



GUJARAT TECHNOLOGICAL UNIVERSITY

Chandkheda, Ahmedabad
Affiliated

G H Patel College of Engineering and Technology

Department of Computer Engineering (07)

A.Y. 2016-17(EVEN), SEMESTER 6

SUBJECT CODE: 2160701

SUBJECT NAME: SOFTWARE ENGINEERING (SE)

Software Requirements Specification

Bharat Interface for Money (BHIM) Mobile App

-: **Made by:** -

Jemish Paghadar (140110107030)

Deep Patel (140110107037)

-: **Guided by:** -

Prof. Khyati Mehta

-: INDEX: -

1.0 Introduction

- 1.1. Purpose
- 1.2. Scope of Project
- 1.3. Glossary
- 1.4. References
- 1.5. Overview of Document

2.0 Overall Description

- 2.1 System Environment
- 2.2 Functional Requirements Specification
 - 2.2.1 Send money on BHIM
 - 2.2.3 Collect money on BHIM
 - 2.2.3 Scan & Pay
 - 2.2.4 My Profile
 - 2.2.5 Reset/change UPI pin on BHIM
- 2.3 User Characteristics
- 2.4 Non-functional requirements

3.0 Processing Model: The Waterfall Model

- 3.1 Description:
- 3.2 Phases of waterfall model
 - 3.2.1 Communication
 - 3.2.2 Planning
 - 3.2.3 Modeling
 - 3.2.4 Construction
 - 3.2.5 Deployment

4.0 Various Diagrams

- 4.1 Data Flow Diagram(DFD)
 - 4.1.1 Level 0 DFD
 - 4.1.2 Level 1 DFD
 - 4.1.3 Level 2 DFD
- 4.2 Sequence Diagram

5.0 Data Dictionary

6.0 Unit Testing

- 6.1 Black-Box Testing
- 6.2 White-Box Testing

1.0 Introduction

1.1 Purpose

The purpose of this document is to present a detailed description of the BHIM Application. It will explain the purpose and features of the system, the interfaces of the system, what the system will do, the constraints under which it must operate and how the system will react to external stimuli. This document is intended for both the stakeholders and the developers of the system.

1.2 Scope of Project

This Application system will be a BHIM app for a citizen of the India. This Application will be designed to maximize the interaction between citizens and Banks. By maximizing transparency between both sides, the Application will meet public's needs while remaining easy to understand and use. More specifically, this system is designed to Time bound delivery of essential services to the citizens of the India. The Application will facilitate communication between general public and Banks via Application interface. Preformatted reply forms are used in complaint page through the system to provide a uniform review process. The system also contains a relational database containing a list of various users.

1.3 Glossary

<u>Term</u>	<u>Definition</u>
Dashboard	Dashboards often provide at-a-glance views of user activity relevant to a particular Transaction or log-activity.
Activity	Money Transaction via mobile , Check Balance , Add multiple Bank account etc.
User	Citizens of India
User's Database	The existing member's Transactions database.
Reader	Anyone visiting the site who are authorized bank account linked with mobile number.
Review	user can see their Money transaction or pending Transaction.
Software Requirements Specification	A document that completely describes all of the functions of a proposed Application and the constraints under which it must operate. For example, this document.

1.4 References

- IEEE. IEEE STD 830-1998 IEEE Recommended Practice for Software Requirements Specifications. IEEE Computer Society, 1998.
- https://d2pv62lkmtdxww.cloudfront.net/npcipay/faq/faq/en_US/index.html
- <https://en.wikipedia.org/wiki/BHIM#Benefits>
- <http://indianexpress.com/article/india/bhim-app-narendra-modi-digi-dhan-mela-demonetisation-4452004>
- https://d2pv62lkmtdxww.cloudfront.net/npcipay/faq/faq/en_US/index.html
- <https://play.google.com/store/apps/details?id=in.org.npci.upiapp&hl=en>

