow diagram is a graphical representation that depicts information flow and the transforms that are applied as data move from input to output. It is known as data flow graph or bubble chart. The DFD may be used to represent a system or software at any level of abstraction. DFD may be partitioned into levels that represent increasing information flow and functional detail. Therefore, the DFD provides a mechanism for functional modeling as well as information flow modeling. A level 0 DFD, which is also known as fundamental system model or a Context model, represents the entire software or system element into as a single bubble with input and output data indicated by in Coming and outgoing arrows respectively. Then bubble of context model should be decomposed into several levels.

In DFD, there are four symbols:

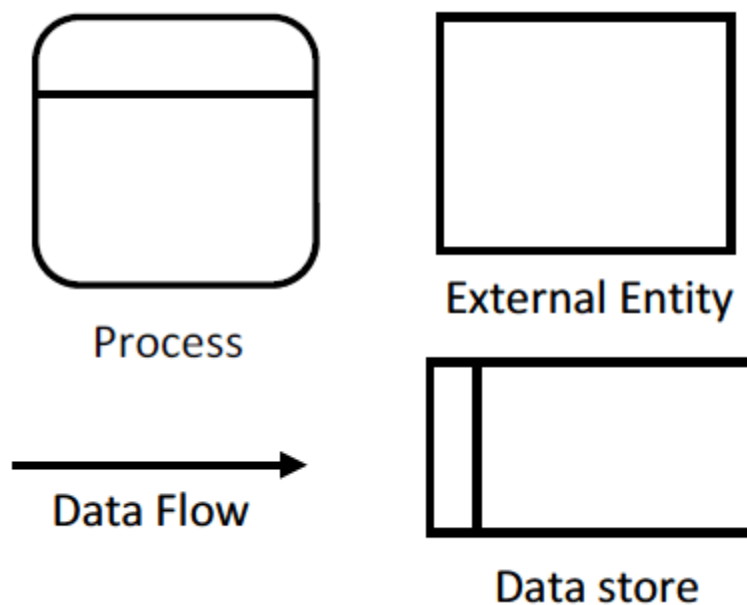


Figure 7.1: DFD symbol

- A circle or a bubble represents a **process** that transforms incoming data flow(s) into outgoing data flow(s).
- This **square box** defines a source or destination that is **external entity** of system data.
- An arrow identifies **data flow** that data is motion. It is pipeline through which information flows.
- An **open rectangle** is a **data store** or a temporary repository of data.

