



Submitted in partial fulfillment of the
requirements for the degree of

Master Of Computer Application

Project

SUBMITTED BY

Abhi Sheth & Jigar Mashru

Report

(Enrolment Number : **220441020** & **220441013**)

UNDER THE ESTEEMED GUIDANCE OF

Banking App

Mr. Uday Shah
Faculty of MCA
Noble University

Certificate

This is to certify that the Project report entitled **Bank App** is a bonafide record of research work done by **Abhi Sheth & Jigar Mashru** Enrollment No.: 220441020 & 220441013 under my supervision and submitted to **Noble University** in partial fulfillment for **MASTER OF COMPUTER APPLICATION (MCA) SEMESTER 3.**

Signature of the Guide

Designation:

PREFACE

There is a wide difference between theory and practical. If one has only theoretical background of any subject, one would not succeed in own aim therefore it is necessary for any person to have acceptable practical knowledge of the concerned subject. As I know MCA is a course based on "**Information Technology**" and it is totally practical field. With only theoretical knowledge one can't be succeeded or one can't be on the peak position.

In the course of MCA designed by the "**Noble University**" they have taken full care of these things and designed the course in such a manner with which student can get theoretical and practical both type of knowledge perfectly. According to the rules & regulation of "Android", I have a subject named "**Bank**". In which we have to create an Application that provide information about financial or bank transaction.

As a MCA student, I have gathered general information about Numbers of this project. Then I decided to develop the application for that. In this application you can Transfer one to other account, Give payment to any payee and in the end of day we see all transaction. You can also create user account with the help of sign up, on this site.

In this project report I have covered all the information, which is required for the android project of MCA student.

I have tried as my best present this project report in such a way that it makes easy to understand the project work.

ACKNOWLEDGEMENT

I am thankful to all, who have helped me in preparing this project. I am very much happy to present this “Project Report”. Before you, expecting that you will acknowledgement it. It is a matter of great pleasure for me that I had an opportunity to express my view on the same.

As a part of my academic study as the student of MCA 3rd – semester. I am required to experienced training android project an institute or industry in order to obtain practical knowledge and inform regarding the same.

At first, I would like to express my & humble thanks & gratitude to the who has provided me such a great, Co-operative & progressive environment.

Secondly at this moment, I would like to express my deepest sense of gratitude to my professor as well as project guides and who have contributed their precious time for the purpose of giving me the correct information with special interest & guidance throughout my project work.

I am also thankful to my classmate and few others who helped me directly or indirectly in solving problem & in making my software project more efficient & good working.

INDEX

Ch. No.	Title	Page No.
1	Introduction and Objects of Project	1,2
2	Tools or Platform	3
3	System Analysis: 3.1 Identification of needs 3.2 Preliminary investigation 3.3 Feasibility study 3.4 Project planning 3.5 Project scheduling 3.6 Software requirement specification 3.7 Data models	4 to 22 4 to 7 8,9 10 to 13 14 15,16 17 to 18 19 to 22
4	System Design	23 to 33
5	Testing: 5.1 Techniques and strategies 5.2 Cost estimation model and SDLC Models 5.3 Future scope and further Enhancement of the project 5.4 Bibliography 5.5 Appendices 5.6 Glossary	34 to 84 34 to 36 37 to 45 46 46 47 47

CHAPTER – 1

INTRODUCTION AND OBJECT OF PROJECT

INTRODUCTION

This project is specially designed for Users, such as Businessman or any person who wants to store their bank transaction separately and securely, in this project user can uploads their bank transaction on “**Bank**” and also can view it.

The “**Bank**” allows user to add bank account, transfer one to other account, pay any payee and also only that particular user can view their account’s transaction and balance by login.

This app is work like cashbook app. In this app we can manage our bank accounts and check it daily transaction and history easily.

The Application is developed in Java, Firebase, XML and some other software like Android Studio, Photoshop for designing purpose.

So, this Application is very useful for all users.

How It Works

In this project first of all a Login Screen is displayed for all Users if user already completed his/her registration then he/she can select login option.

Registration page asks about user's basic information such as Name, email, username and at last user have to set their password.

If User already registered, they have to enter user name and password to in login page to use this app.

If user wants to use “**Bank**” then they have to Register or login first.

If the user is Registered or Login first they can add their account on app by clicking on **Account** and they can also view their account details by clicking on **Account**.

CHAPTER – 2

TOOLS OR PLATFORM

Requirements Details

SOFTWARE: -

- Android Studio

HARDWARE: -

- Processor : Intel(R) Core(TM) i3
- RAM : 4 GB or higher
- Hard Disk : 500 GB or higher
- SSD : 256 GB or higher

TOOLS USED: -

- Front End : Android XML
- Back End : JAVA
- Operating System : Windows 10

CHAPTER – 3

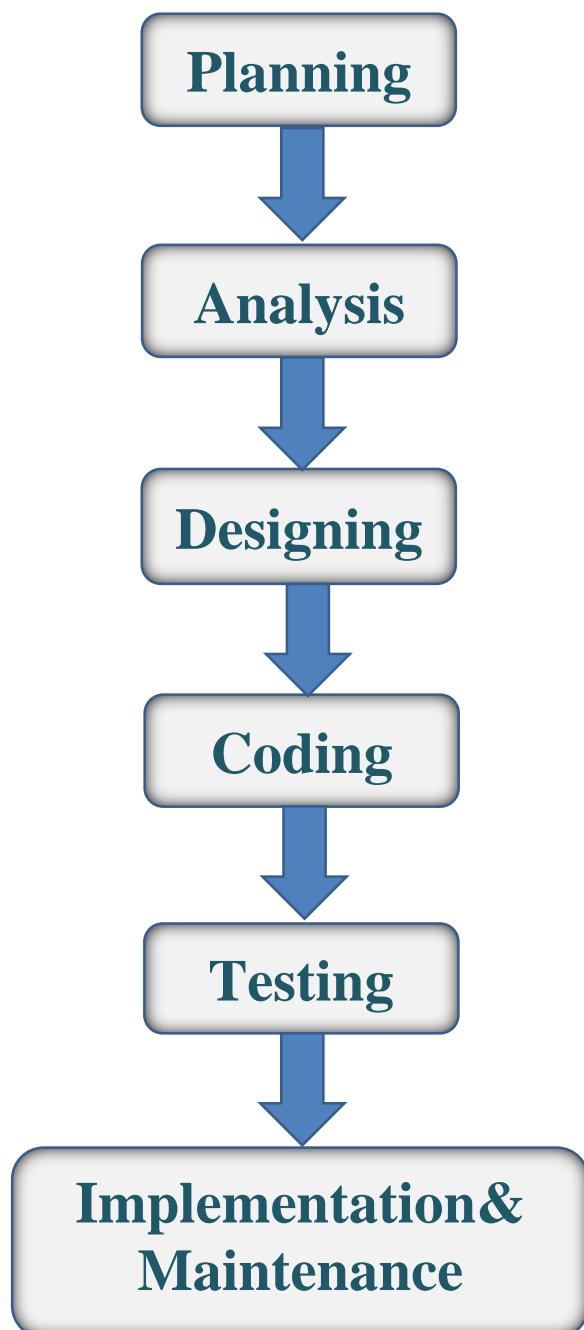
SYSTEM ANALYSIS

- 3.1 Identification of needs
- 3.2 Preliminary investigation
- 3.3 Feasibility Study
- 3.4 Project Planning
- 3.5 Project Scheduling (PERT Chart and Gantt chart)
- 3.6 Software Requirement Specification
- 3.7 Data Models

3.1 IDENTIFICATION OF NEEDS

- The User wants to add account they must be registered to our App.
- User need to provides username and email when creating an profile.

SDLC MODLE



SDLC STEP

- Requirement Gathering
- Requirement Specification and Analysis
- Design
- Coding
- Maintenance

REQUIREMENT GATHERING: -

In this phase of SDLC necessary information are collected. We collect the information in this phase through questionnaire and online websites also. We are also note down the requirements of the User.

REQUIREMENT SPECIFICATION AND ANALYSIS: -

The requirements are being specify in this phase. Analyst checks how the current system is working and then plans how to develop the proposed system and implement it.

 **DESIGN: -**

In this phase the design of both front-end and back-end is decided. In this phase we using Android XML for designing, and designing of activity layouts are also done properly in this phase.

 **CODING: -**

After the completion of designing there is a turn for coding. Coding is completed in various files. In most case coding takes the more time than the design.

 **TESTING: -**

In testing phase first unit level testing is done. In unit level testing the software is tested in an individual file. After the unit testing of each file the integration testing done. In integration testing the files are tested into their own package. At last, in the system testing the whole system is run on the INTERNET ENVIRONMENT.

 **MAINTENANCE: -**

The proper maintenance is given as indicate in the SRS (Software Requirement Specification). It is a responsibility of Developer to maintain the software during its criteria mentioned in SRS document.

3.2 - PRELIMINARY INVESTIGATION

OUR OBJECTIVES: -

- User can add their account.
- User can also deposit amount in their account if they received any amount.
- User can transfer amount one to other account.
- User can pay to payee.
- User can check their transaction easily.

 **PLATFORM SPECIFICATION: -**

- We have used windows based developing graphical based applications our software will provide support GUI efficiently. It must for our system because for our project.
- It is very useful and support of Java and XML control and component.
- Use of Linear Layout and Relative Layouts for PDF Uploading form and display Form in app.
- JAVA platform is mostly use for development of Android Application.
- Regular version of operating system use with it.
- Always need to back end tool for developed android Application with front tool.
- Starting a new business all of the organization have some requirement for it.
- There are many reasons for starting a new business or modify the old system.
- Business require set is depend on top level organization that is manager, administrator etc.

3.3 – FEASIBILITY STUDY

- A feasibility analysis usually involves a through assessment of the operational (need), financial and technical aspects of a proposal.
- Feasibility study is the test of the system proposal made to identify whether the user needs may be satisfied using the current software and hardware technologies, whether the system will be cost effective from a business point of view and whether it can be developed with the given budgetary constraints.
- A feasibility study should be relatively cheap and done at the earliest possible time.
- Depending on the study, the decision is made whether to go ahead with a more detailed analysis.
- When a new project is proposed, it normally goes through feasibility assessment.
- Feasibility study is carried out to determine whether the proposed system is possible to develop with available resources and what should be the cost consideration.
- Facts considered in the feasibility analysis were.

1) TECHNICAL FEASIBILITY: -

- Technical feasibility is considered. In terms of technical requirements and their availability in the market. It determines whether the current level of technology supports the proposed system or not. The technical possibility of proposed system is as follows.
- The unit does possess the hardware as well as related software for the project.
- The proposed system does not require much technical detail.
 - ✓ It just requires window operating system.
 - ✓ The organization has already purchased all the enough devices for latest technical.
 - ✓ These technical specifications are easily available in the market.
- Hence, the proposed system is technically feasible.

2) ECONOMICAL FEASIBILITY STUDY: -

- Economic justification is generally the “Bottom Line” consideration for most systems.
- Economic justification includes a broad range of concerns that includes cost benefit analysis.
- In this we weight the cost and the benefits associated with the candidate system and if it suits the basic purpose of the organization i.e. profit making, the project is making to the analysis and design phase.
- The financial and the economic questions during the preliminary investigation are.

Verified to Estimate the Following:

- ✓ The cost to conduct a full system investigation.
- ✓ The cost of hardware and software for the class of application being considered.
- ✓ The benefits in the form of reduced cost.
- ✓ The proposed system will give the minute information, as a result the performance is improved which in turn may be expected to provide increased profits.
- This feasibility checks whether the system can be developed with the available funds. This can be done economically if planned judicially, so it is economically feasible. The cost of project depends upon the number of man hours required.