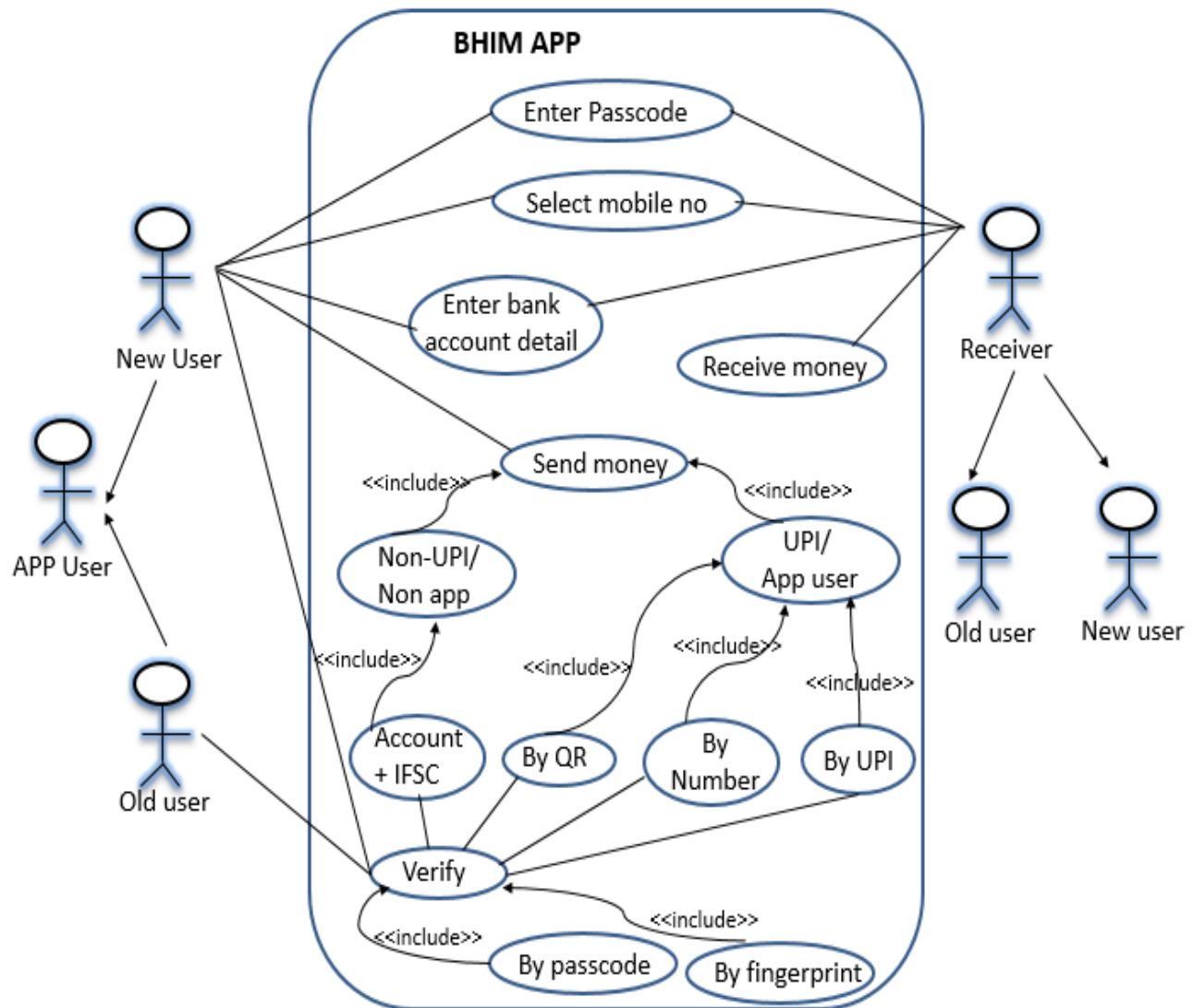
1.5. Overview of Document

In the next, the Overall Description section, of this document gives an overview of the functionality of the product. It describes the informal requirements and is used to establish a context for the technical requirements specification. Requirements Specification section, of this document is written primarily for the developers and describes in technical terms the details of the functionality of the product. The document describes the same software product in its entirety.

The information captured using this document is intended to provide us with information about your specification and the requirements for BHIM Application. To enable us to gain as thorough an understanding of what you require this document contains questions about your overall development along with detailed questions regarding what you want from BHIM and the information (content) that is to be displayed on it.