7.1.1 Context Level Diagram

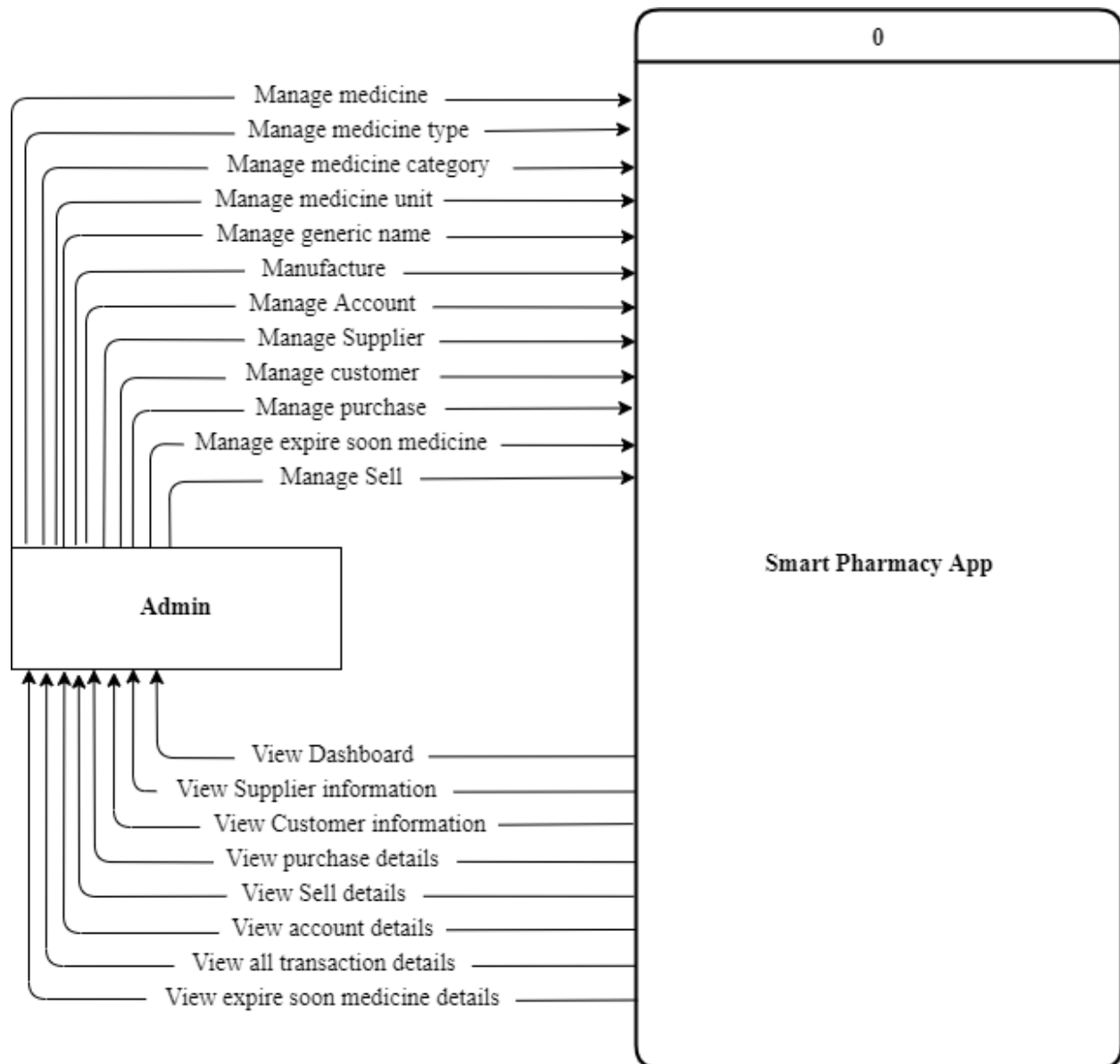


Figure: 7.2: Context Level DFD

7.1.2 Level 1 Data Flow Diagram

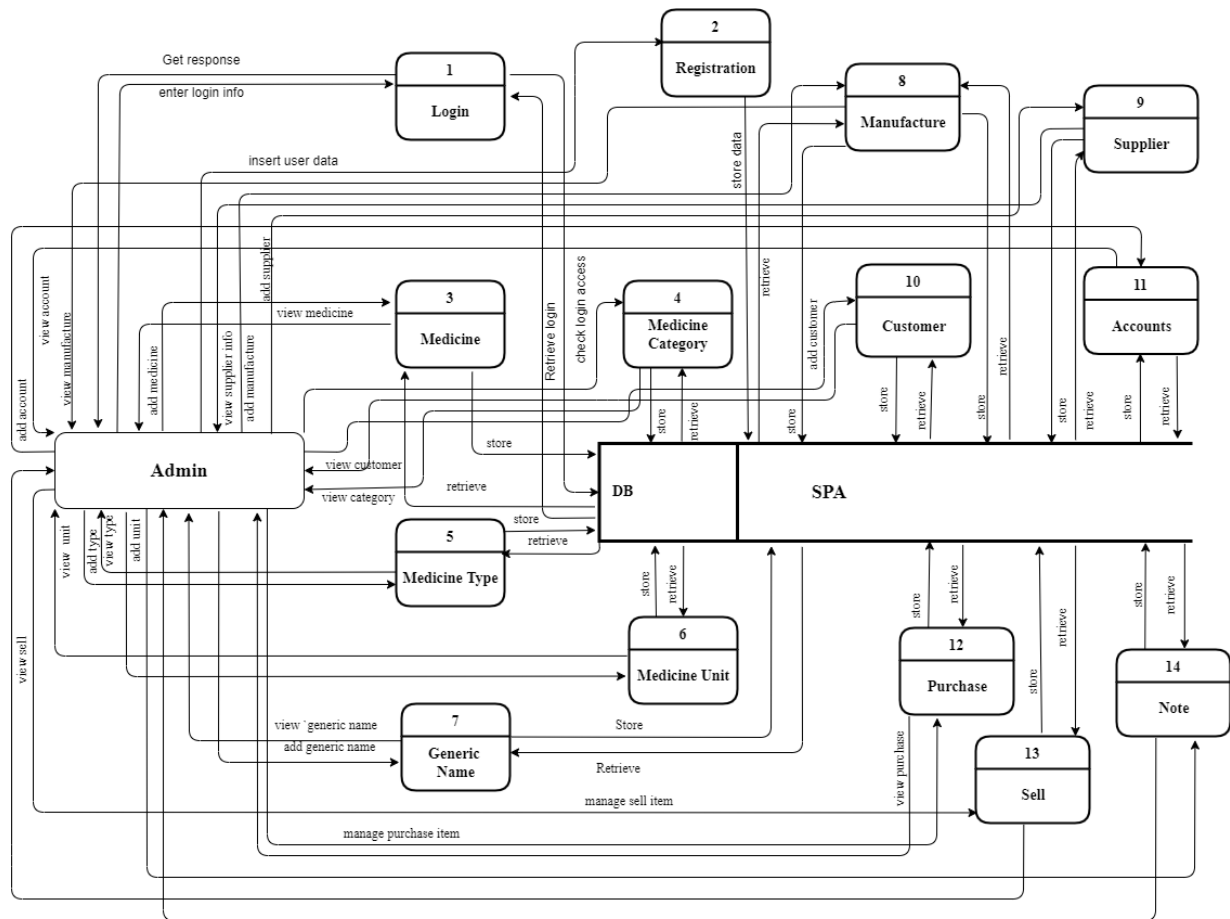


Figure: 7.3: Level 1 DFD

7.1.3 Level 2 DFD of Process 1(Login)

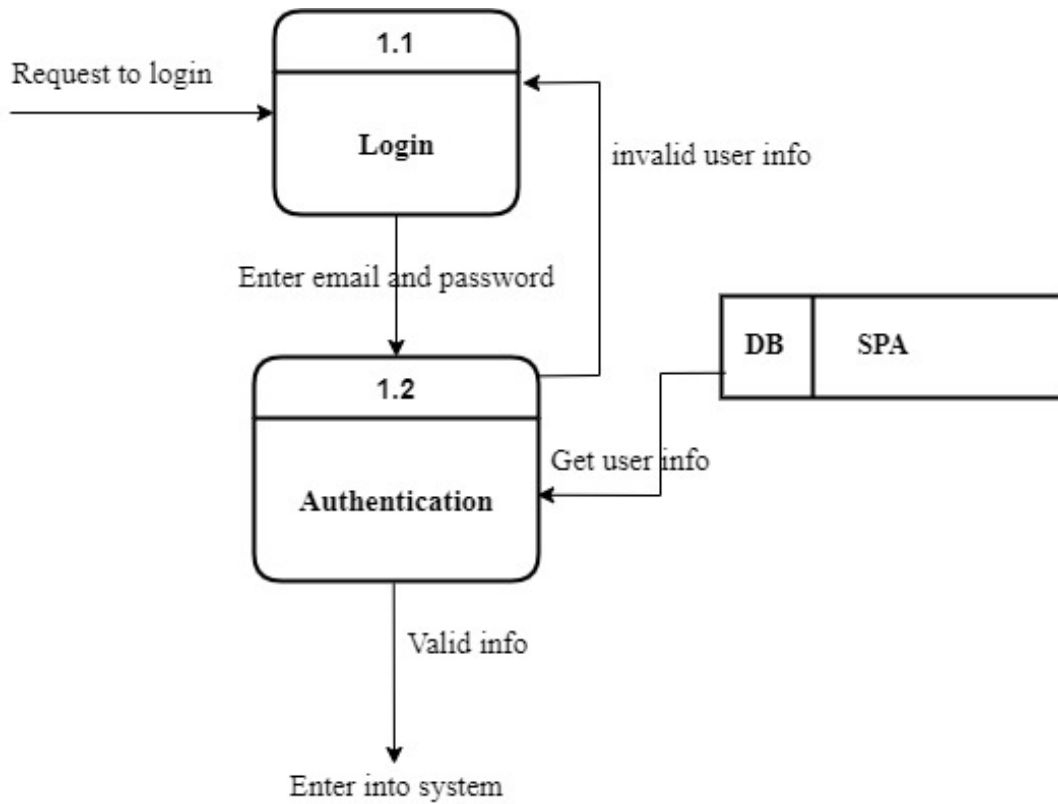


Figure 7.4: Level 2 DFD process 1

7.1.4 Level 2 DFD of Process 2(Register)

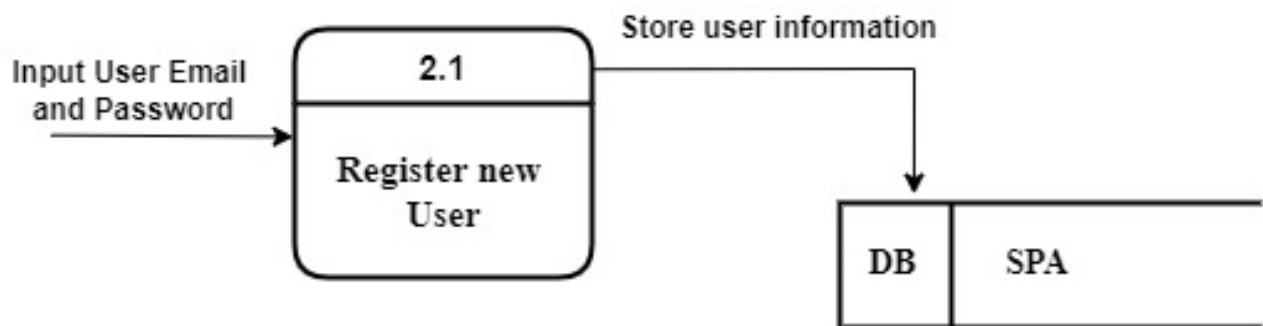


Figure 7.5: Level 2 DFD process 2

7.1.5 Level 2 DFD of Process 3

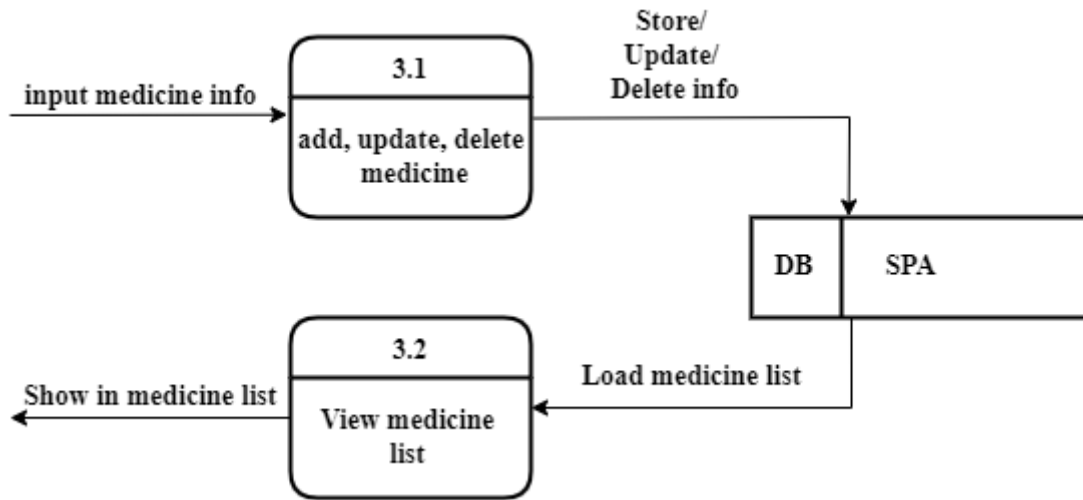


Figure 7.6: Level 2 DFD process 3

7.1.6 Level 2 DFD of Process 4

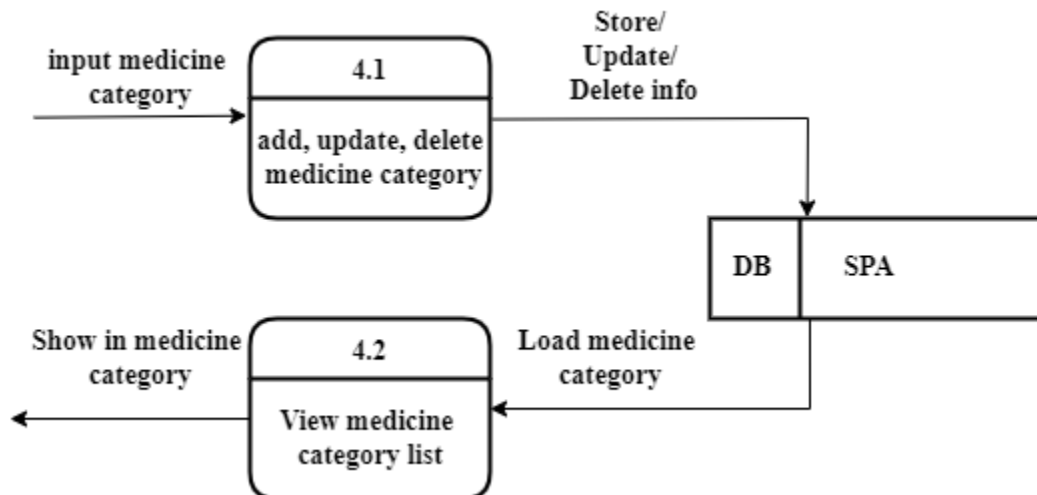


Figure 7.7: Level 2 DFD process 4

7.1.7 Level 2 DFD of Process 5

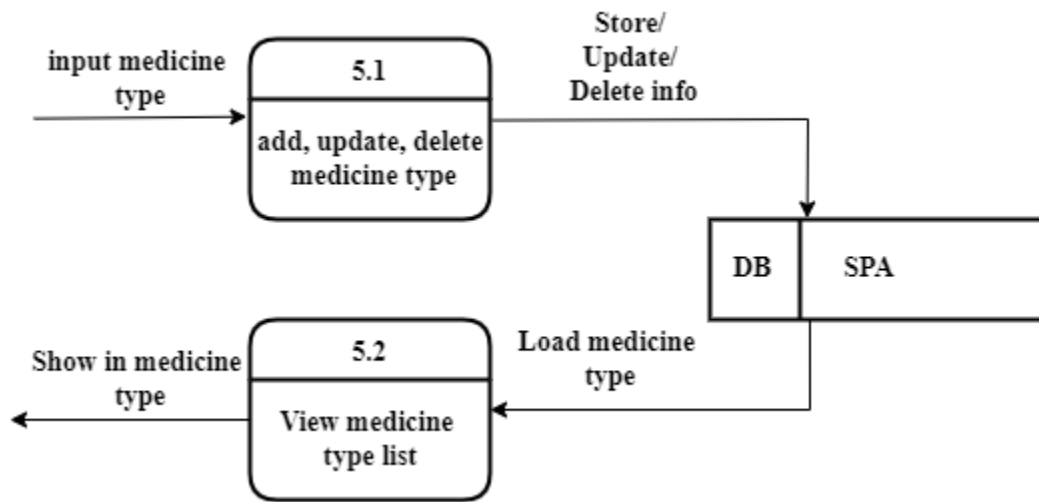


Figure 7.8: Level 2 DFD process 5

7.1.8 Level 2 DFD of Process 6

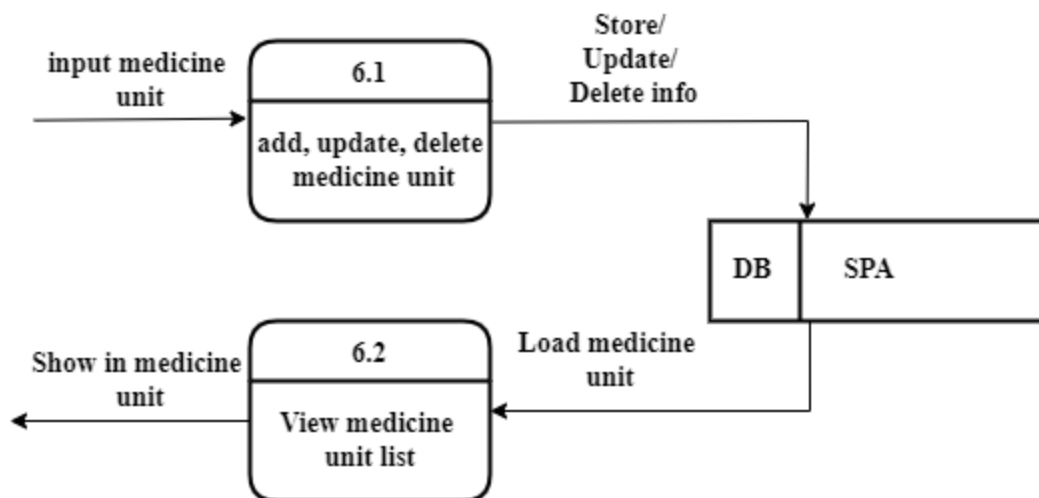


Figure 7.9: Level 2 DFD process 6

7.1.9 Level 2 DFD of Process 7

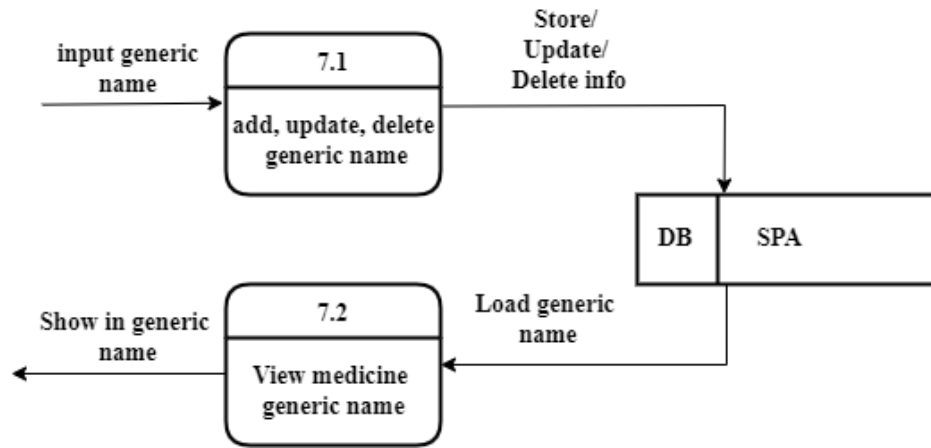


Figure 7.10: Level 2 DFD process 7

7.1.10 Level 2 DFD of Process 8

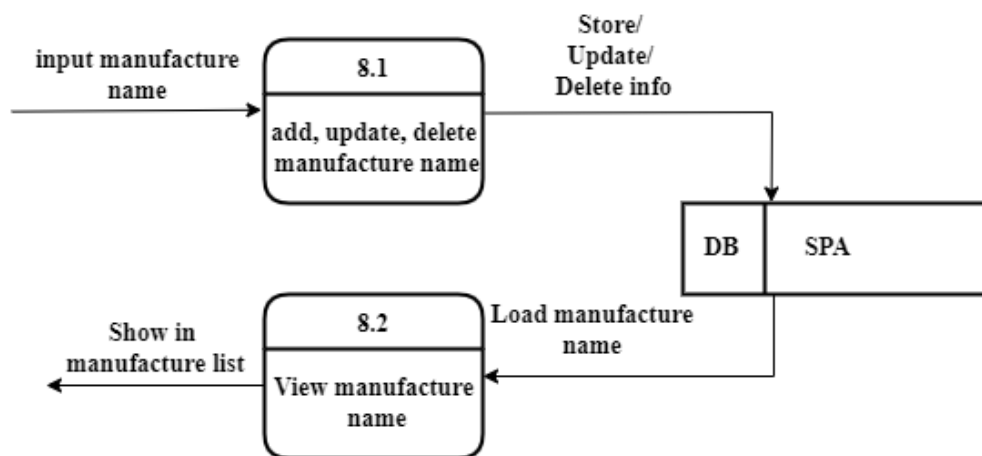


Figure 7.11: Level 2 DFD process 8

7.1.11 Level 2 DFD of Process 9

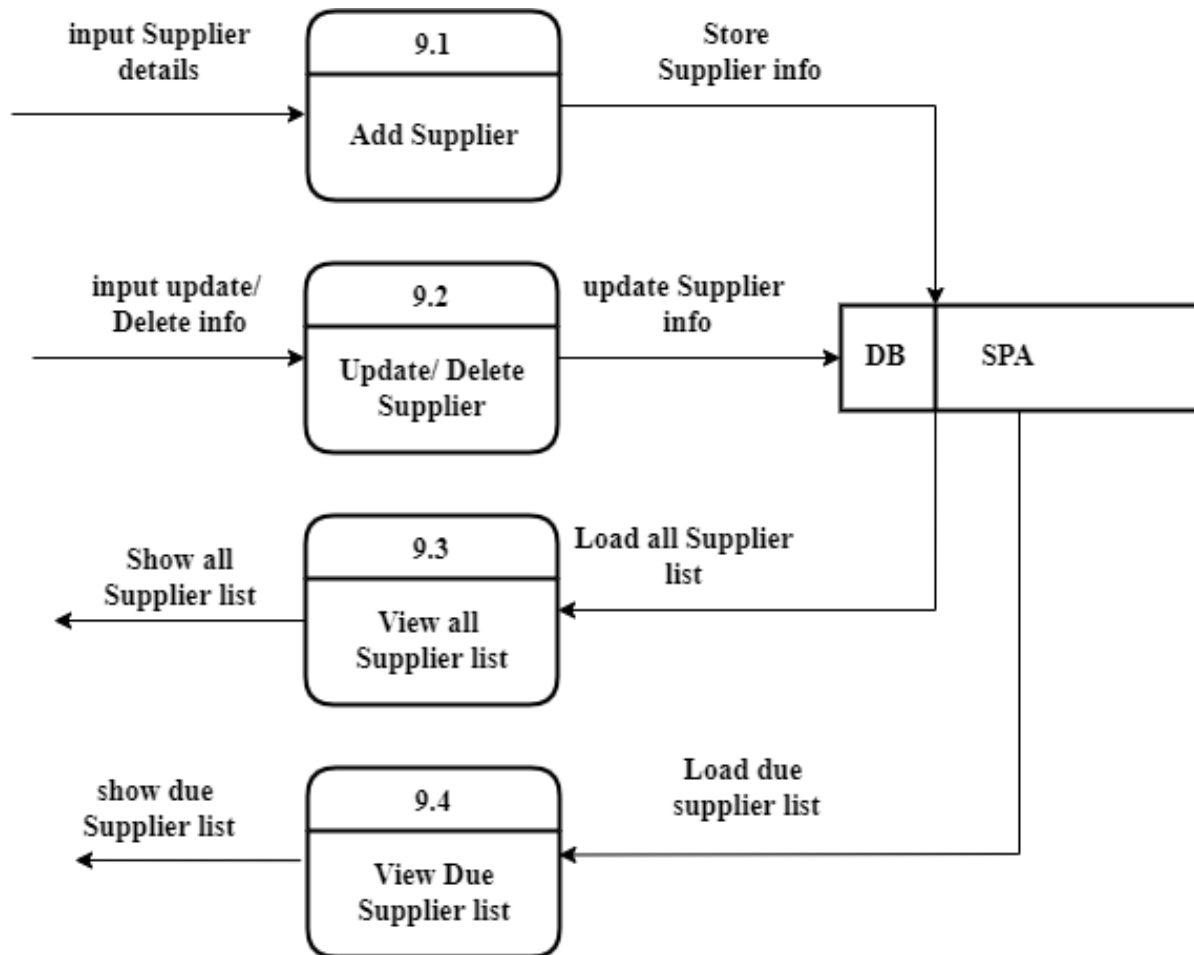


Figure 7.12: Level 2 DFD process 9

7.1.12 Level 2 DFD of Process 10

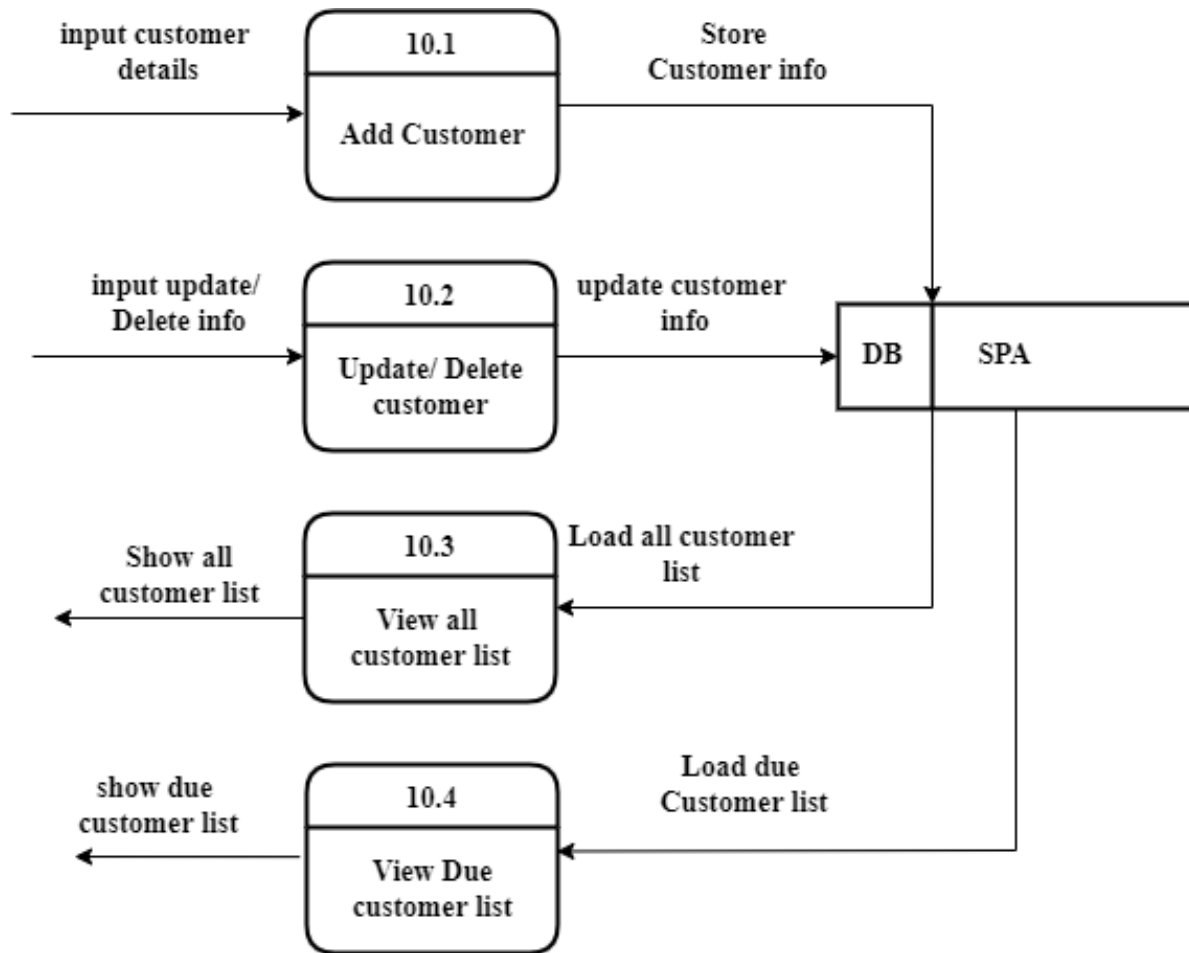


Figure 7.13: Level 2 DFD process 10

7.1.13 Level 2 DFD of Process 11

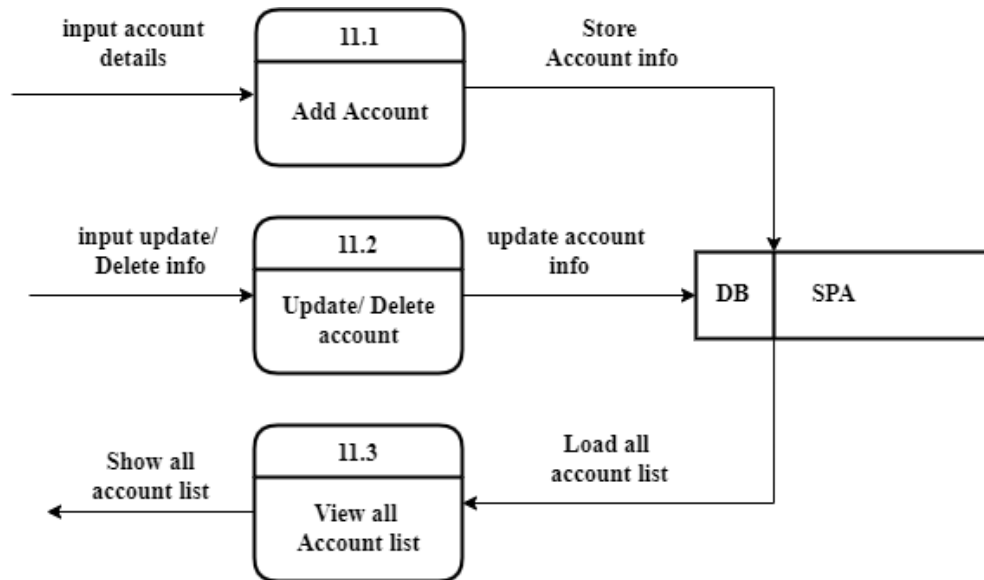


Figure 7.14: Level 2 DFD process 11

7.1.15 Level 2 DFD of Process 12

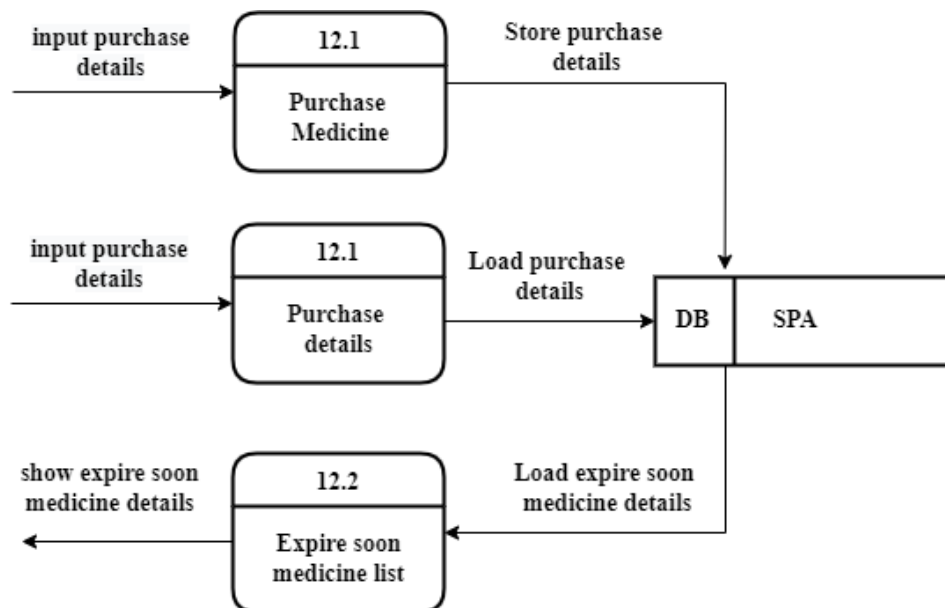


Figure 7.15: Level 2 DFD process 12

7.1.16 Level 2 DFD of Process 13

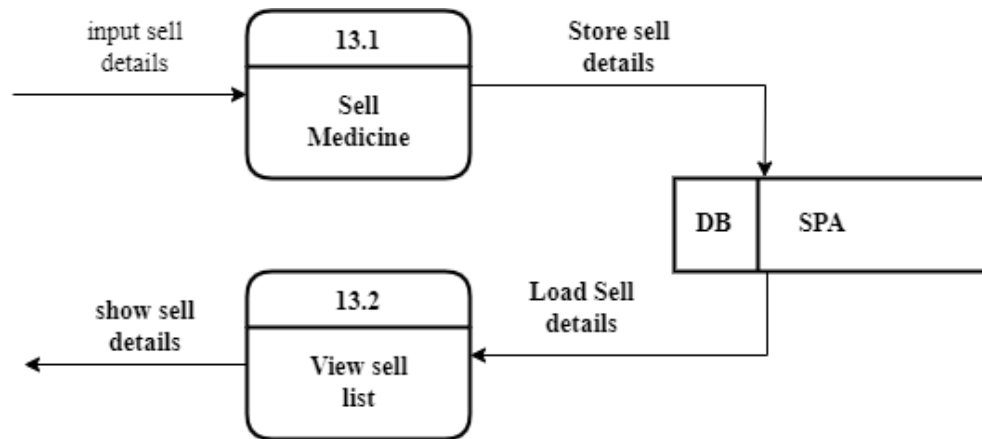


Figure 7.16: Level 2 DFD process 13