3) OPERATIONAL FEASIBILITY STUDY: -

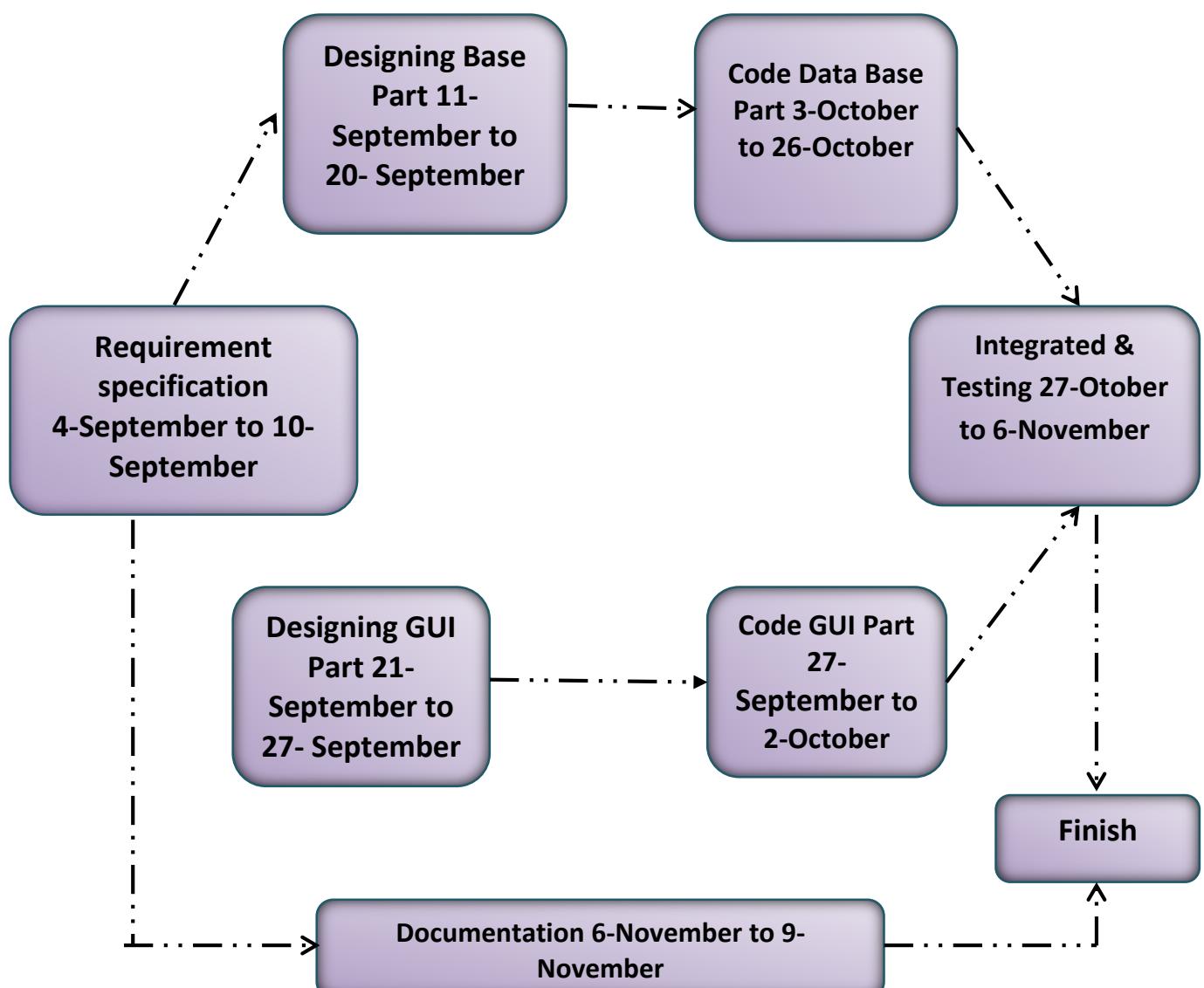
- It is mainly related to human organizations and political aspects. The points to be considered are:
 - ✓ What changes will be brought with the system?
 - ✓ What organization structures are disturbed?
 - ✓ What new skills will be required?
 - ✓ Do the existing staff members have these skills? If not, can they be trained in due course of time?
- The system is operationally feasible as it very easy for the End users to operate it. It only needs basic information about Windows platform.

3.4 – PROJECT PLANNING

No.	Task Name	Start	Finish	Duration
1.	Planning	04-September	10-September	7 Days
2	Analysis	11-September	20-September	10 Days
3	Designing	21-September	02-October	12 Days
4	Coding	03-October	26-October	24 Days
5	Implementation	27-October	01-November	6 Days
6	Testing	02-November	06-November	5 Days
TOTAL		64 DAYS		

3.5 – PROJECT SCHEDULING

 **PERT CHART:** - This chart represents the development of our system Date wise



 **GANTT CHART: -**

This is a graphical representation of the date wise development of our project system. It is very similar to the PERT chart except that it is represented graphically.

ID	TASK NAME	September 23	September 23	September/October 23	October/November 23
1	Planning	04-September to 10-September			
2	Analysis		11-September to 20-September		
3	Designing			21-September to 02-October	
4	Coding				03-October to 26-October
5	Implementation				27-October to 01-November
6	Testing				02-November to 06-November

3.6 Software Requirement

Android Studio: -

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

- A flexible Gradle-based build system
- A fast and feature-rich emulator
- A unified environment where you can develop for all Android devices
- Apply Changes to push code and resource changes to your running app without restarting your app
- Code templates and GitHub integration to help you build common app features and import sample code
- Extensive testing tools and frameworks
- Lint tools to catch performance, usability, version compatibility, and other problems
- Java and XML support
- Built-in support for [Google Cloud Platform](#), making it easy to integrate Google Cloud Messaging and App Engine

This page provides an introduction to basic Android Studio features. For a summary of the latest changes.

 **SQLite Database: -**

SQLite is a open source SQL database that stores data to a text file on a device. Android comes in with built in SQLite database implementation.

SQLite supports all the relational database features. In order to access this database, you don't need to establish any kind of connections for it like JDBC, ODBC etc....

 **Key Features:-**
1. Authentication

- It supports authentication using passwords, phone numbers, Google, Facebook, Twitter, and more. The Firebase Authentication (SDK) can be used to manually integrate one or more sign-in methods into an app.

2. Real time database

- Data is synced across all clients in real time and remains available even when an app goes offline.

3. Hosting

- Firebase Hosting provides fast hosting for a web app; content is cached into content delivery networks worldwide.

4. Test lab

- The application is tested on virtual and physical devices located in Google's data centers.

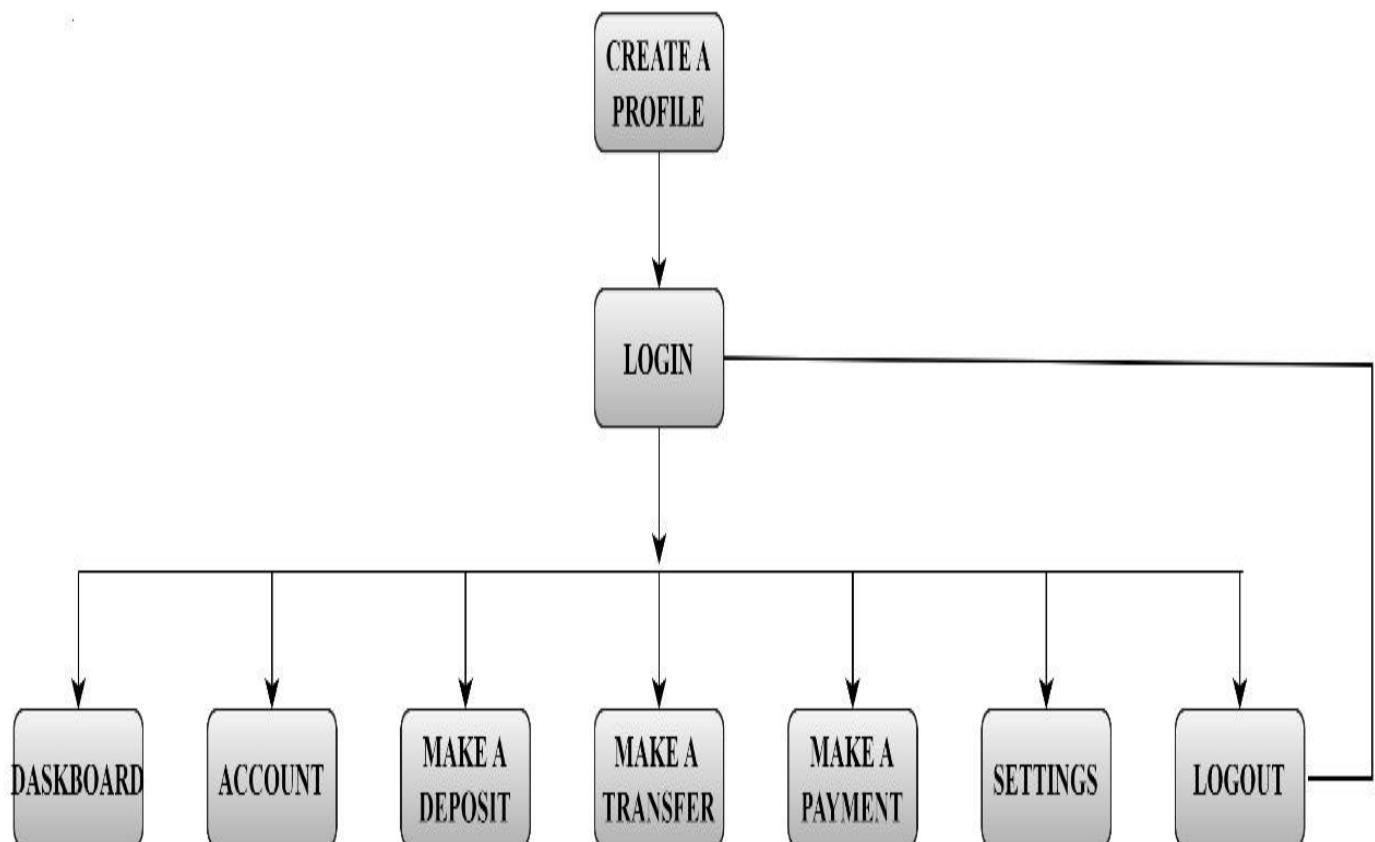
5. Notifications

- Notifications can be sent with firebase with no additional coding.

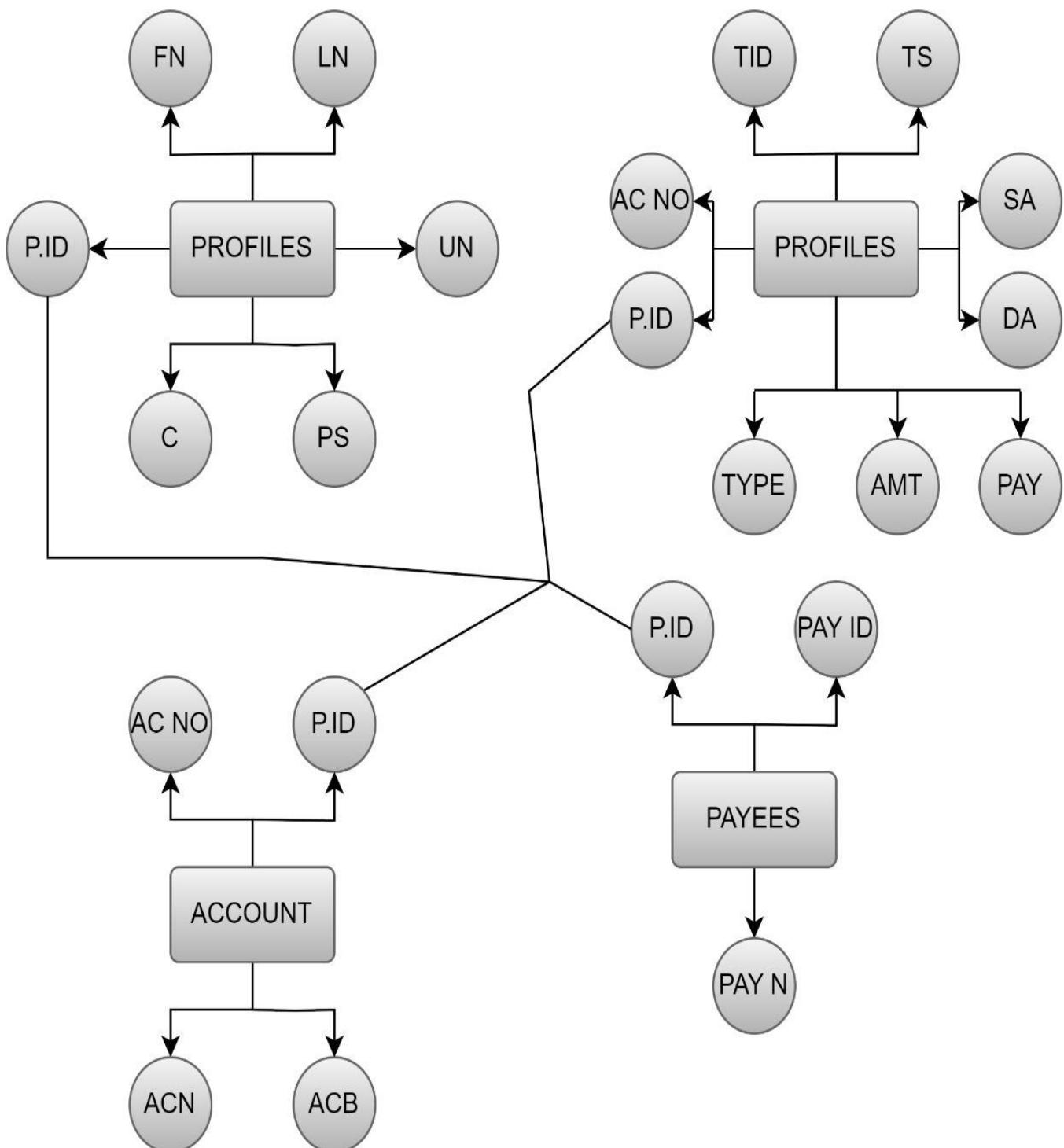
3.7 Data Models

Data Flow Diagram: -

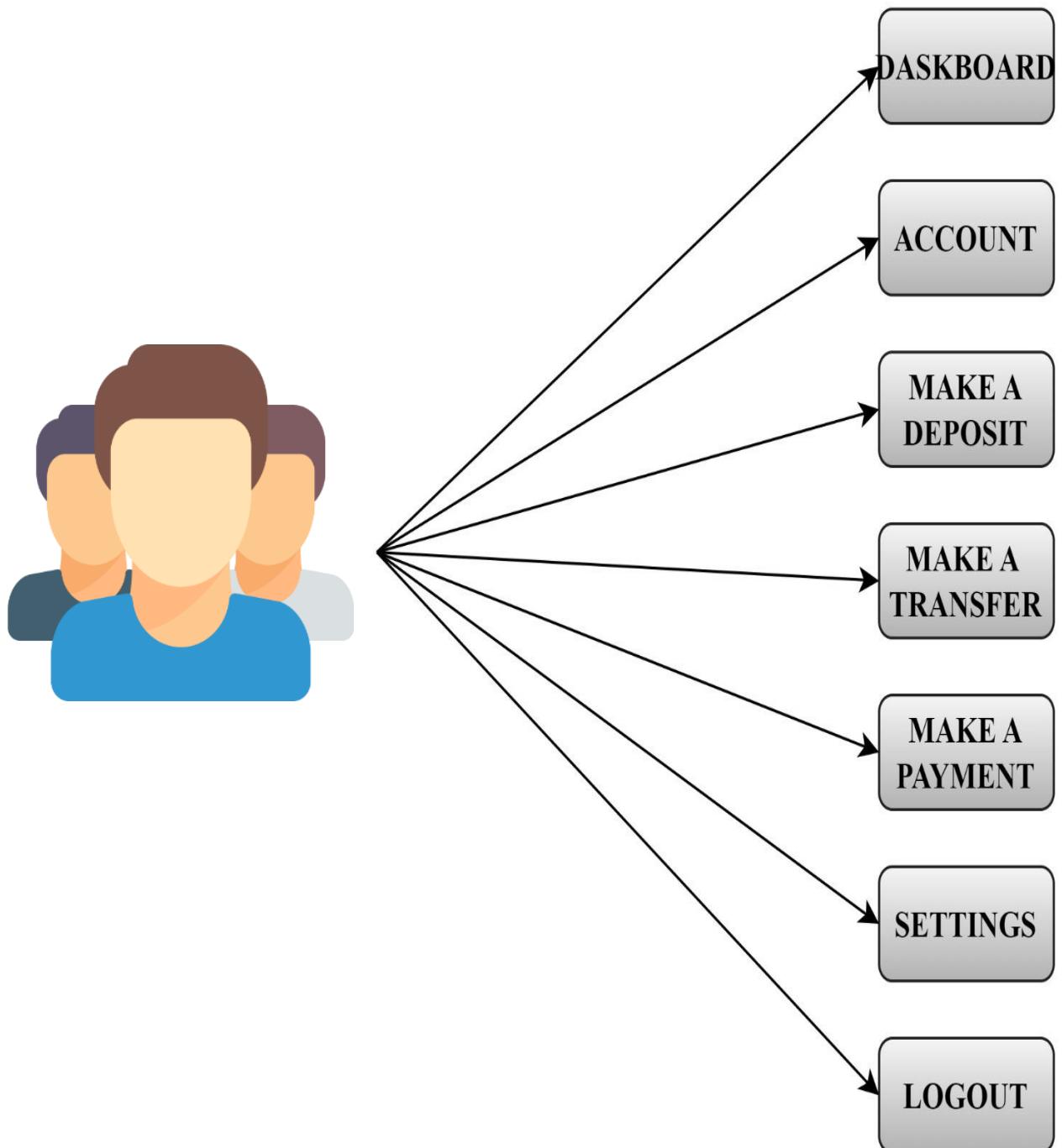
1) Data Flow Diagram: -



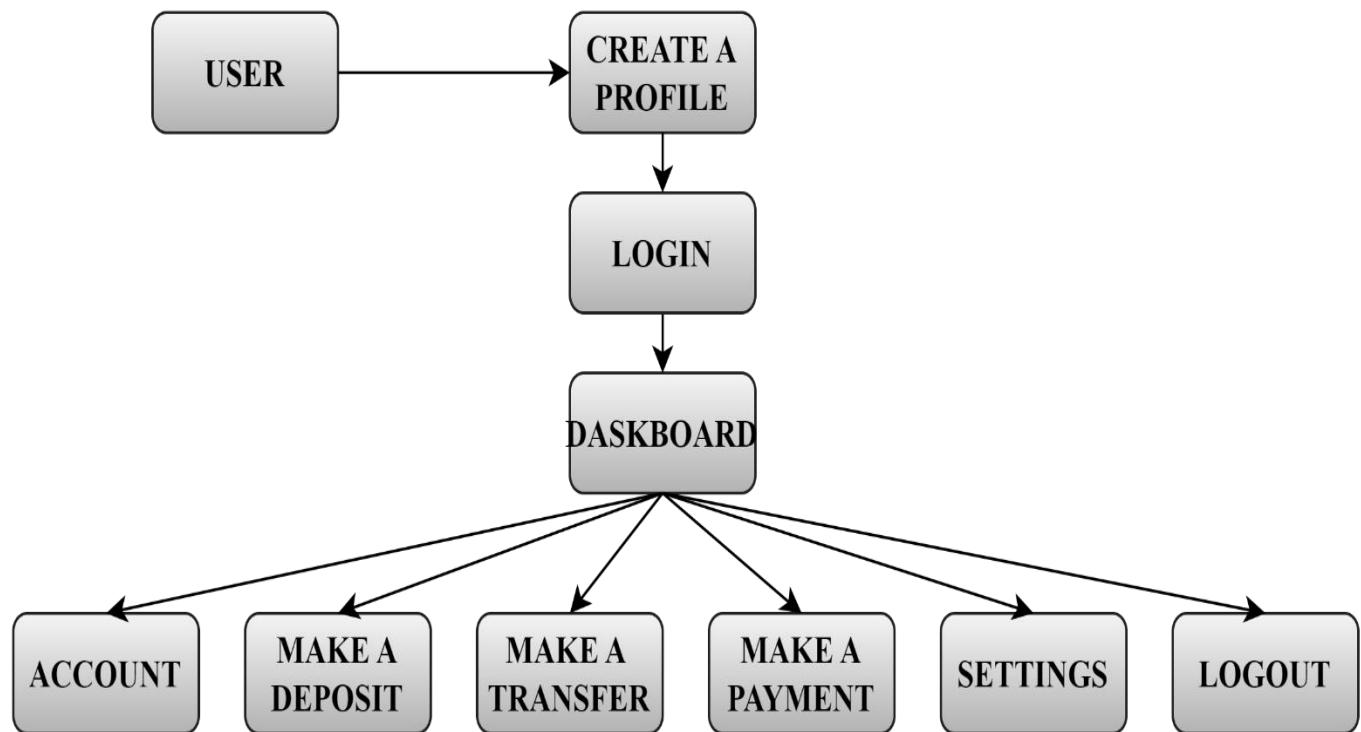
2) Er-Diagram: -



❖ Case Diagram: -



Activity Diagram

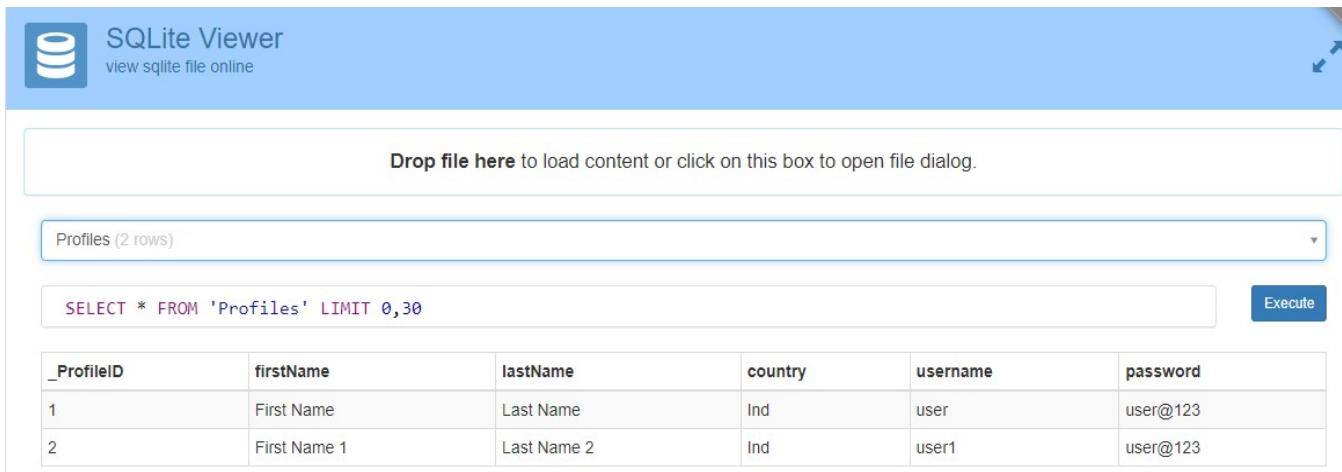


CHAPTER – 4

System Design

Database Structure & Tables: -

1) Profiles



The screenshot shows the SQLite Viewer interface. At the top, there's a blue header bar with the title "SQLite Viewer" and a sub-instruction "view sqlite file online". Below the header is a large input field with the placeholder "Drop file here to load content or click on this box to open file dialog.". Underneath this is a table titled "Profiles (2 rows)". The table has six columns: "_ProfileID", "firstName", "lastName", "country", "username", and "password". The data is as follows:

_ProfileID	firstName	lastName	country	username	password
1	First Name	Last Name	Ind	user	user@123
2	First Name 1	Last Name 2	Ind	user1	user@123

At the bottom of the viewer, there's a SQL query input field containing "SELECT * FROM 'Profiles' LIMIT 0,30" and a "Execute" button.