7.1.17 Level 2 DFD of Process 14

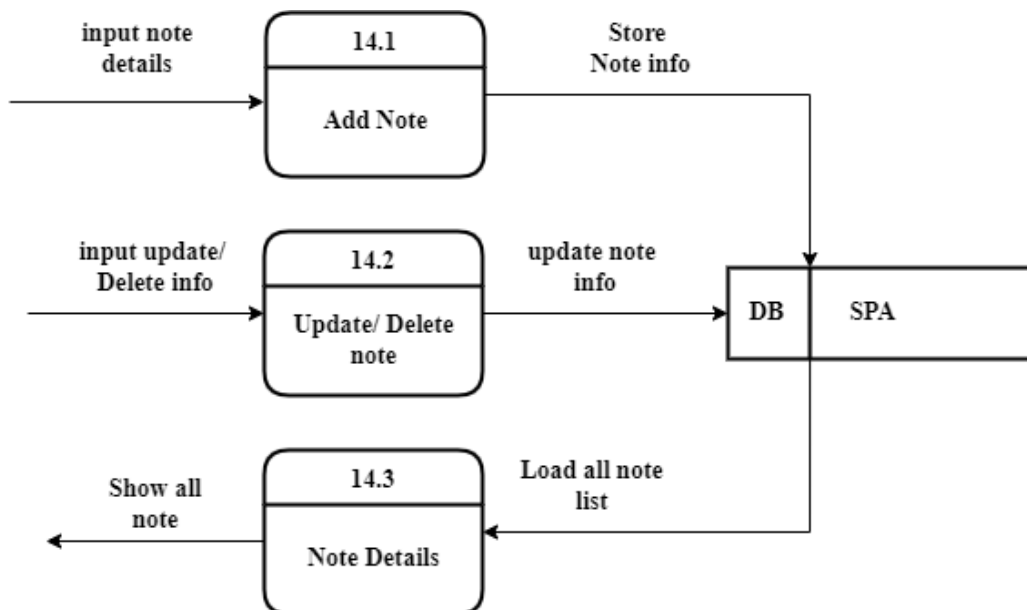


Figure 7.17: Level 2 DFD process 14

7.2 Database Design

7.2.1 Entity Relationship Diagram

An entity relationship diagram (ERD) shows the relationships of entity sets stored in a 76 database.

An entity in this context is a component of data. In other words, ER diagrams illustrate the logical structure of databases. An Entity Relationship (ER) diagram is a type of flowchart that illustrates how, “entities” such as people, objects or concepts relate to each other within a system. ER diagrams are most often used to design or debug relational databases in the fields of software engineering, business, information systems, education and research. Also known as ERDs or ER models, they use a defined set of symbols such as rectangles, diamonds, ovals and connecting lines to depict the interconnectedness of entities, relationships and their attribute. An ER diagram is a means of visualizing how the information a system produces is related.

Entity Relationship Diagram of Smart Pharmacy App

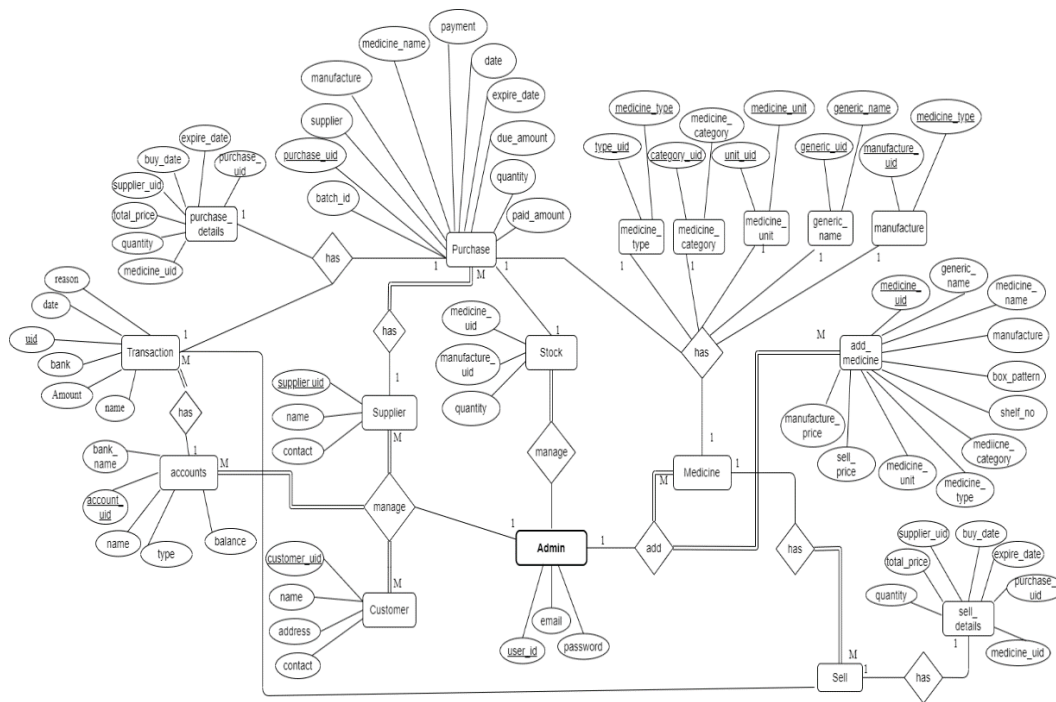


Figure 7.18: Entity Relationship Diagram

7.2.2 Database Design

Database Structure

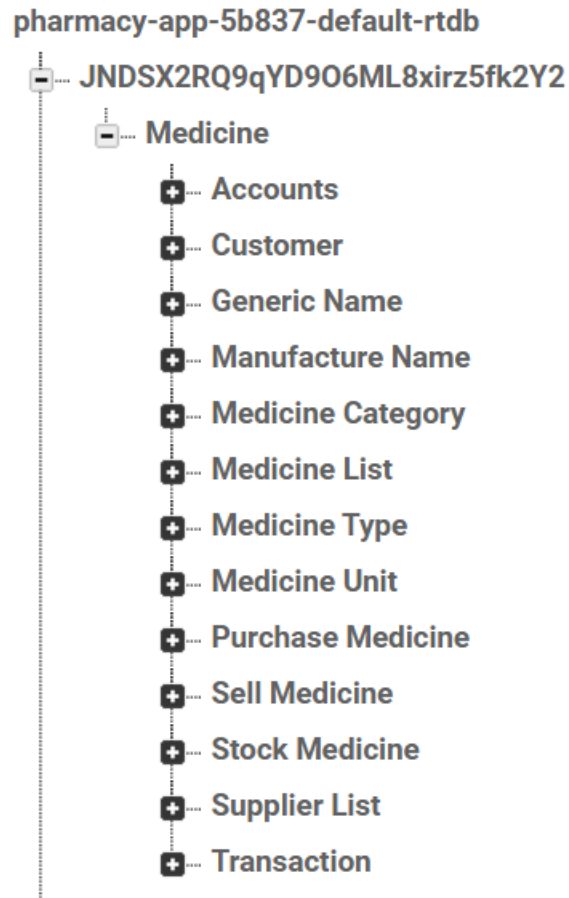


Figure 7.19: Database Structure

Account Table Structure:

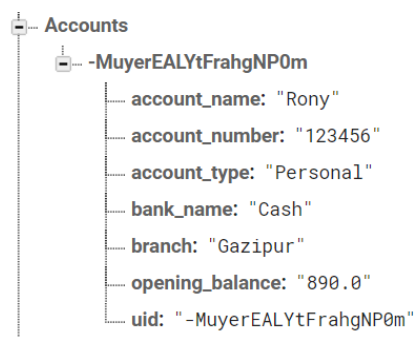


Figure: 7.20: Accounts Table structure

Customer Table Structure:

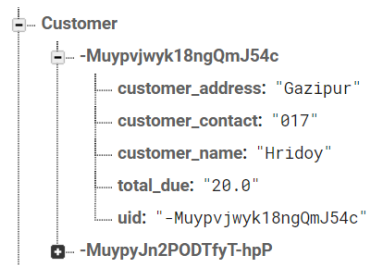


Figure: 7.21: Customer Table structure

Generic Name Table Structure:

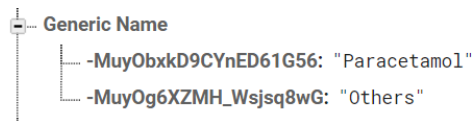


Figure: 7.22: Generic Name Table structure

Manufacture Table Structure:

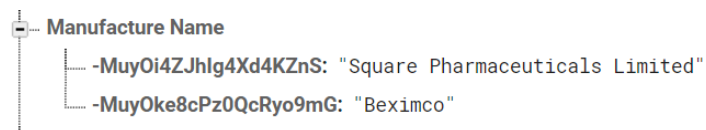


Figure: 7.23: Manufacture Table structure

Medicine Category Table Structure:

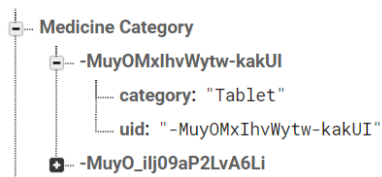


Figure: 7.24: Medicine Category Table structure

Medicine List Table Structure:

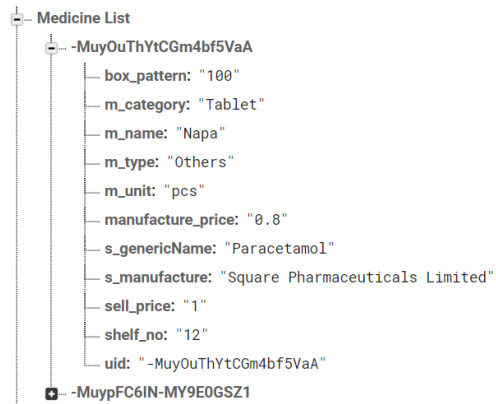


Figure: 7.25: Medicine List Table structure

Medicine Type Table Structure:

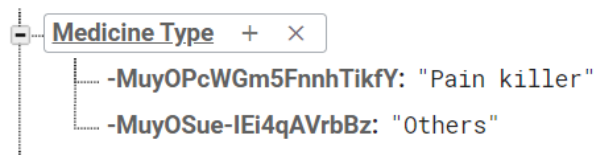


Figure: 7.26: Medicine Type Table structure

Medicine Unit Table Structure:

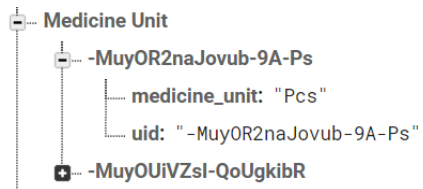


Figure: 7.27: Medicine Unit Table structure

Purchase Medicine Table Structure:

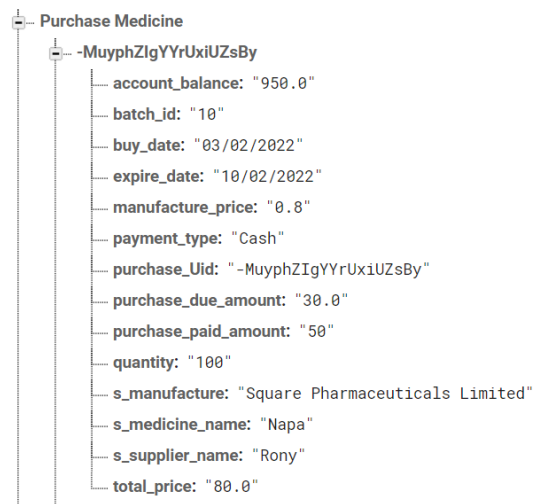


Figure: 7.28: Purchase Medicine Table structure

Sell Medicine Table Structure

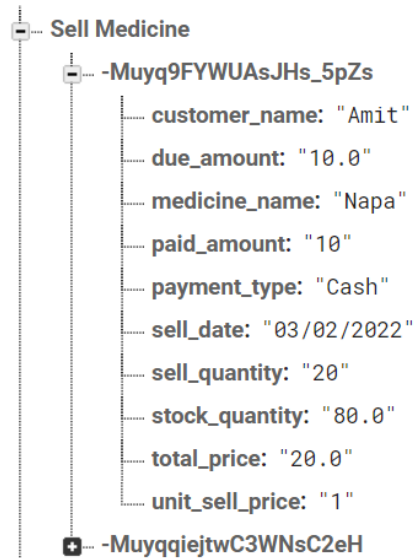


Figure: 7.29: Sell Medicine Table structure

Stock Medicine Details Table:

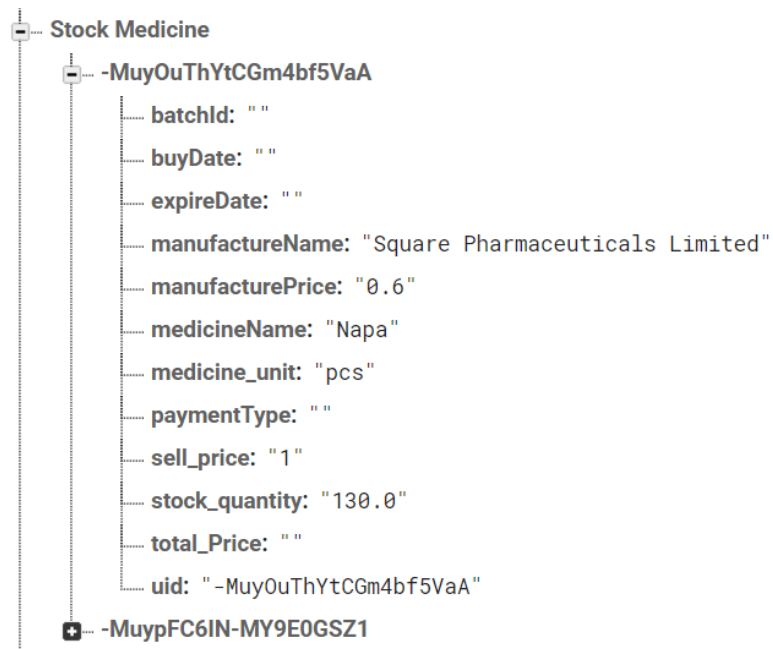


Figure: 7.30: Stock Medicine Table structure

Supplier List Table Structure:

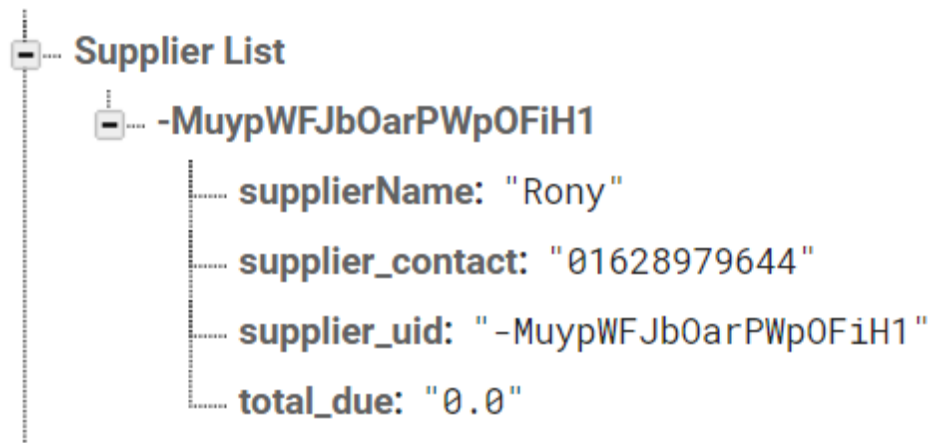


Figure: 7.31: Supplier Table structure

Transaction Table Structure:

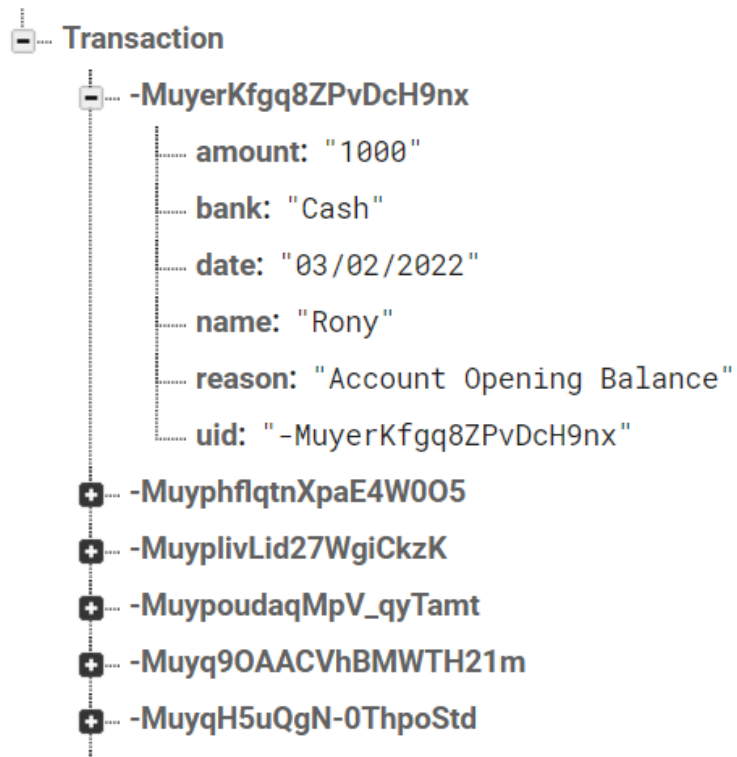
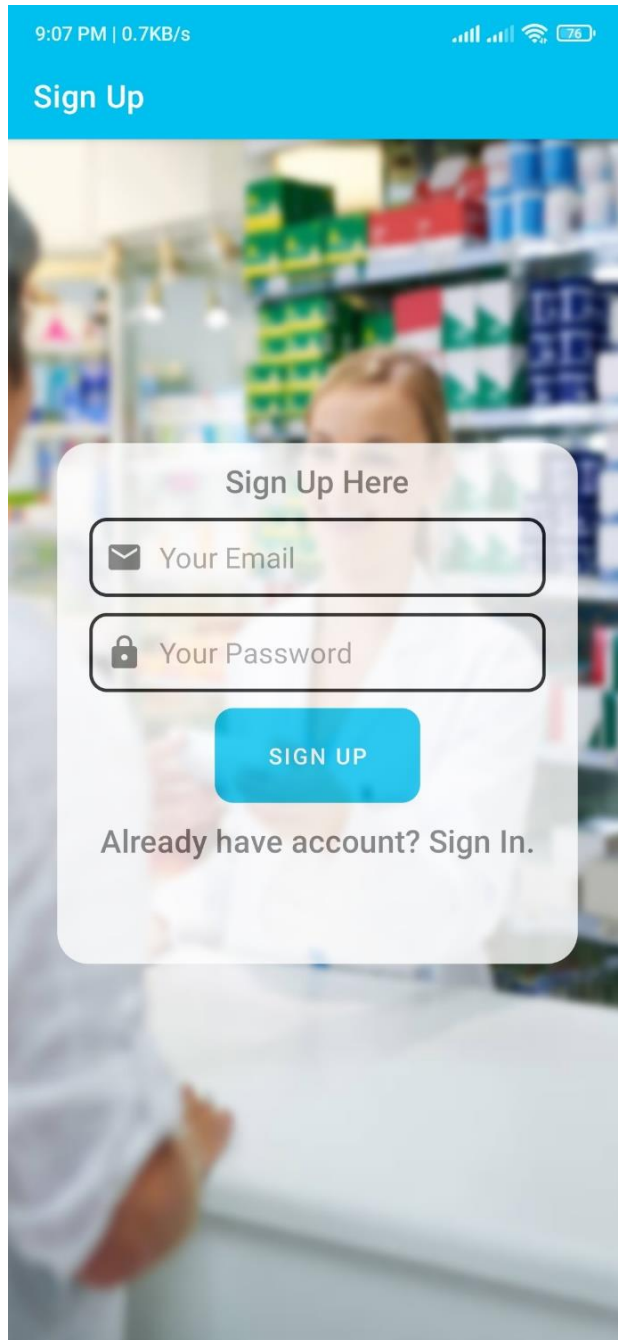


Figure: 7.32: Transaction Table structure

7.3 Interface Design

A mobile application interface for a sign-up page. The screen has a blue header bar with the text "Sign Up" in white. Below the header is a blurred background image of a person in a pharmacy. Overlaid on this is a white rounded rectangle containing the sign-up form. The form has a title "Sign Up Here", two input fields labeled "Your Email" and "Your Password" with envelope and lock icons respectively, a blue "SIGN UP" button, and a link "Already have account? Sign In." at the bottom.

9:07 PM | 0.7KB/s

Sign Up

Sign Up Here

✉ Your Email

🔒 Your Password

SIGN UP

Already have account? Sign In.