2) Accounts

SQLite Viewer
view sqlite file online

Drop file here to load content or click on this box to open file dialog.

Accounts (4 rows)

```
SELECT * FROM 'Accounts' LIMIT 0,30
```

<u>ProfileID</u>	<u>AccountNo</u>	<u>AccountName</u>	<u>AccountBalance</u>
1	A1	Acc 1	7000
1	A2	Acc 2	2000
2	A1	U1A1	8000
2	A2	U1A2	1000

3) Payees

SQLite Viewer
view sqlite file online

Drop file here to load content or click on this box to open file dialog.

Payees (4 rows)

```
SELECT * FROM 'Payees' LIMIT 0,30
```

<u>ProfileID</u>	<u>PayeeID</u>	<u>PayeeName</u>
1	P1	Payee1
1	P2	Payee 2
2	P1	U1payee1
2	P2	U1 Payee 2

4) Transactions

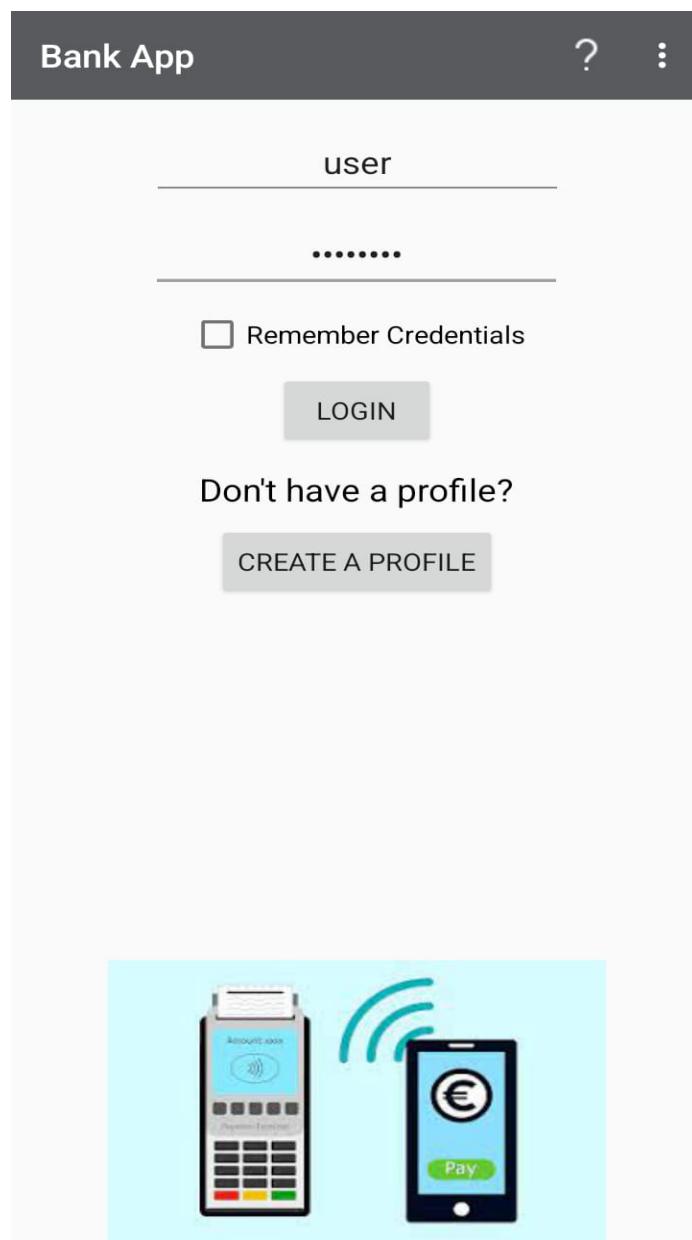
 SQLite Viewer
view sqlite file online

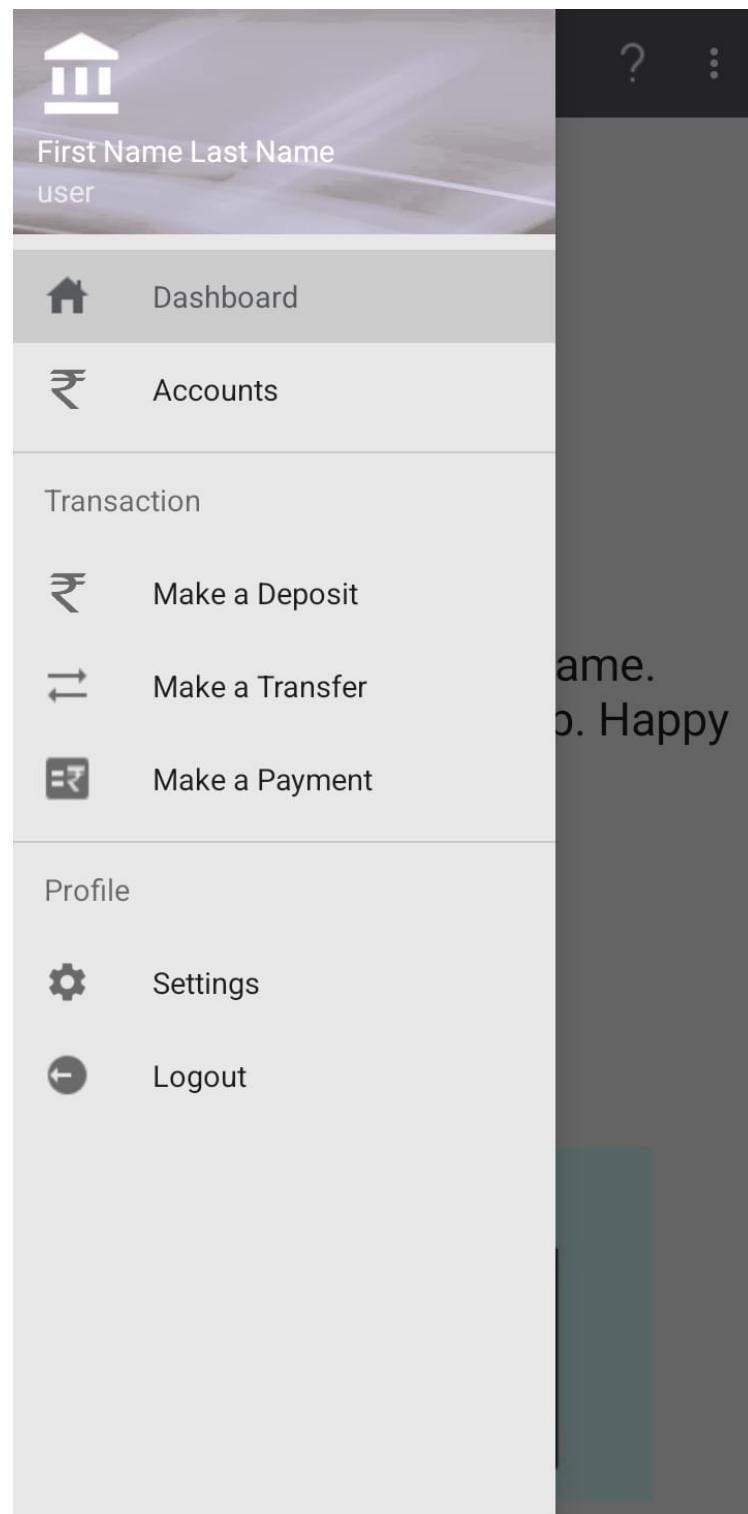
Drop file here to load content or click on this box to open file dialog.

Transactions (12 rows)

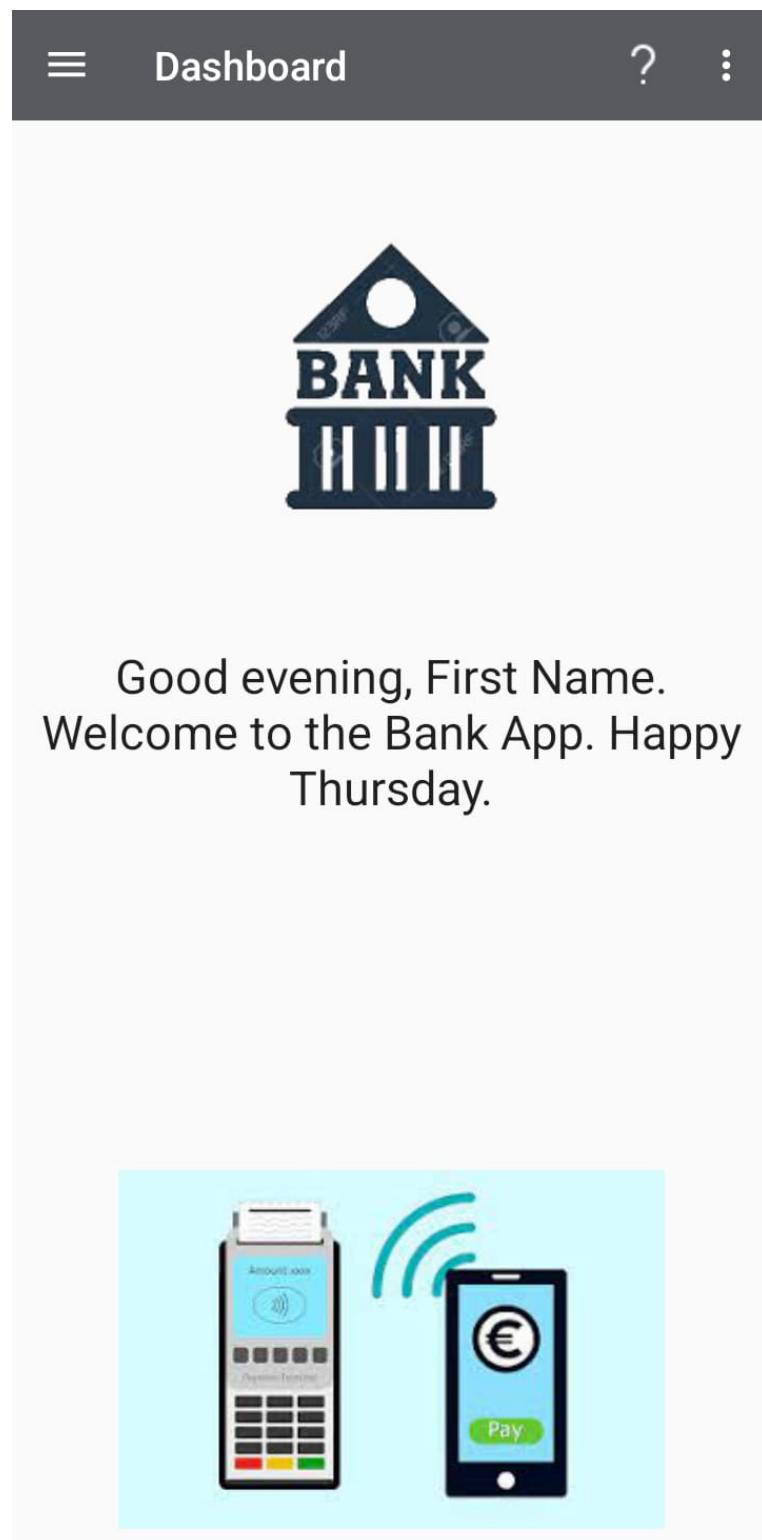
```
SELECT * FROM 'Transactions' LIMIT 0,30
```

_ProfileID	_AccountNo	_TransactionID	Timestamp	SendingAccount	DestinationAccount	Payee	Amount	Type
1	A1	T1-D1	2023/12/07 - 06:55 pm	null	null	null	10000	DEPOSIT
1	A2	T1-D1	2023/12/07 - 07:34 pm	null	null	null	1000	DEPOSIT
1	A1	T2-T1	2023/12/07 - 07:34 pm	Acc 1 (A1)	Acc 2 (A2)	null	1000	TRANSFER
1	A2	T2-T1	2023/12/07 - 07:34 pm	Acc 1 (A1)	Acc 2 (A2)	null	1000	TRANSFER
1	A1	T3-P1	2023/12/07 - 07:51 pm	null	null	Payee1 (P1)	1000	PAYMENT
1	A1	T4-P2	2023/12/07 - 07:52 pm	null	null	Payee 2 (P2)	1000	PAYMENT
2	A1	T1-D1	2023/12/07 - 08:00 pm	null	null	null	10000	DEPOSIT
2	A2	T1-D1	2023/12/07 - 08:01 pm	null	null	null	500	DEPOSIT
2	A1	T2-T1	2023/12/07 - 08:01 pm	U1A1 (A1)	U1A2 (A2)	null	500	TRANSFER

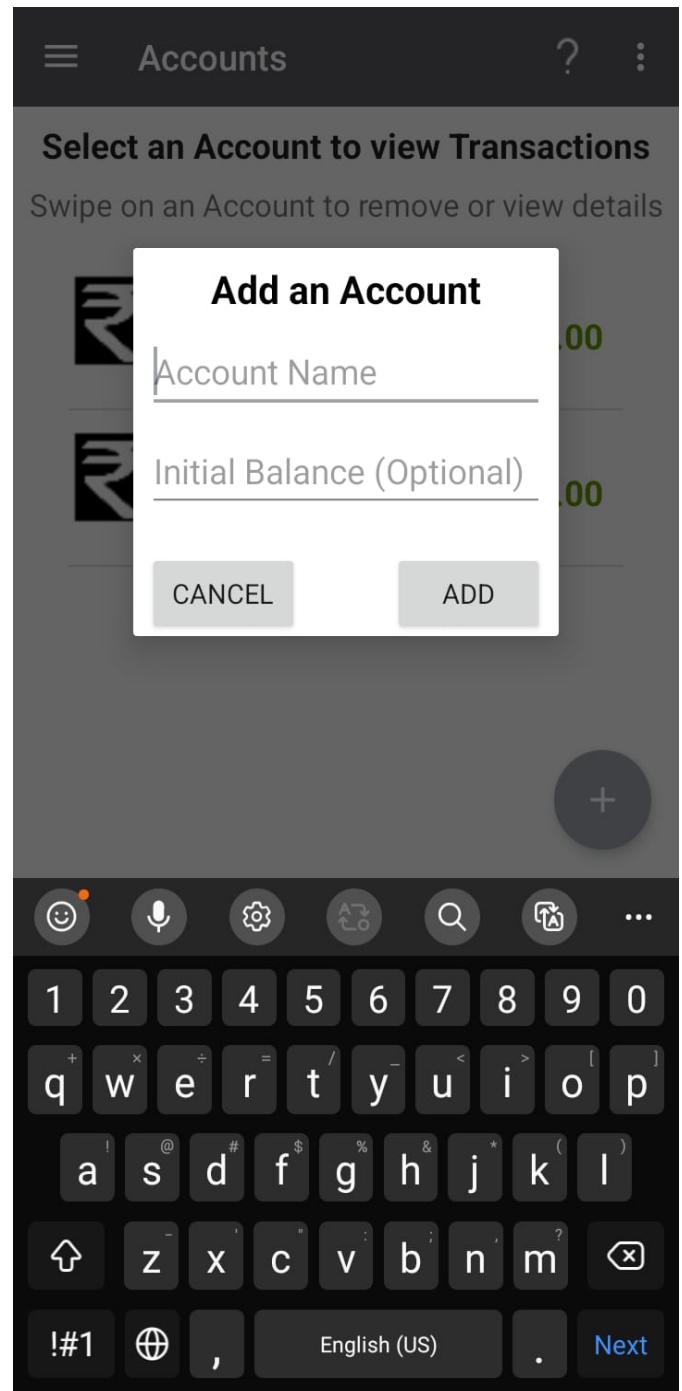
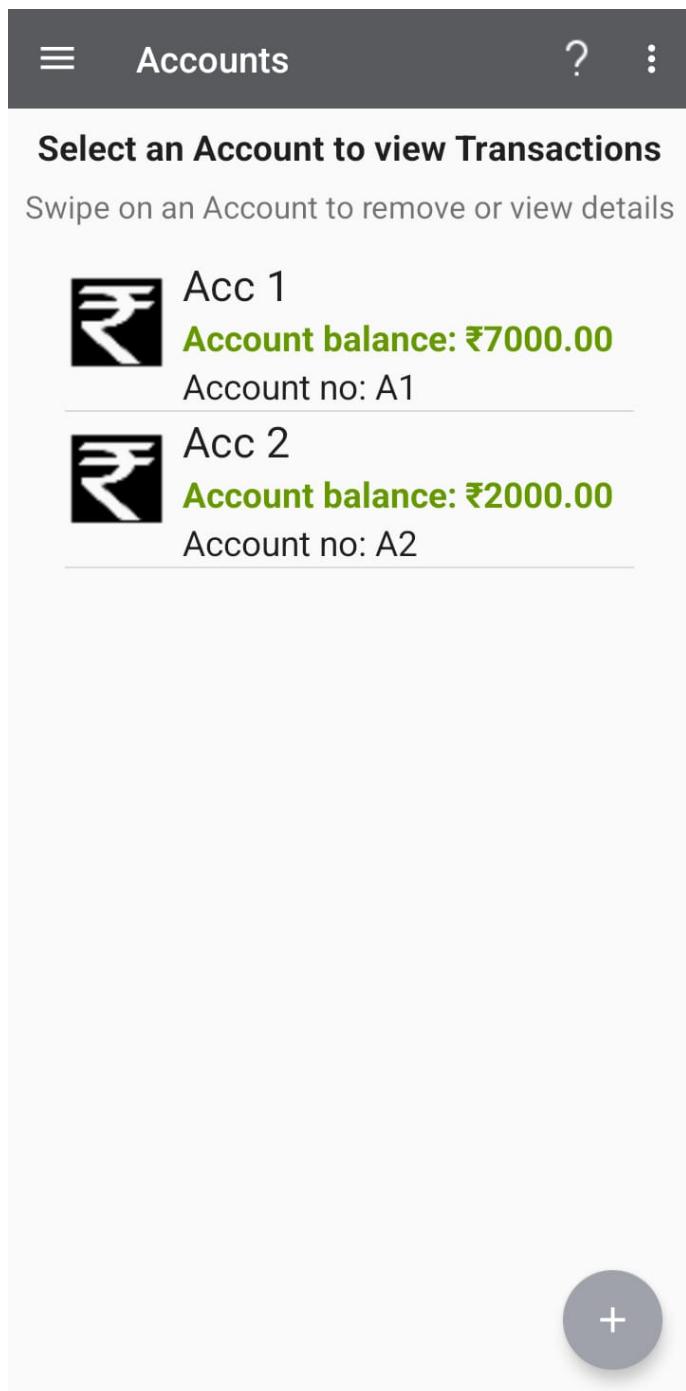
 **Interface:** -**1) Login Page**