Figure: 7.33: Sign up page

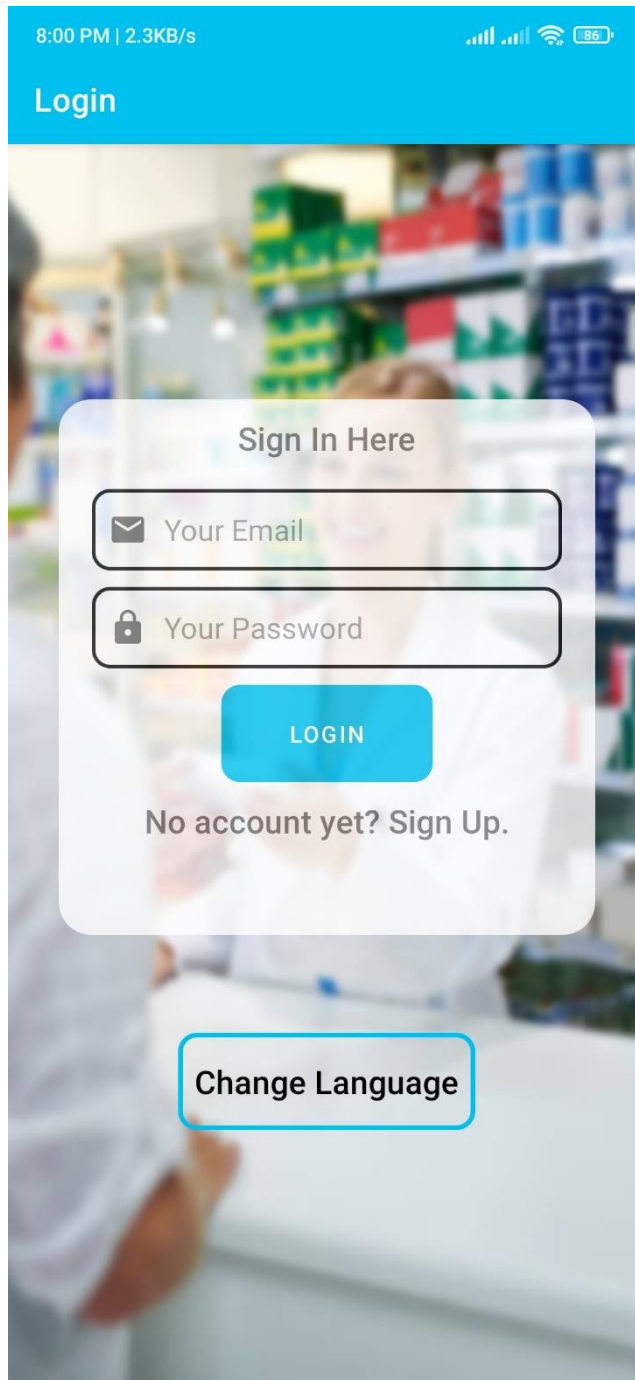


Figure: 7.34: Login page

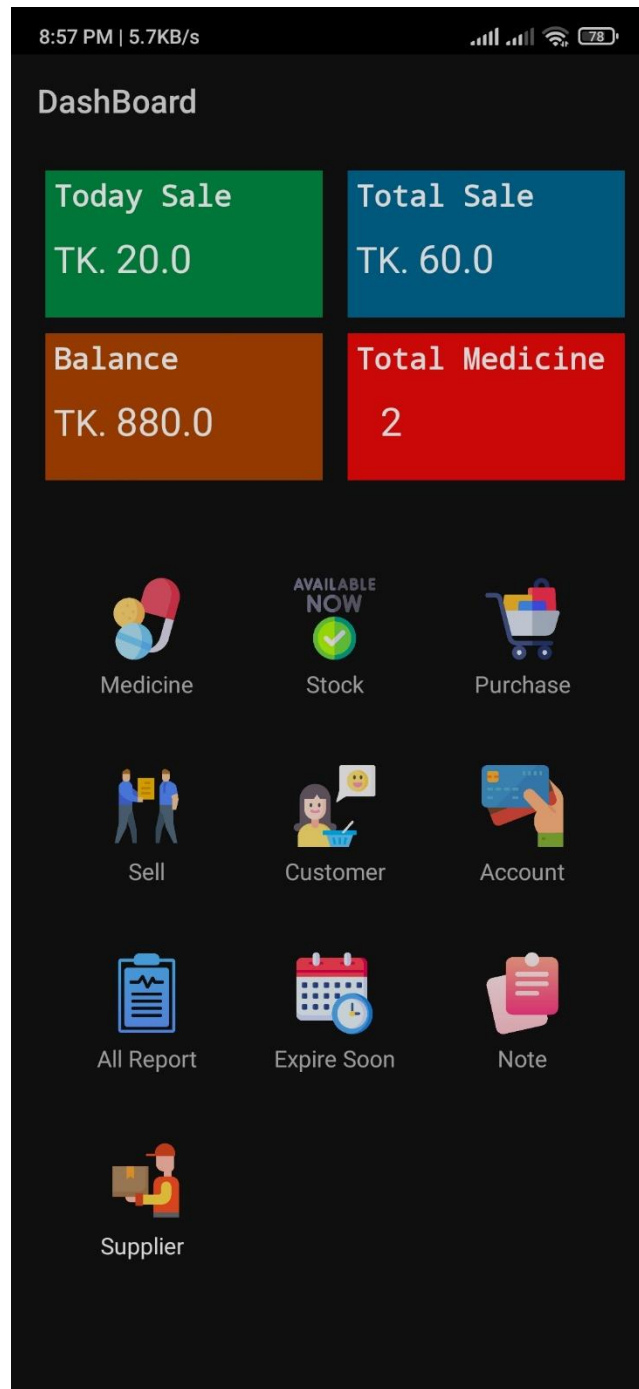


Figure: 7.35: Dashboard Design

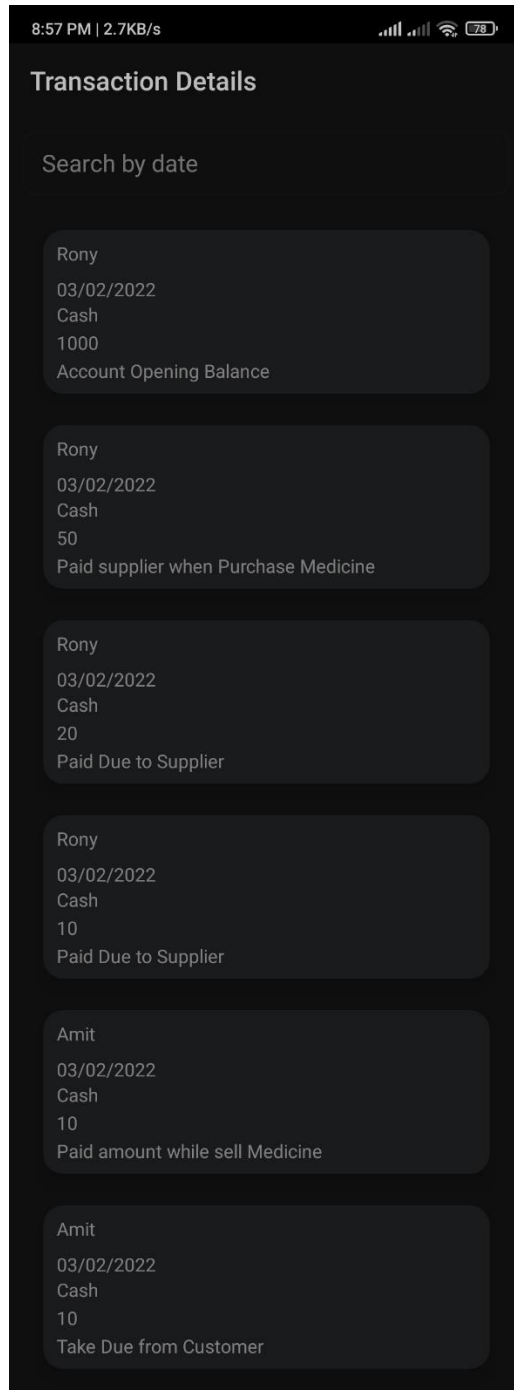


Figure: 7.36: Transaction Details

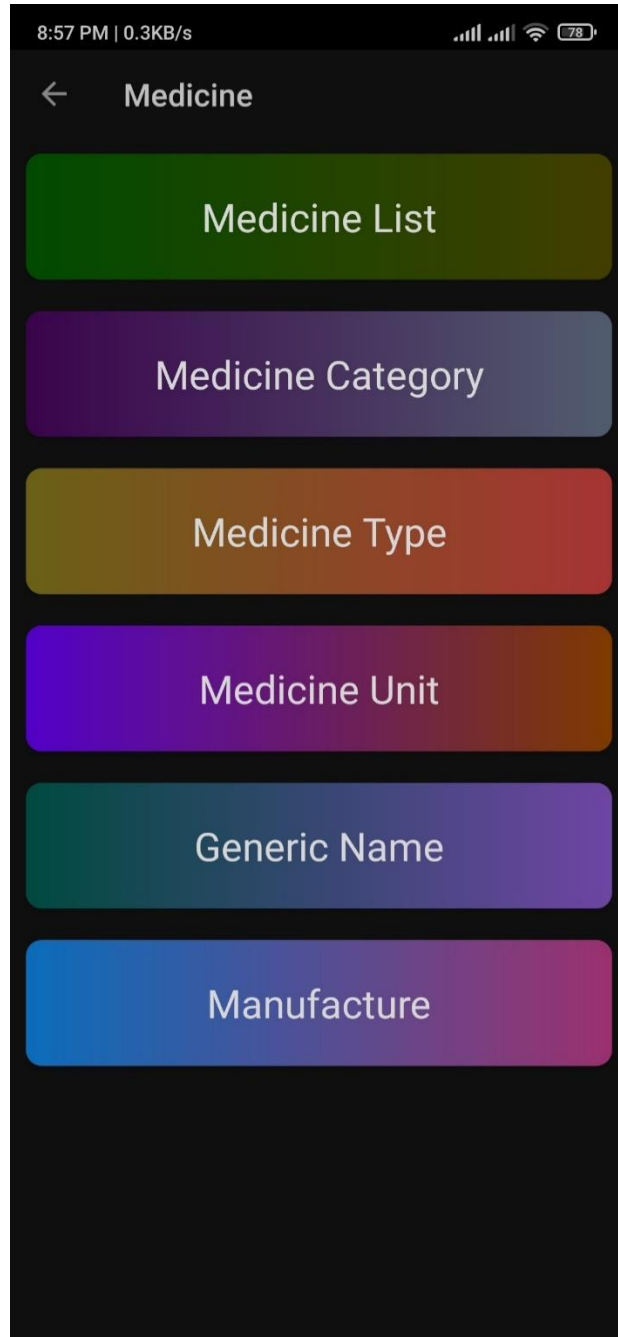


Figure: 7.37: Medicine Dashboard

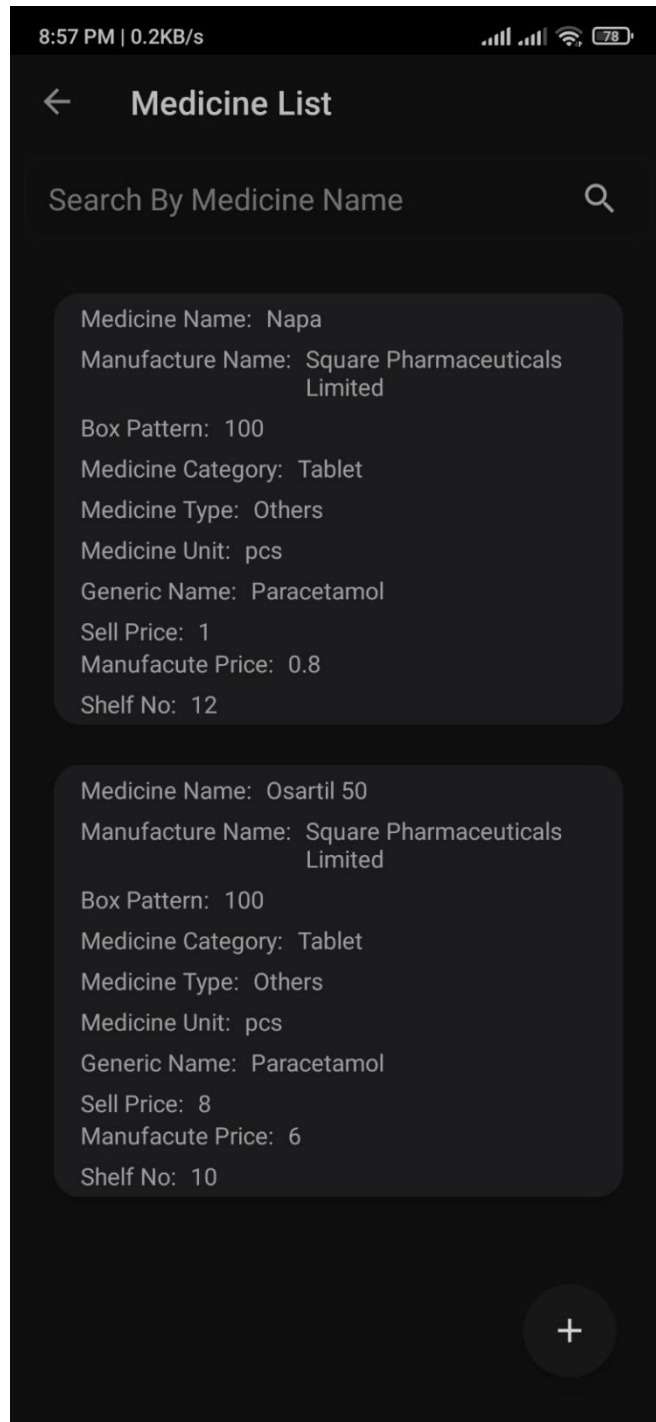


Figure: 7.38: Medicine List

8:57 PM | 0.6KB/s

← Medicine List

Search By Medicine Name

Update Medicine

Osartil 50

Square Pharmaceuticals Limited

100

Tablet

Others

pcs

Paracetamol

8

6

10

Update

+

Figure: 7.39: Update Medicine List

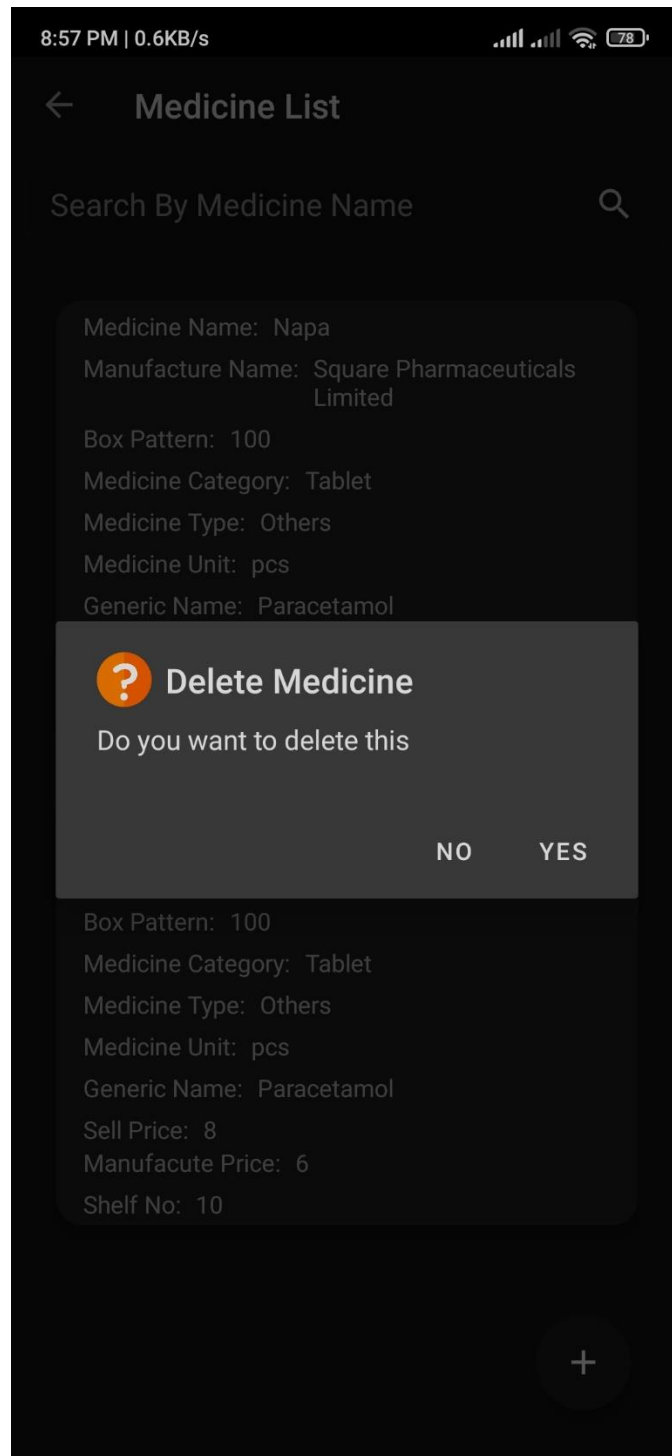


Figure: 7.40: Delete Medicine List

8:57 PM | 0.2KB/s 📶 🔋 78

← **Add Medicine**

Medicine Name

Manufacture Square Pharm.. ▾

Box pattern

Medicine category Tablet ▾

Medicine Type Pain killer ▾

Medicine Unit

Generic Name Paracetamol ▾

Sell price

Manufacture price

Shelf Number

Add Medicine

Figure: 7.41: Add Medicine to the List

8:58 PM | 0.3KB/s

← Medicine Category

Medicine Category (e.g. Tablet, Syrup..)