2) Menu

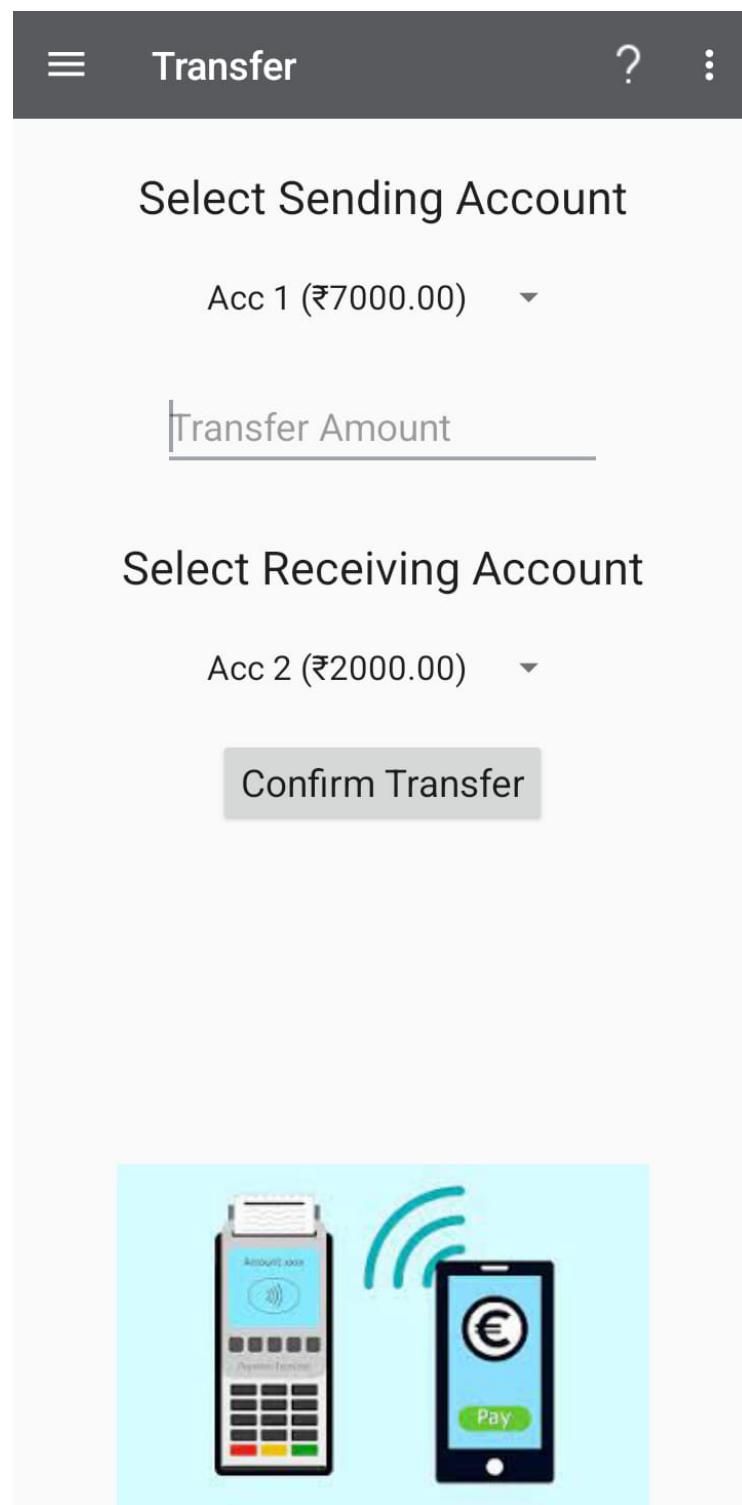
3) Dashboard Page



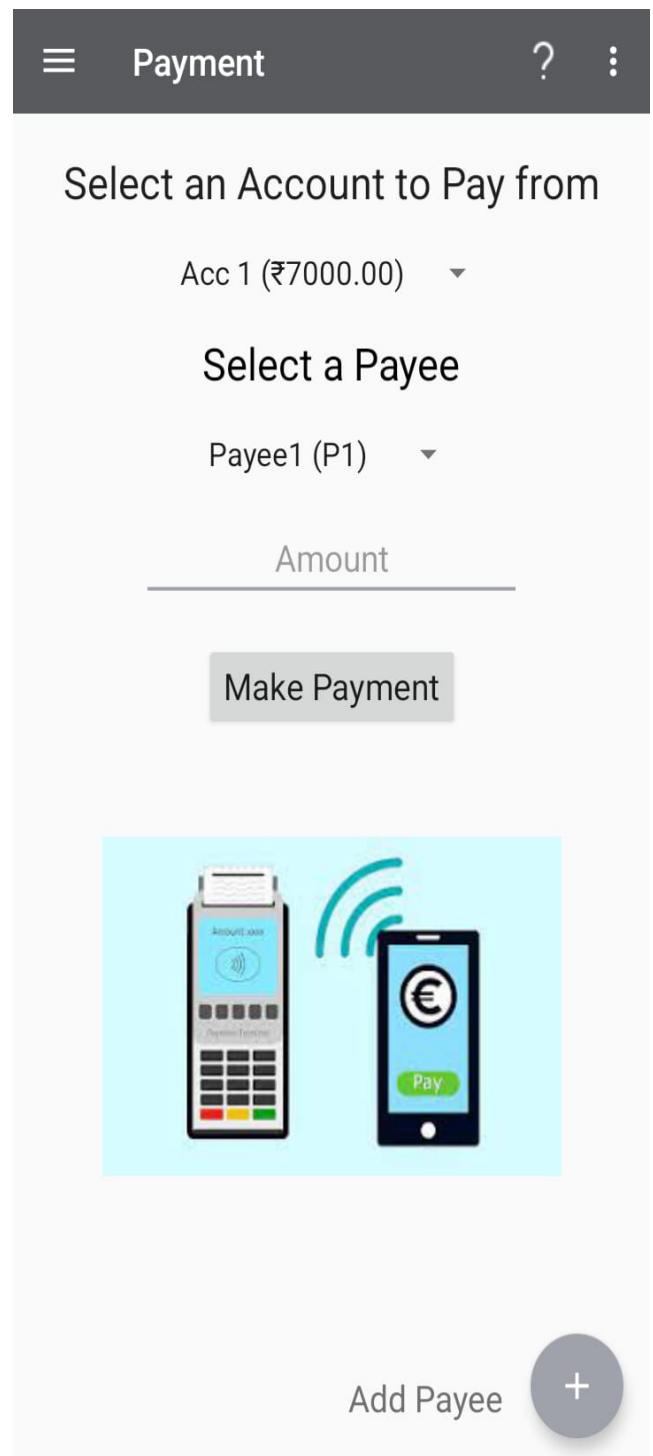
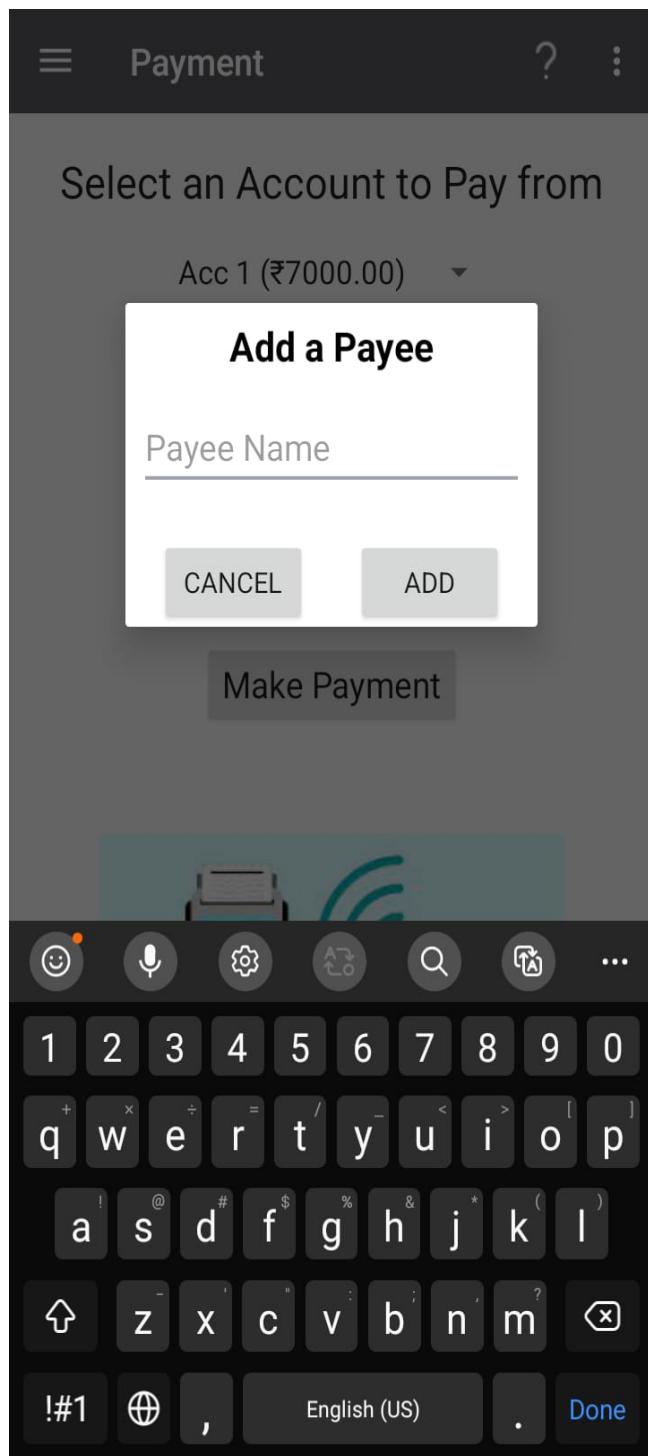
4) Accounts Page



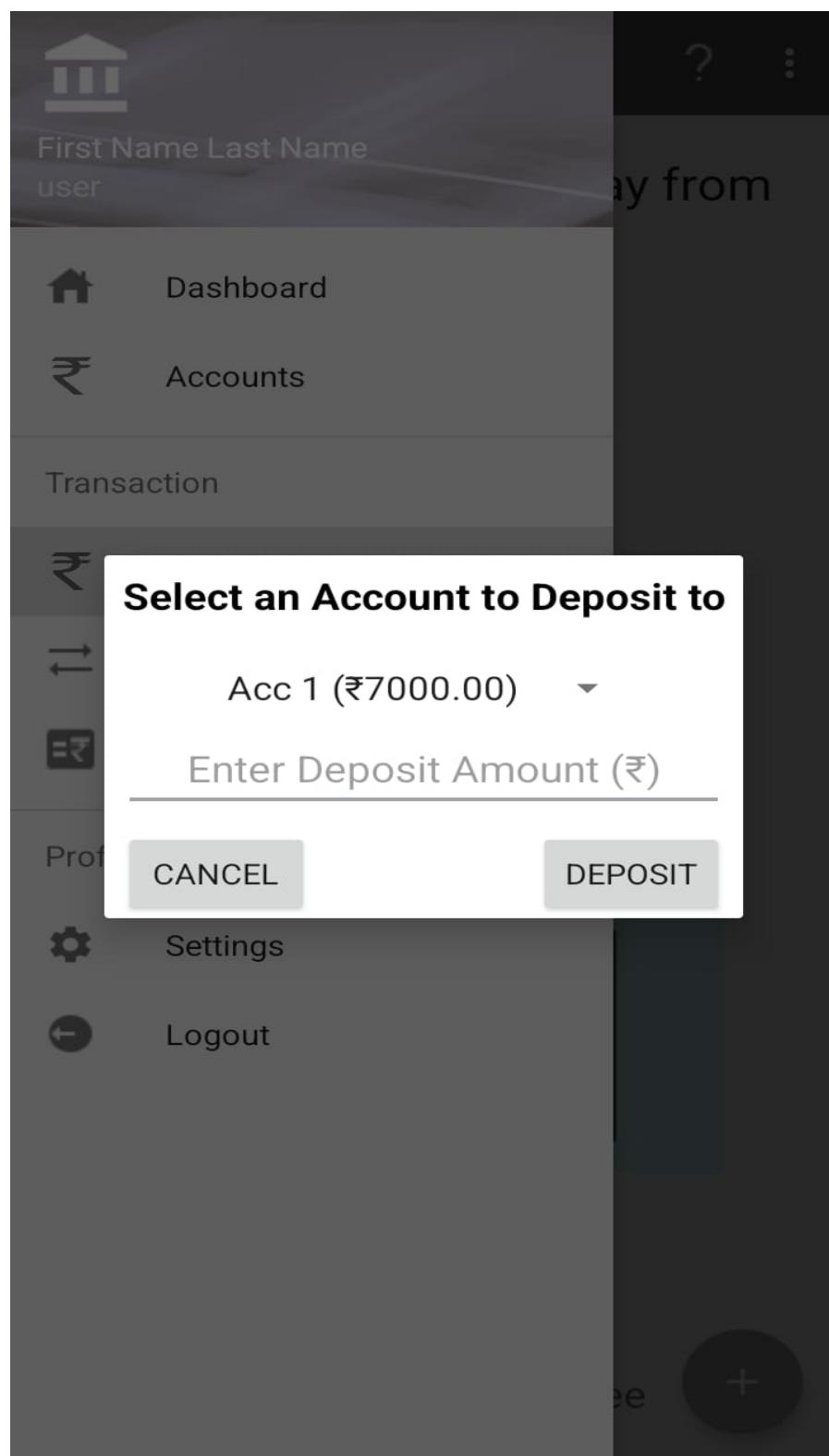
5) Transfer Page



6) Payee Page



7) Deposit Page



8) Transaction Page

Transactions		?	:
Account	Type	Order by	
Acc 1 (..)	All	Old - N..	▼
Account: Acc 1 (A1) Balance: ₹7000.00			
	DEPOSIT - T1-D1 2023/12/07 - 06:55 pm Amount: ₹10000.00		
	TRANSFER - T2-T1 2023/12/07 - 07:34 pm From: Acc 1 (A1) - To: Acc 2 (A2) Amount: ₹1000.00		
	PAYMENT - T3-P1 2023/12/07 - 07:51 pm To Payee: Payee1 (P1) Amount: ₹1000.00		
	PAYMENT - T4-P2 2023/12/07 - 07:52 pm To Payee: Payee 2 (P2) Amount: ₹1000.00		

CHAPTER - 5

Testing

- 5.1 Techniques and strategies
- 5.2 Cost estimation model
- 5.3 Future scope and further Enhancement of the project
- 5.4 Bibliography
- 5.5 Appendices
- 5.6 Glossary

5.1 – Techniques and strategies

Testing is a process to show the corrections of the program. Testing is needed to show completeness, to improve the quality of the software and to provide the maintenance aid. Some testing standards are therefore necessary reduce the testing costs and operation time.

Testing software extends throughout the coding phase and it represents the ultimate review of configurations, design and coding. Based on the way the software reacts to these testing.

We can decide whether the configuration that has been built is study or not. All components of an application are tested, as the failure to do so many results in a series of bugs after the software is put to use.