Save

Figure: 7.42: Add Medicine Category

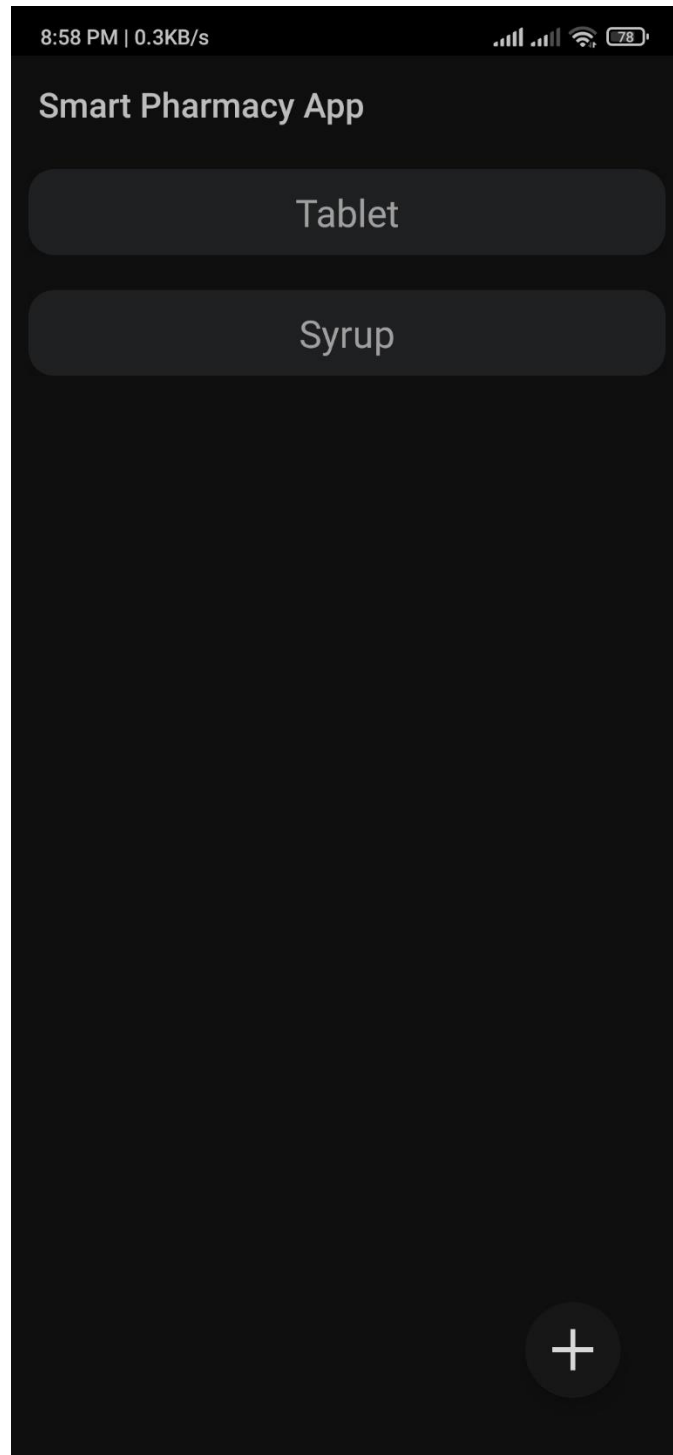


Figure: 7.43: View Medicine Category

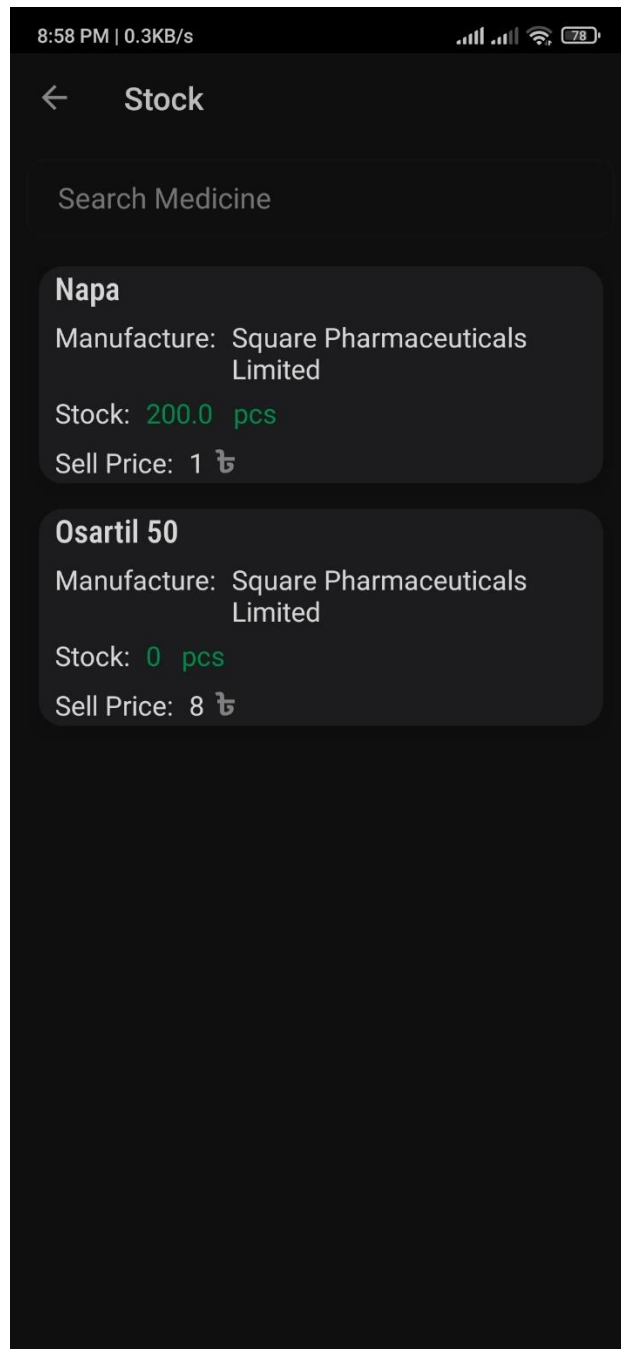


Figure: 7.44: Stock Medicine

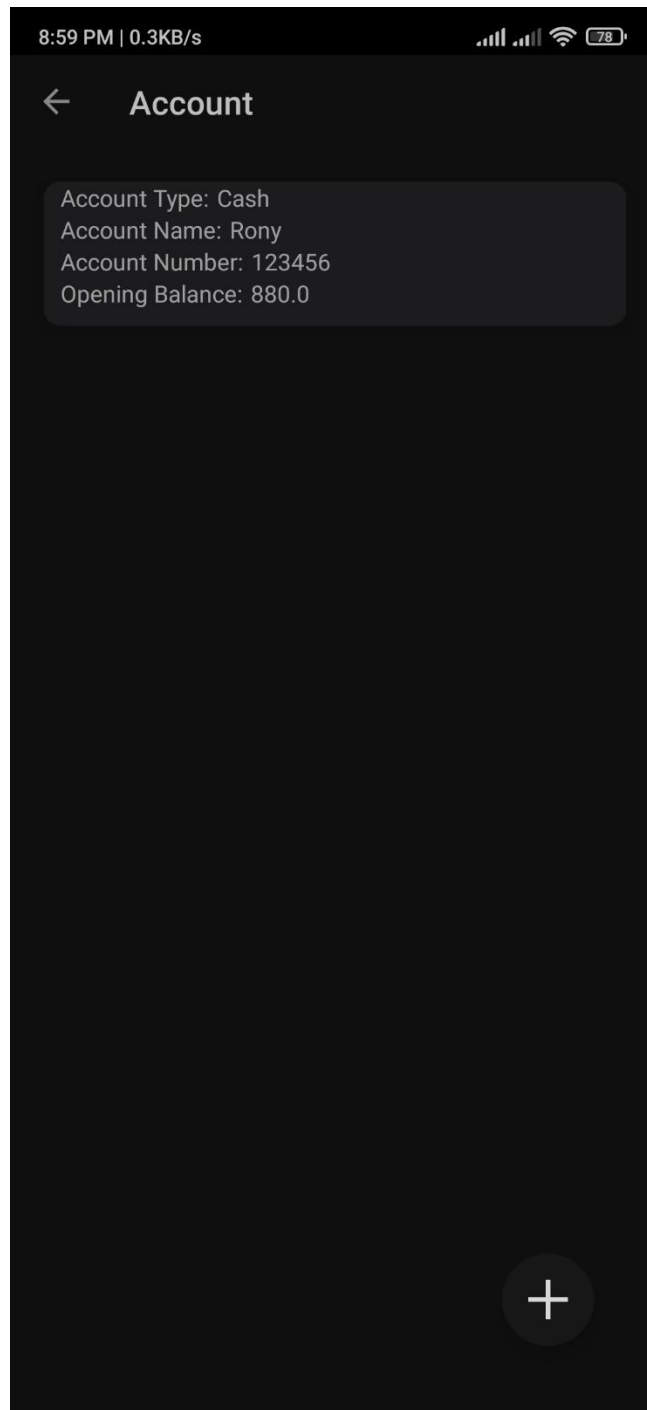


Figure: 7.45: Accounts

8:58 PM | 0.2KB/s

← Purchase

Supplier

Rony

Manufacture

Square Pharmace..

Medicine Name

Napa

Date

Payment Type

Cash

Account Balance:

880.0

Batch Id

Expire Date

Purchase Quantity

Stock quantity

200.0

Manufacture Price

0.8

Total Price

Paid Amount

Due Amount

PURCHASE

Figure: 7.46: Purchase Medicine

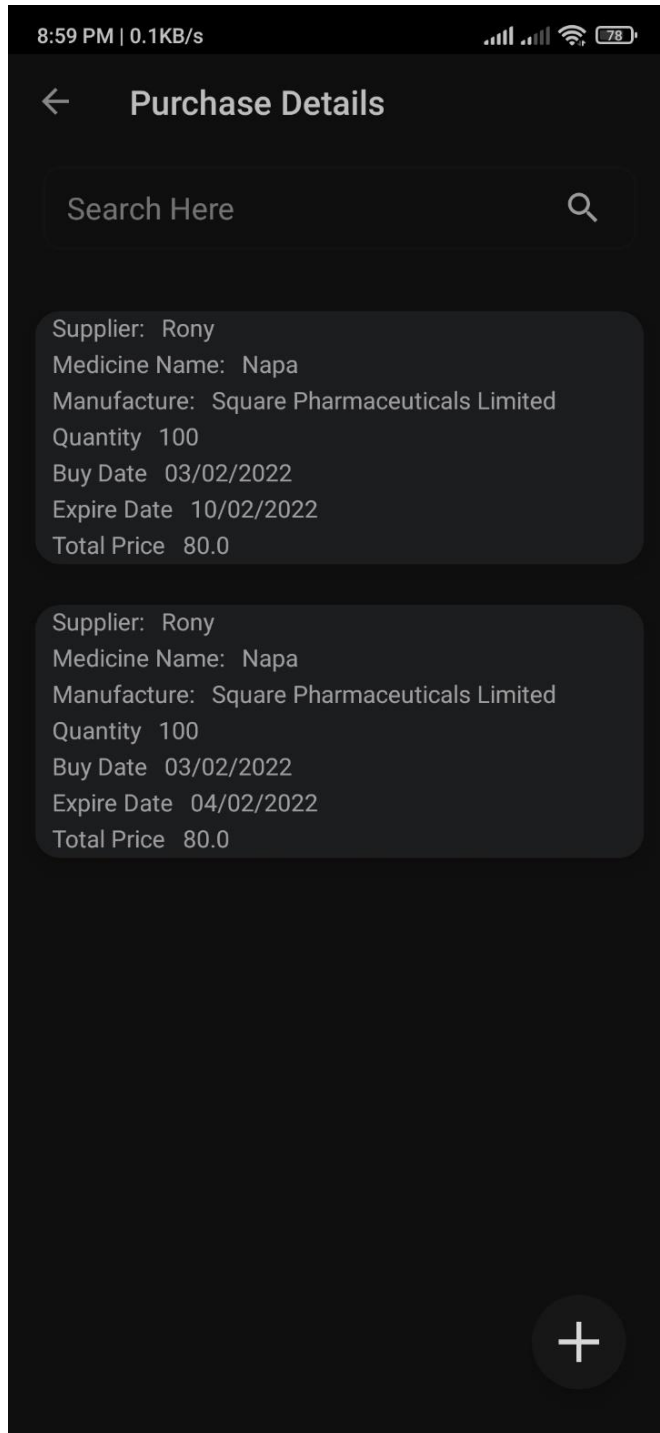


Figure: 7.47: Purchase Details

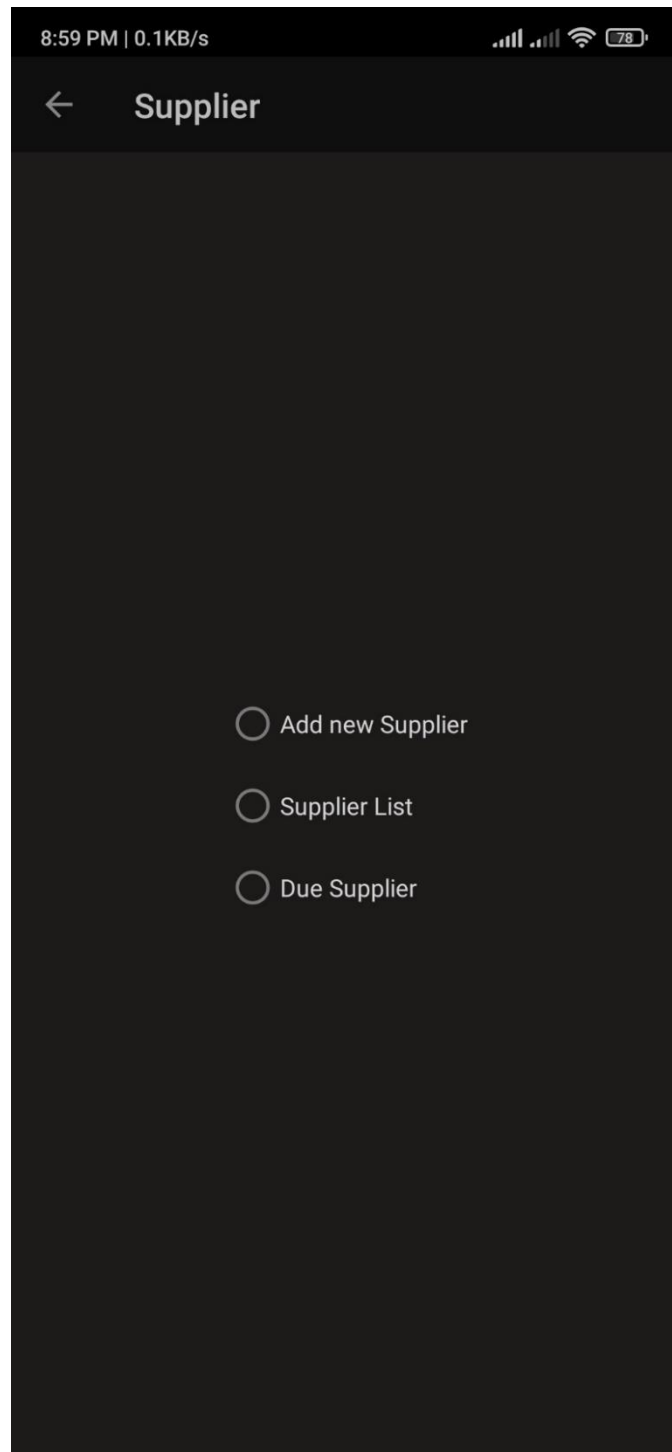


Figure: 7.48: Select Supplier Option

8:59 PM | 0.2KB/s

← Add new Supplier

Add Supplier

Phone

Save

Figure: 7.49: Add Supplier

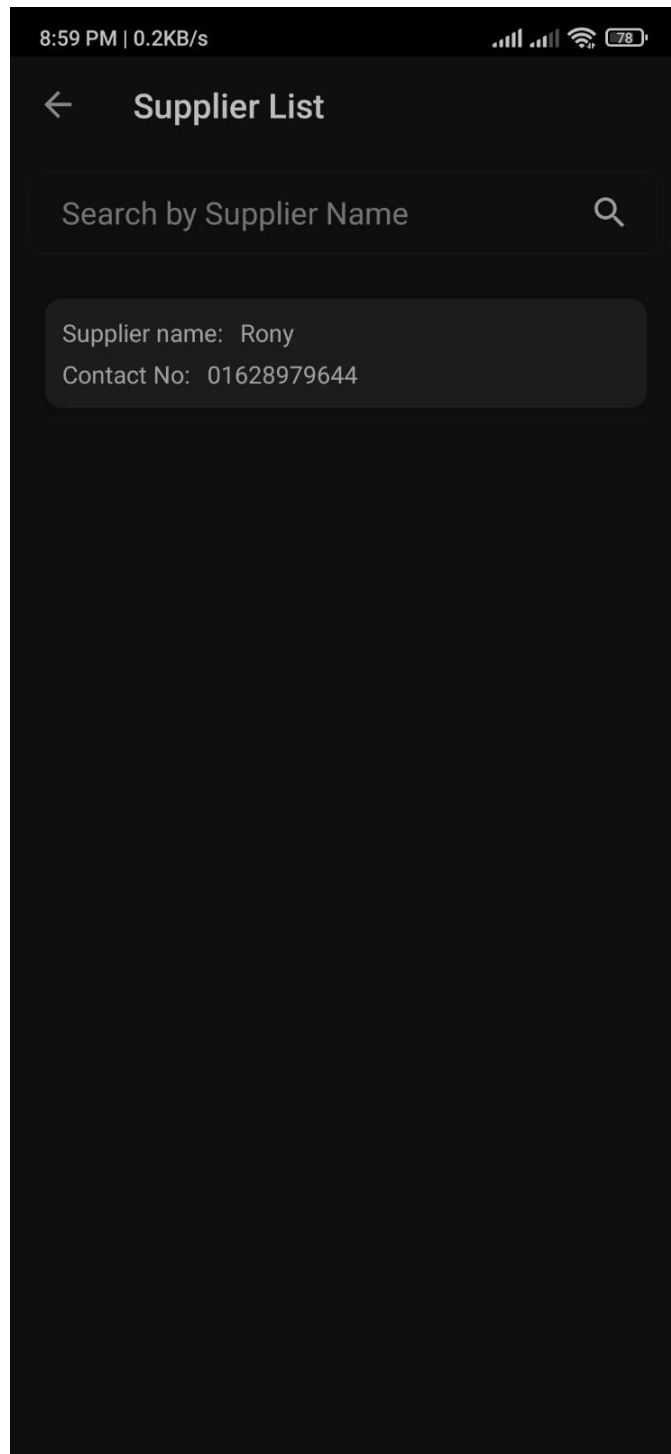


Figure: 7.50: All Supplier List

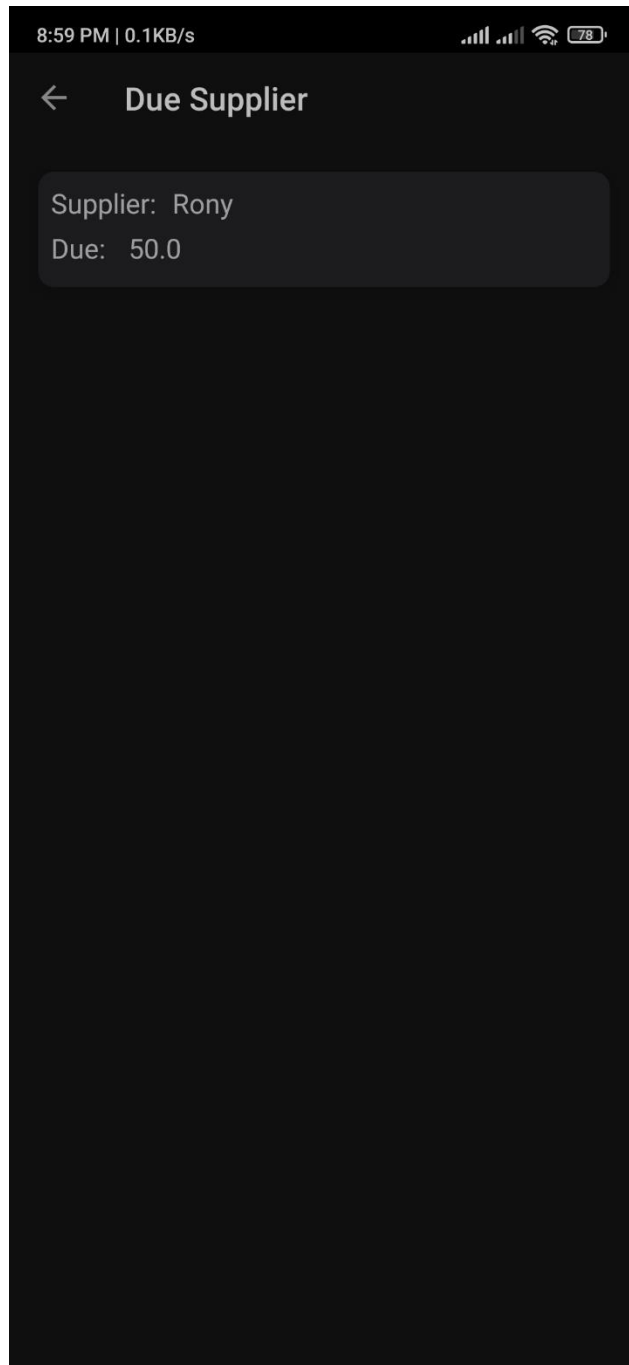


Figure: 7.51: Due Supplier List

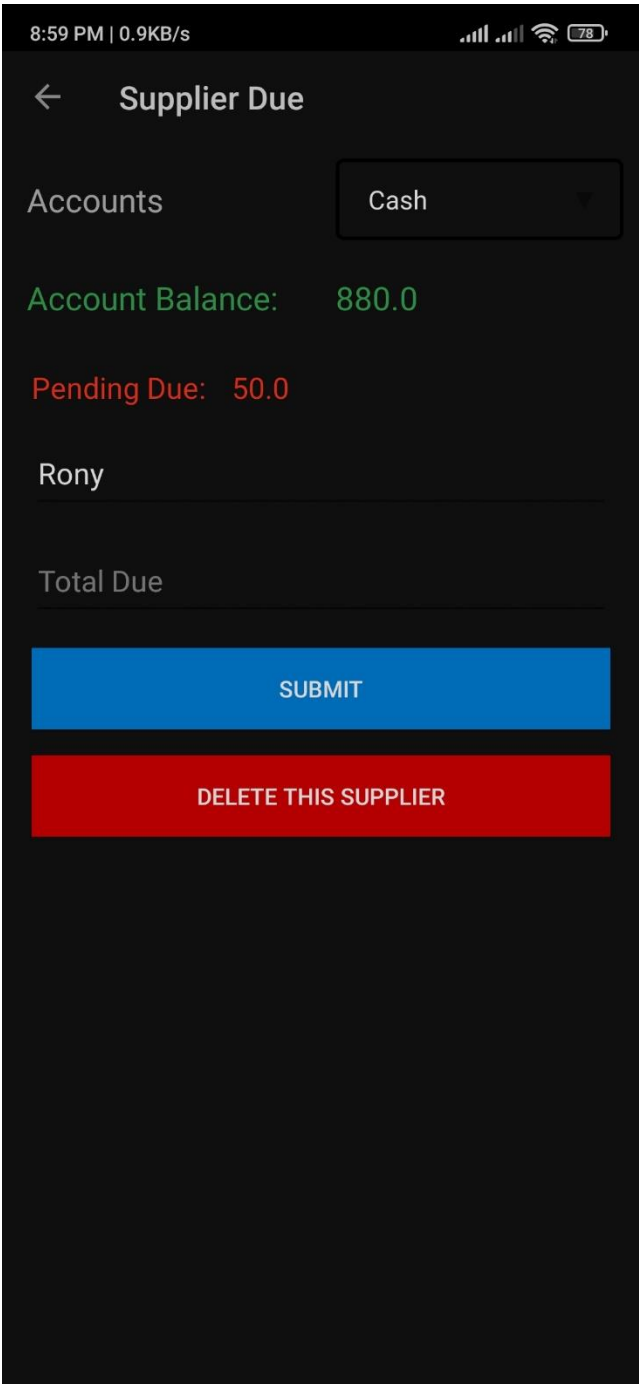


Figure: 7.52: Supplier due Paid

8:58 PM | 2.8KB/s

← Sell

Customer Name Hridoy ▼

Medicine Name Napa ▼

Date

Payment Type Cash ▼

Account Balance: 880.0

Stock quantity 200.0

Sell Quantity

Unit Sell Price

1

Total Price

Paid Amount

Due Amount

SELL

Figure: 7.53: Sell Medicine

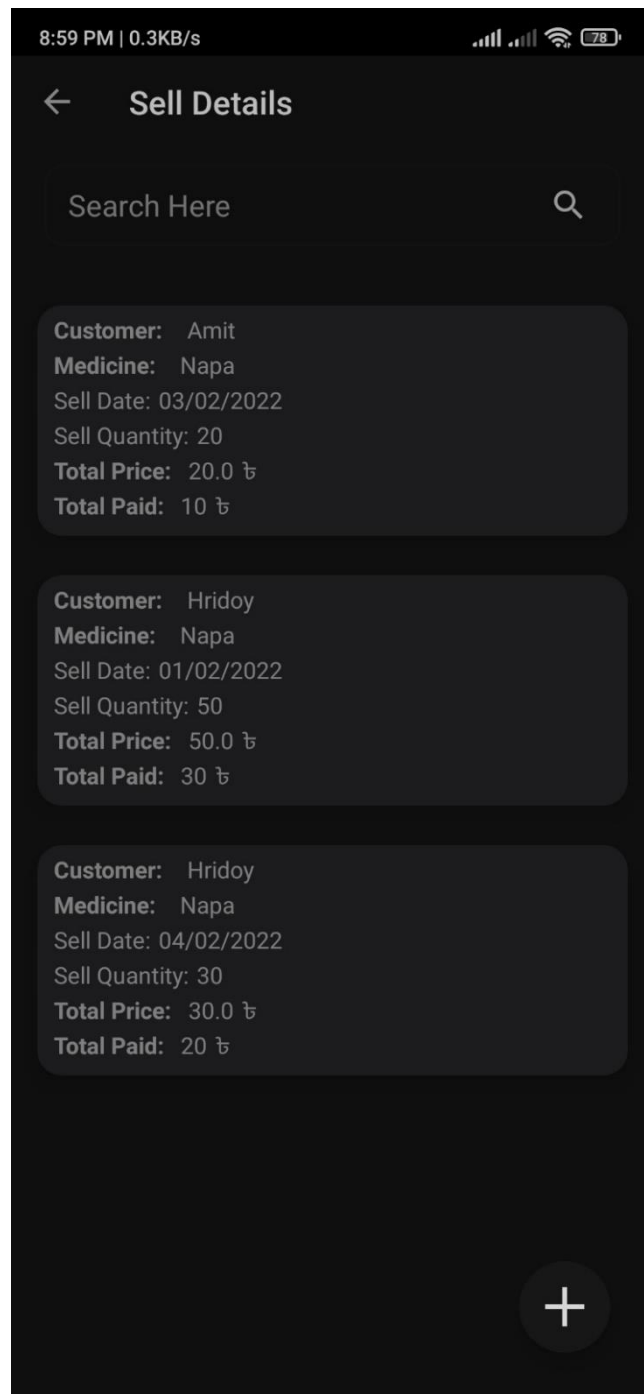


Figure: 7.54: Sell Details

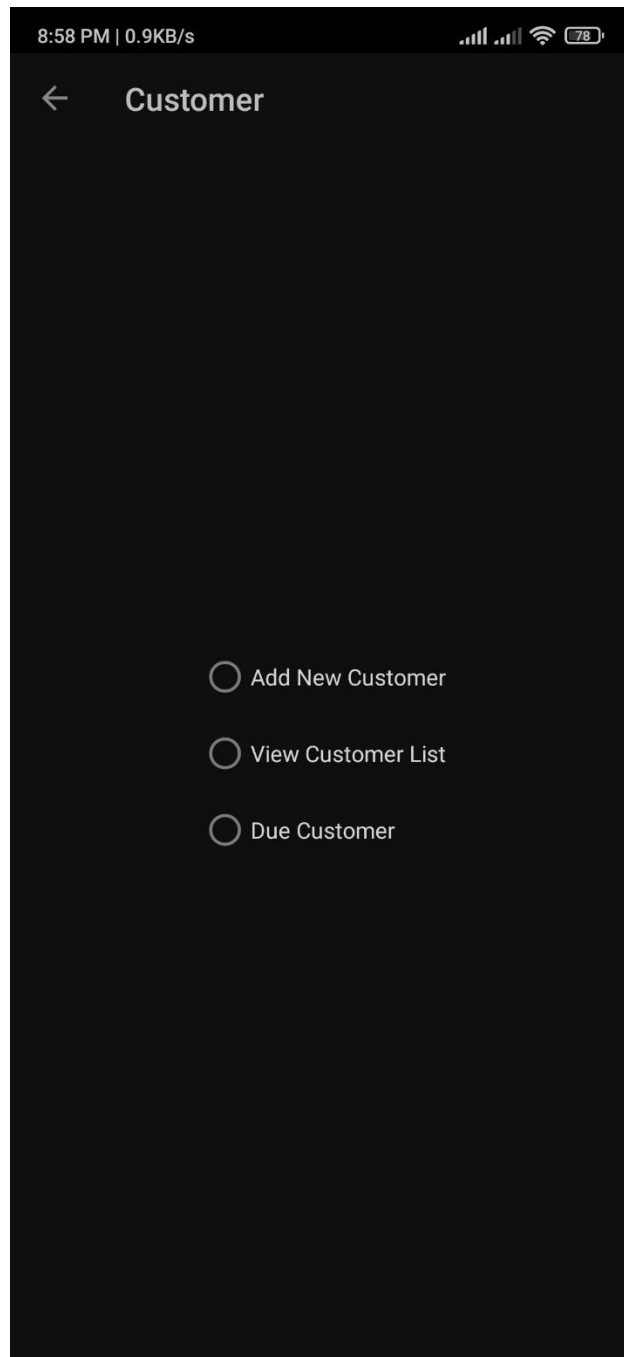


Figure: 7.55: Select Customer Option

8:58 PM | 0.8KB/s

← Add Customer

Customer Name

Phone

Customer Address

SAVE

Figure: 7.56: Add new customer

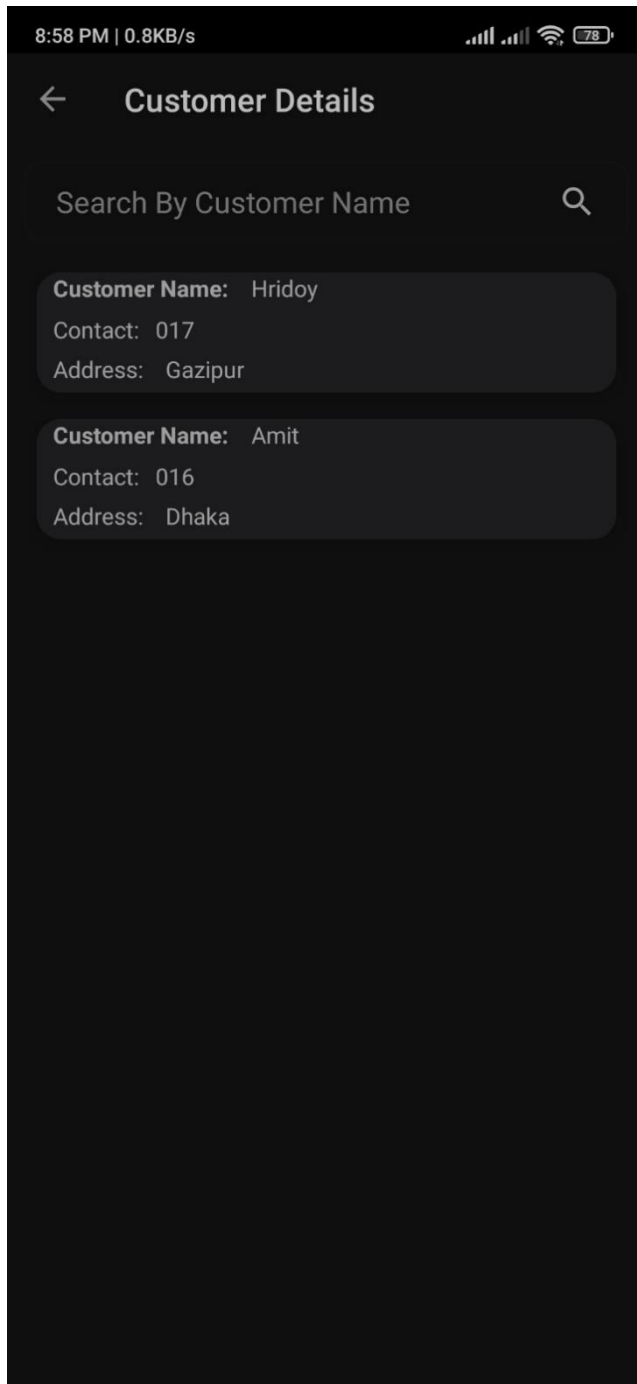


Figure: 7.57: Customer Details

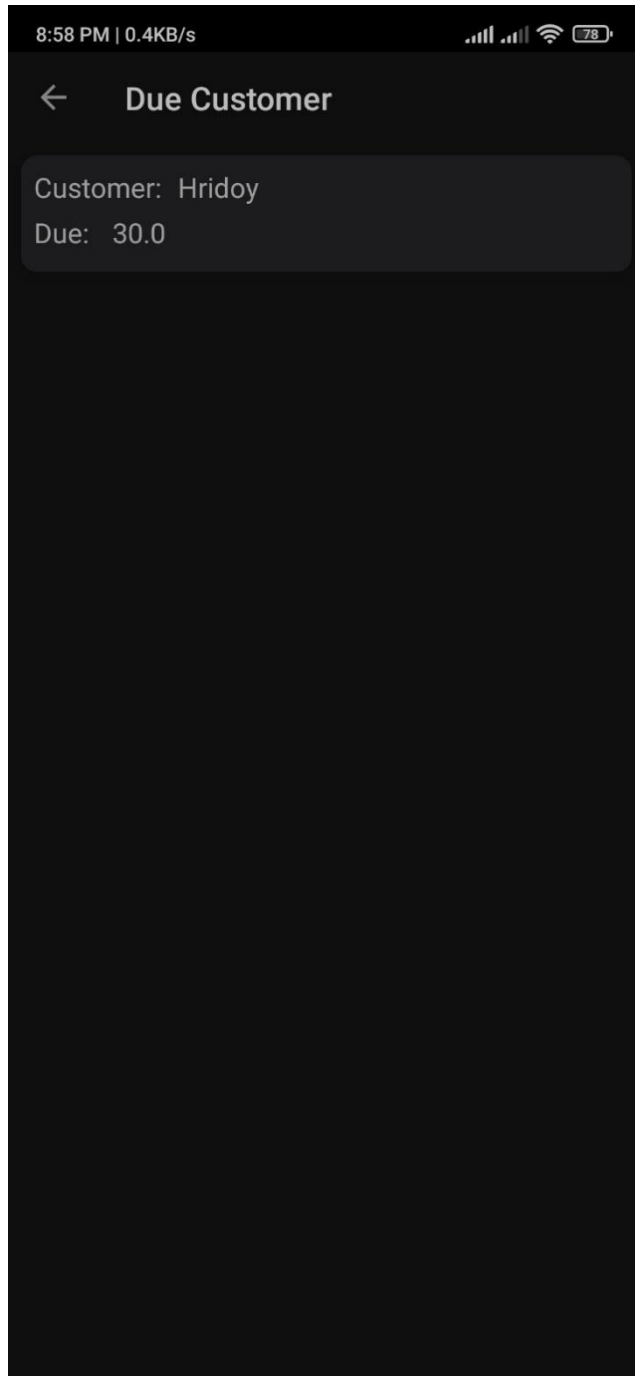


Figure: 7.58: Due Customer Details

8:59 PM | 0.4KB/s

← Customer Due

Accounts Cash ▼

Account Balance: 880.0

Pending Due: 30.0

Hridoy

Total Due

SUBMIT

DELETE THIS CUSTOMER

Figure: 7.59: Customer Due Paid

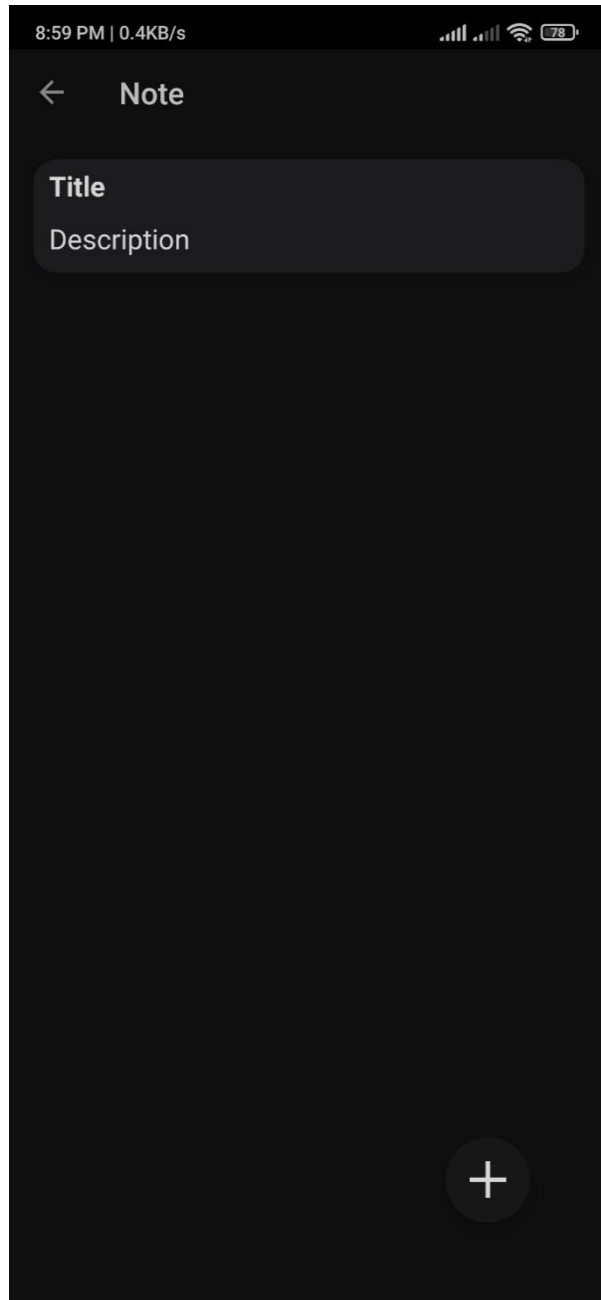


Figure: 7.60: Note Details

Chapter 8: System Quality and Testing

8.1 System Testing

Software verification is the process of judgment a spreadsheet article to detect distinctness's middle from two points' likely recommendation and anticipated manufacturing. Also, to evaluate the feature of A software part. Testing evaluates the character of the fruit. Software experiment is a process that bear be finished during the growth process. In other words, operating system experiment is a proof and confirmation process.

Verification: Verification is the process to make sure the product satisfies the conditions and client imposed at the start of the development phase. In other words, to make sure the product behaves the right way we want.

Validation: Validation is the process to make sure the product satisfies the specified requirements at the end of the development phase. In other words, to make sure the product is built as per customer requirements.

The objectives of software testing are:

- Testing is a process of executing a program with the intent of finding an error.
- A good test case is one that has a high probability of finding an as-yet-undiscovered error.
- A successful test is one that uncovers an as-yet-undiscovered error.
- The design of tests for software can be challenging as the initial design of the product itself. Software can be tested in one of two ways:
- Knowing the specified function that the software has been designed to perform, tests can be conducted that demonstrate each function fully while at the same time searching for errors in each function. This approach is known as black-box testing.
- Knowing the internal workings of software, tests can be conducted to ensure that internal operations are performed according to specifications and all internal components have been adequately exercised. This approach is known as white-box testing.

8.2 Software Testing Strategy

A strategy for software testing integrates software test case design methods into a well-planned series of steps that result in the successful construction of a software. The strategy provides a road map that describes the steps to be conducted as part of testing.

Testing strategy that will be followed in this software project –

- Unit testing
- Integration testing
- Validation testing

The first step in software testing is unit testing. Unit testing concentrates on each unit of the software as implemented in source code. Unit testing focuses on each component individually. The unit test is white-box oriented. Thus, unit testing of this library software will be done after completion of every module or component.

The next step is integration testing. Integration testing is a systematic technique for constructing the program structure while at the same time conducting tests to uncover errors associated with interfacing. The objective of integration testing is to take unit tested components and build a program structure that has been dictated by design.

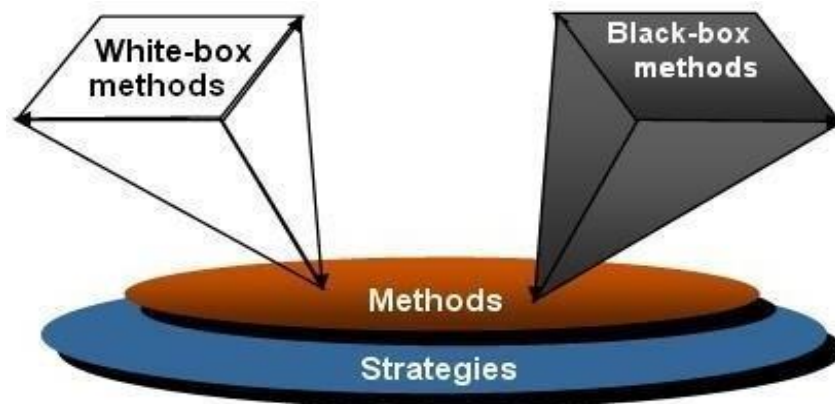
The integration testing strategy that has been chosen for this project is top down testing. Black-box testing method is the most prevalent for integration testing. Top down integration strategy will be used to perform integration testing. Top down integration will be done by breadth-first manner. Breadth-first integration incorporates all components directly subordinate at each level, moving across the structure horizontally.