White Box testing: -

White Box (or glass box) testing is the process of giving input to the system and checking how the system processes input to generate output.

It refers to the testing a system with full knowledge and access to all source code and other architecture documents. This testing enables to reveal bugs and vulnerabilities quickly in comparison with trial and error method. More complete testing coverage is ensured by exactly knowing what to test.

White box testing involves thorough testing of the application. It requires knowledge of code and the test cases chosen verifies if the system is implemented as expected. It typically includes checking with the data flow, exceptions, and errors, how they are handled, comparing if the code produces the expected results.

Black Box testing: -

Black Box testing is the process of giving input to the system and checking the output of the system without bothering how the output is generated.

It refers to testing a system without knowledge of specification to the internal workings of the system, access to the source code, and knowledge of the architecture.

Essentially this approach mimics in a close approach, how an attacker typically follows approach to the application. However, the uncovering of issues or vulnerabilities could be further longer, because of lacking internal application knowledge.

Black box testing is done at an outer level of the system. Test cases merely check if the output is correct for the given input. User is not expected to the internal flow or design of the system.

Grey Box testing: -

Grey Box testing is a combination of White Box and Glass Box Testing. In this, the tester has little knowledge about the internal working of the software.

So, he tests the output as well as process carried out to generate the output.

It refers to a testing system by knowing limited information about the internals of the system. The knowledge is always limited for detailed design documents and architecture diagrams. In concise, it is a good blend of black and white box testing, which leverages the strengths of each of the testing.

Grey box testing is a combination of both black box and white box testing. This is because it involves access to the system; however, at an outer level. A little knowledge of the system is expected in Grey box testing.

Non-functional testing: -

Non-functional testing is the testing of a software application or system for its non-functional Requirements: the way a system operates, rather than specific behaviours of that system. This is contrast to functional testing, which tests against functional requirements that describe the functions of a system and its components. The names of many non-functional tests are often used interchangeably because of the overlap in scope between various non-functional requirements. For example, software performance is a broad term that includes many specific requirements like reliability and scalability.

Software testing strategies: -

Testing Involves:

- A. Unit Testing
- B. Integration Testing
- C. Acceptance testing

A. Unit Testing: -

The unit testing is purpose of unit testing is to ensure that each program is fully tested.

B. Integration Testing: -

The integration testing is individual program units or programs are integrated and tested as a complete system to ensure that the software.

C. Acceptance Testing: -

This testing involves planning and the execution of various types of test in order to demonstrate that the implemented software system satisfied the requirements. Finally, our project meets the requirements after going through all the levels of testing.

5.2 – Cost Estimation Model

Costing Objectives: -

- **To ensure viability:**
 - ❖ Feasibility study
 - ❖ Resource planning
 - ❖ Cost/benefit analysis
- **Provide input for pricing (including bidding negotiations etc.)**
- **To serve as a management tool:**
 - ❖ Cost Control and Management
 - ❖ Risk management
 - ❖ Budget planning
- **Criteria for good project costing:**
 - ❖ Accurate
 - ❖ Realistic (good procurement and engineering practice)
 - ❖ Consistent
 - ❖ Transparent
 - ❖ self-effective
 - ❖ Good Documentation.

Many looks upon project testing as a cost. While it is true that software testing does cost money, in many cases significant amounts of money, it is also an activity that an organization to avoid costly failures further on in the development process.

Most understand this relationship project testing is spending money to save money. What many do not also realize is that software testing also produces valuable assets for the organization. This article will discuss those assets of software testing.

Cost Estimation:

Working time estimation is as given,

$$2 \text{ months} + 4 \text{ days} = 64 \text{ days}$$

$$5 \text{ hr / 64 day} = 320 \text{ hours}$$

Now, the expenses & cost estimation are given below:

Computer rent	= 2 ,000 /-
---------------	-------------

+ Light Bill Rs. 7 / unit	
---------------------------	--

Worth 400 units	= 2,800 /-
-----------------	------------

+ Database design & creation	= 4,200 /-
------------------------------	------------

+ Coding& Validation	= 5000 /-
----------------------	-----------

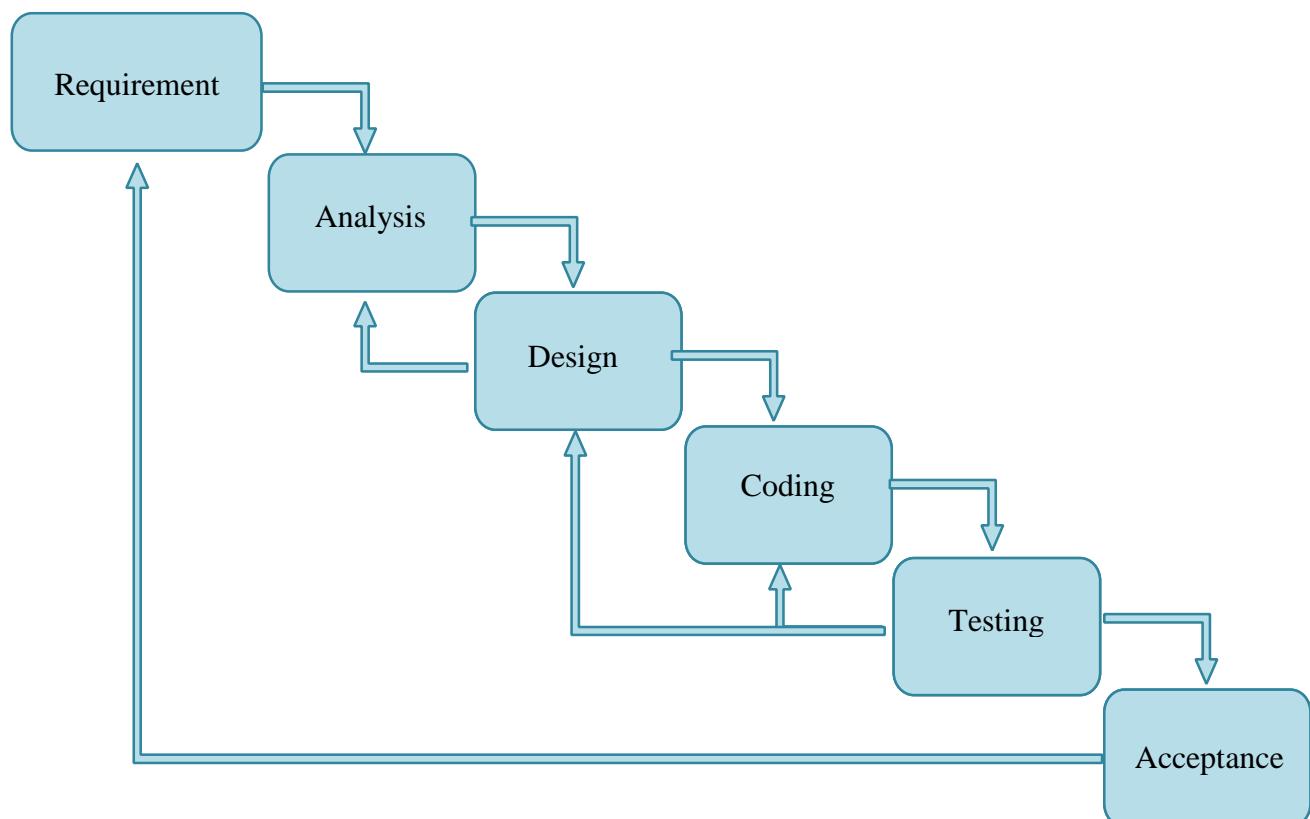
Amount	= 14,000 /-
--------	-------------

Total	= 14,000 /-
-------	-------------

SDLC Model

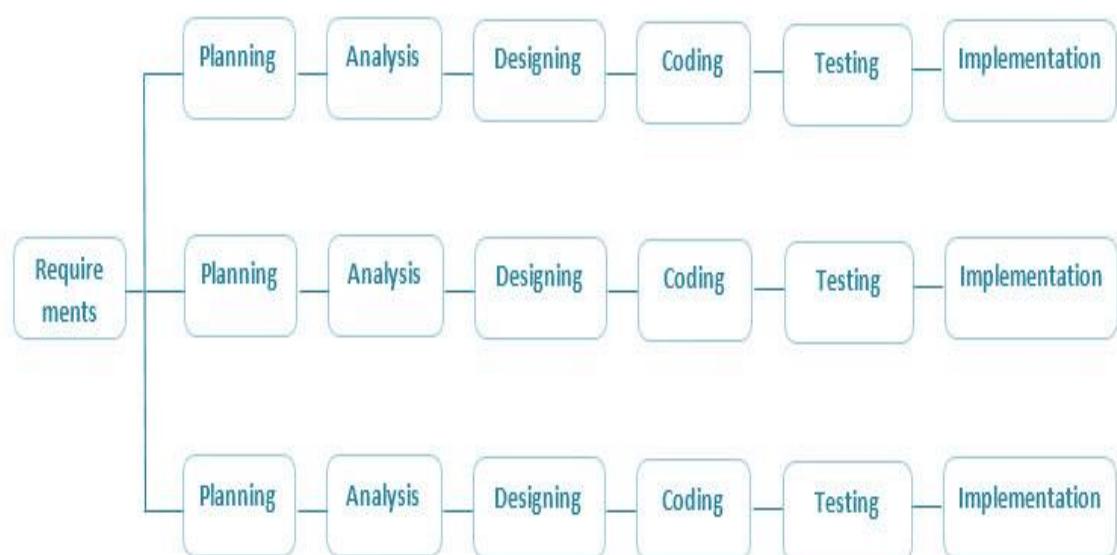
Waterfall Model: -

Waterfall approach was first SDLC Model to be used widely in Software Engineering to ensure success of the project. In "The Waterfall" approach, the whole process of software development is divided into separate phases. In Waterfall model, typically, the outcome of one phase acts as the input for the next phase sequentially. Following is a diagrammatic representation of different phases of waterfall model.



 **Iterative Model: -**

In Iterative model, iterative process starts with a simple implementation of a small set of the software requirements and iteratively enhances the evolving versions until the complete system is implemented and ready to be deployed. An iterative life cycle model does not attempt to start with a full specification of requirements. Instead, development begins by specifying and implementing just part of the software, which is then reviewed in order to identify further requirements. This process is then repeated, producing a new version of the software at the end of each iteration of the model.