After the software has been integrated, a set of high order tests are conducted. Hence, the validation criteria that have been mentioned in requirements engineering should be tested. Validation testing provides final assurance that software meets all functional, behavioral and performance requirements. The black-box testing method is exclusively used in validation.

8.3 System Testing Methodology

8.3.1 Black-box Testing

Black-box testing which is also known as behavioral testing focuses on the functional requirements of the software. It enables the software engineer to derive sets of input conditions that will fully exercise all functional requirements for a program. Black-box testing method will be applied to test the modules of LMS.



8.3.2 White-box Testing

White-box testing, which also known as *glass-box testing*, is a test case design method that uses the control structure of the procedural design to derived test cases. Using white-box testing methods, software engineer can derive test cases that,

- Guarantee that all independent paths within a module have been exercised at least once.
- Exercise all logical decisions on their true and false sides.
- Execute all loops at their boundaries and within their operational bounds.
- Exercise internal data structures to ensure their validity.

The modules that contain some complex calculations or decision making code such as check the availability of the library item will be tested using white-box method.

Chapter 9: Conclusion

9.1 Preface

Today is the age of new wisdom and facts and online ideas that is detracting to incident of more effective functional and administration process. To support better and continuing services to the representative of BINARY-IT is a group of Software guru cooperation to hold the service continually. I was profitable and sanctified to take this stroke of good fortune to work some of these effective hard active intimate engineers. My sincere thanks, appreciation and greeting to these great public from the deep unhappy inside my essence.

9.1.1 Practicum and Its Value

In our career development as with most life issues there is direct relationship between effort and reward. To me, practicum can be as a transition from engineering college study life to a real world workplace through hands on experience of engineering practices.

The four years of undergraduate engineering studies gives a student theoretical and practical knowledge. Using that knowledge and observing live operational system, the practicum program clarifies those subjects ‘matters to another level blessed with practical working skills. Considering this fact, it gives us an immense pleasure to say that my practicum was a successful event.

Practical work experience doesn’t have any other alternatives. Before getting into the job, students should have real world work experiences in a major field of study. Nowadays, a day’s recruiter no longer considers just high grades, good communication skills, and part time work experiences. They highly consider the work experiences of an applicant. Students with better work experiences are getting better job opportunities.

BINARY-IT gives me the opportunity of working in a professional working environment. During the internship period I have tried my level best to make my system efficient. I followed the lessons, methods, tools and techniques that I have learned during my study period at IUBAT. Successful software development is a blend of standard development practices, proper theoretical knowledge and the developer’s creativity.

Students of College of Engineering and Technology (CEAT) at IUBAT go for this practicum program carrying 9 credit hours, which goes for a semester long and usually after the completion

of the course work. A report submitted after the completion of the practicum followed by a presentation and a comprehensive examination on the overall four years education.

9.2 Future Plan

- ❖ Medicine return system.
- ❖ Add multiple medicine sell at a time.
- ❖ Online Payment system.
- ❖ Expire medicine information notification.
- ❖ Manage Expire Medicine

9.3 Conclusion

The most generous happening working at BINARY-IT is actually being any of plotting and achieving software. Our most occurrence was around the plotting issue. We have determined plenty of new things that was very obscure to us. We have also discovered a few mechanics issues that help to do better from now on. The following sign will signify some of our mechanics issues that we have gained and executed from this project. The designing skill of an online based project.

- The thinking strategy of online based project.
- New environment of programming languages and make project.
- Developing a new useful project using the existing project.