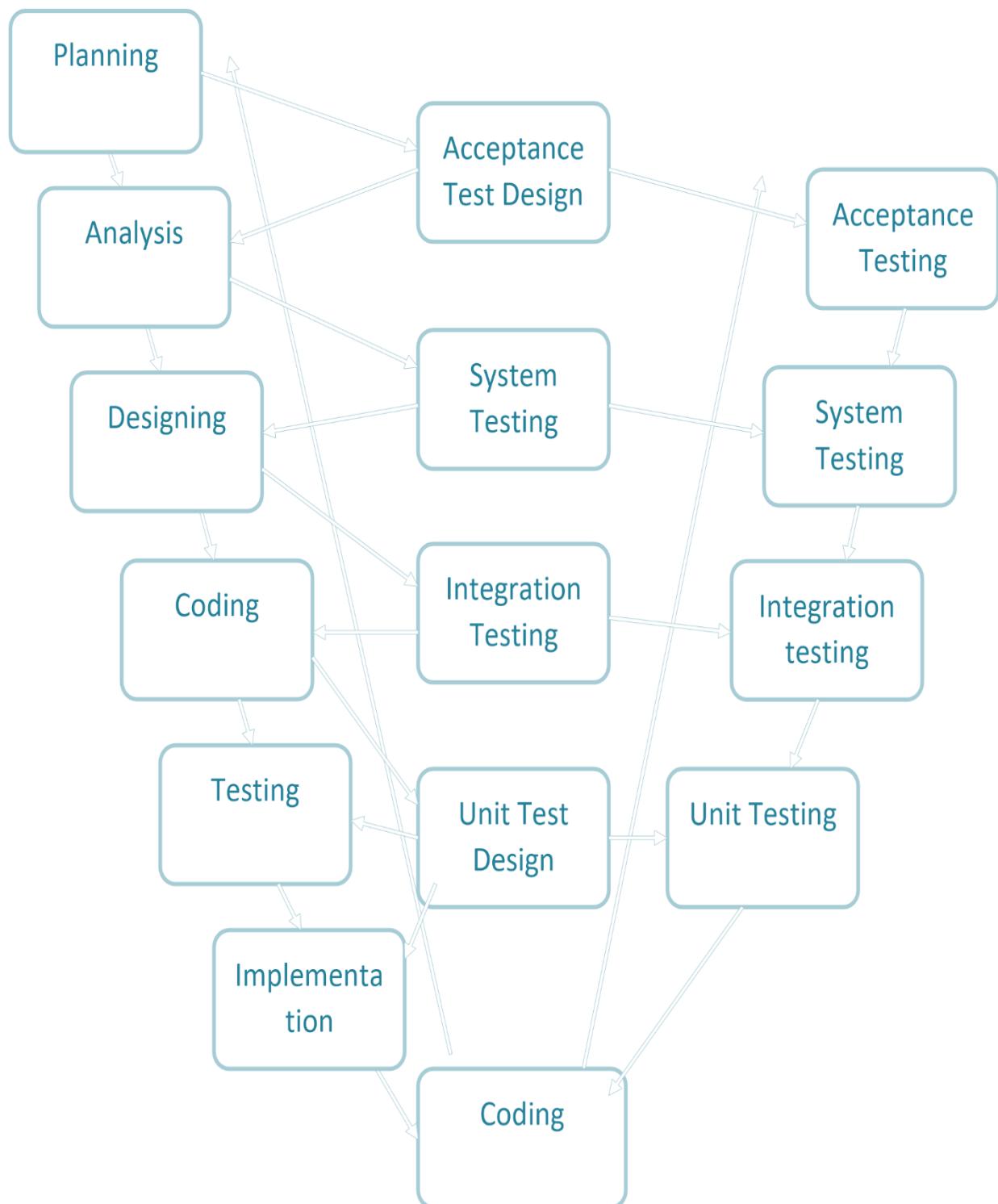


 **V Model:**

The V - model is SDLC model where execution of processes happens in a sequential manner in V-shape. It is also known as Verification and Validation model. V - Model is an extension of the waterfall model and is based on association of a testing phase for each corresponding development stage. This means that for every single phase in the development cycle there is a directly associated testing phase. This is a highly disciplined model and next phase starts only after completion of the previous phase.

 **V Model Design:**

Under V-Model, the corresponding testing phase of the development phase is planned in parallel. So there are Verification phases on one side of the V. and Validation phases on the other side. Coding phase joins the two sides of the V-Model. The below figure illustrates the different phases in V-Model of SDLC.



 **Spiral Model: -**

The spiral model combines the idea of iterative development with the systematic, controlled aspects of the waterfall model. Spiral model is a combination of iterative development process model and sequential linear development model i.e. waterfall model with very high emphasis on risk analysis. It allows for incremental releases of the product, or incremental refinement through each iteration around the spiral.

 **Spiral Model Design: -**

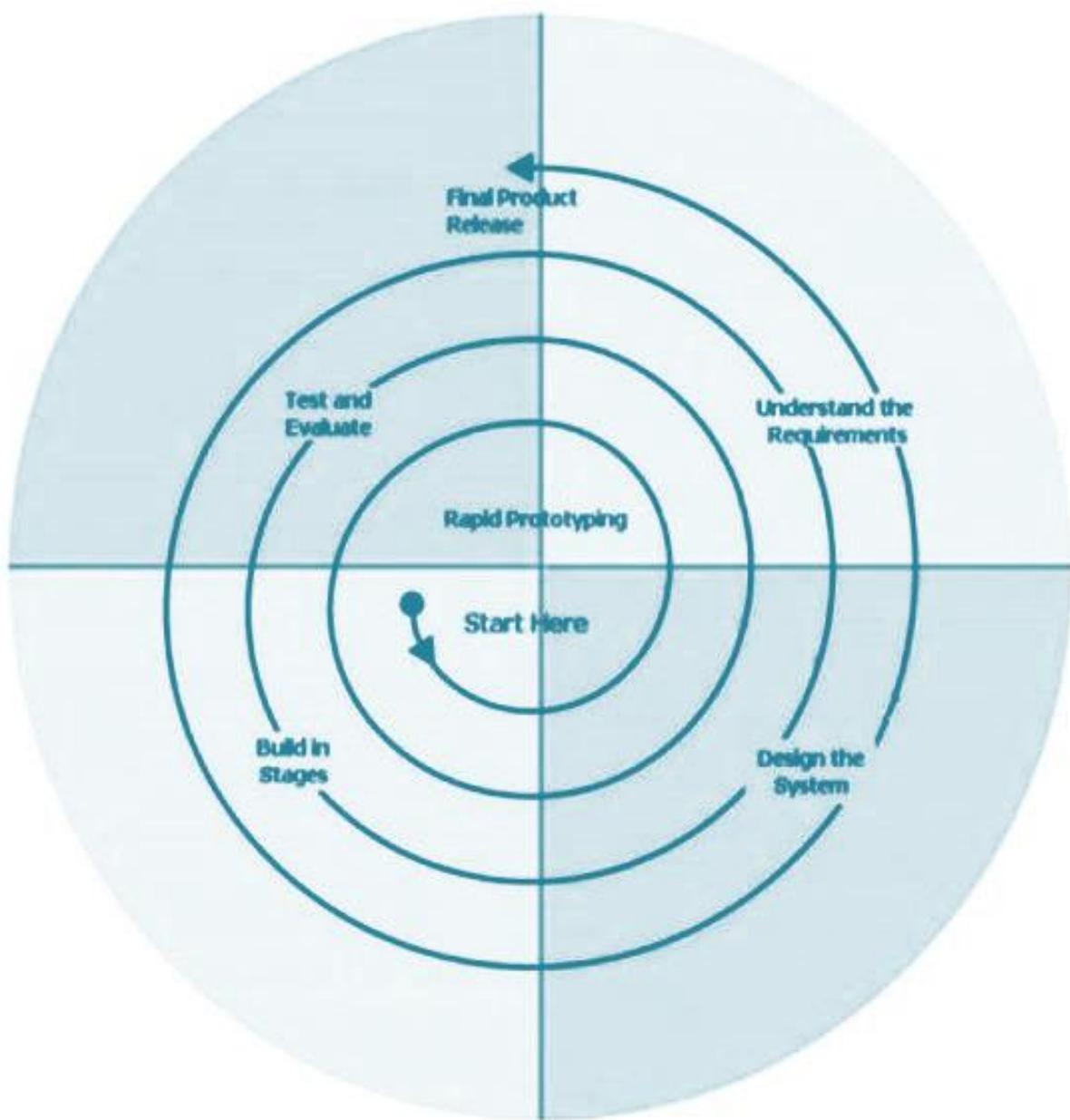
The spiral model has four phases. A software project repeatedly passes through these phases in iterations called Spirals.

Identification:

This phase starts with gathering the business requirements in the baseline spiral. In the subsequent spirals as the product matures, identification of system requirements, subsystem requirements and unit requirements are all done in. This phase.

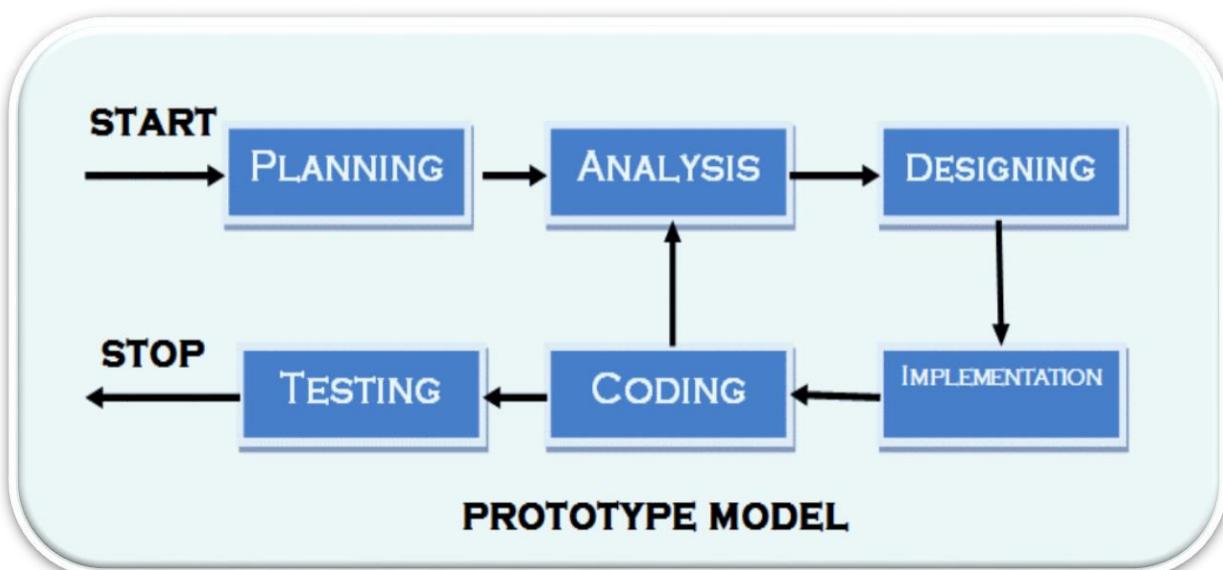
This also includes understanding the system requirements by continuous communication between the customer and the system analyst. At the end of the spiral the product is deployed in the identified market.

➤ Spiral Model Design: -



 **Prototype Model: -**

The basic idea here is that instead of freezing the requirements before a design or coding can proceed, a throwaway prototype is built to understand the requirements. This prototype is developed based on the currently known requirements. By using this prototype, the client can get an “actual feel” of the system, since the interactions with prototype can enable the client to better understand the requirements of the desired system. Prototyping is an attractive idea for complicated and large systems for which there is no manual process or existing system to help determining the requirements. The prototype is usually not complete systems and many of the details are not built in the prototype. The goal is to provide a system with overall functionality.



5.3 - Future Enhancement

A good project is one which never stops developing according to the changing situations and technologies there is a lot of scope of future enhancements.

- ✓ Make it fully dynamic app.
- ✓ Add other bank account details.
- ✓ Client can search any transaction from history.
- ✓ Update their profile.
- ✓ Change password option for client.
- ✓ Add confirmation box for transfer one to other account & payment to payee.
- ✓ Add pdf feature for transaction history.

5.4 - Bibliography

For the successful working of my project I have referred many sources for the code snippets, logic and tips and tricks from the various books as well as web sites. Most I searched for the required possessions on the google.com search engine.

Web Links:

- W3school
- Geeksforgeeks

5.5 - Appendices

- The project Bank is the customize working on Financial Information Provider.
- The application gives to all requirements points and helps to the clients for any transactions.
- We have successfully designed, coded and implemented our project with a lot of Hard work.
- Finally, I would like to thank our project guide **Mr. Uday Shah** helpful guidelines for our project. Also given suggestion when difficulties raised in our project.

5.6 - Glossary

Full Form:-

D.F.D.	Data Flow Diagram
S.R.S.	Software Requirement Specification
S.D.L.C	Software Development Life Cycle
E.R	Entity Relationship
XML	Extensible Mark-up Language.