2.0 Overall Description

2.1 System Environment



Bharat Interface for Money (BHIM) is an app that lets you make simple, easy and quick payment transactions using Unified Payments Interface (UPI). You can make instant bank-to-bank payments and Pay and collect money using just Mobile number or Virtual Payment Address (VPA). To enable transfers directly using your bank account, your bank needs to be live on UPI (Unified Payment Interface) platform. All the banks, which are currently live on UPI, have been listed in the BHIM app

The division of BHIM app mainly into nine component parts -:

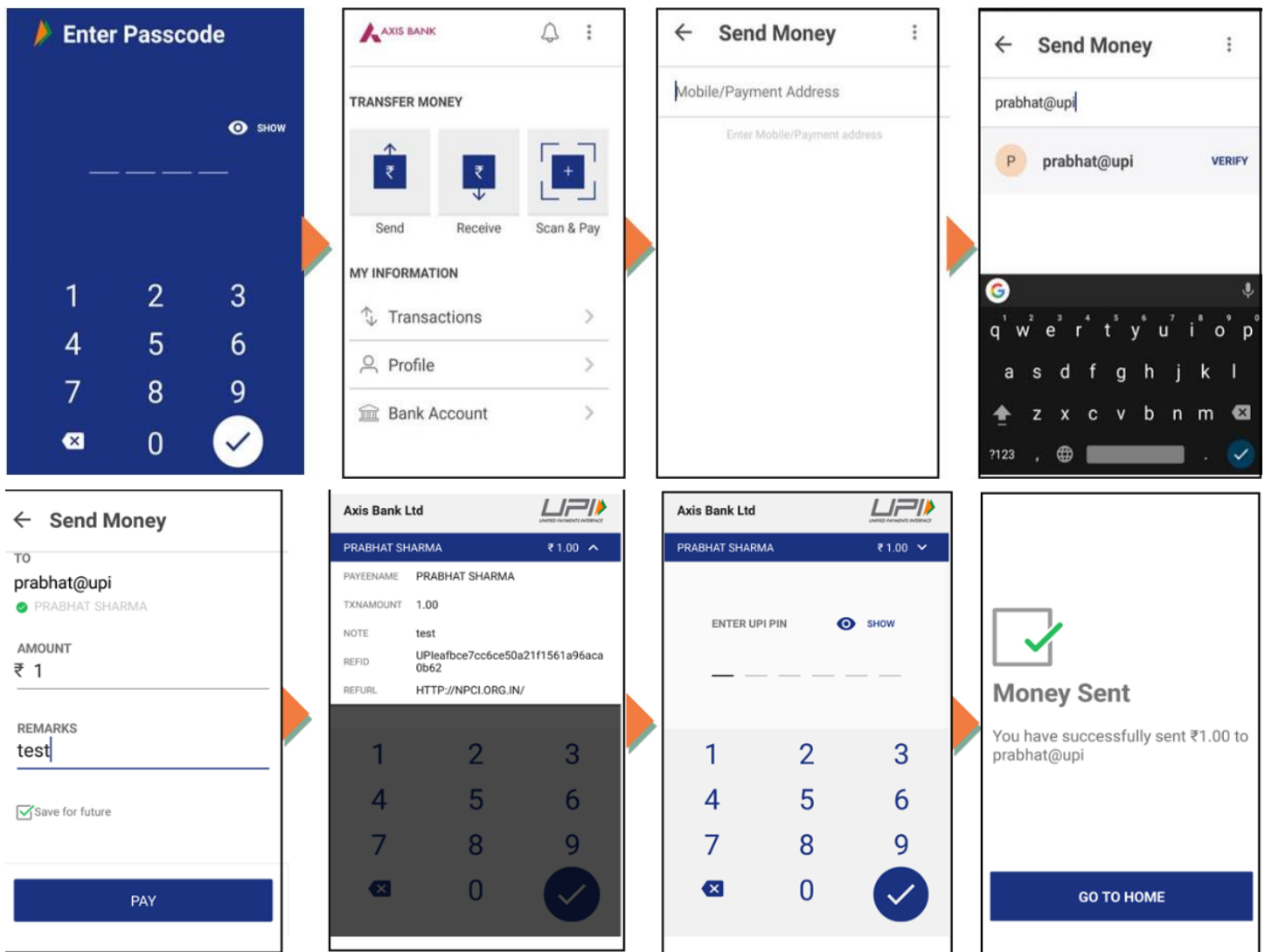
- 1) Send Money
- 2) Request Money
- 3) Scan & Pay
- 4) Transactions
- 5) Profile
- 6) Bank Account
- 7) Language
- 8) Block User
- 9) Privacy

2.2 Functional Requirements Specification

❖ Send Money on BHIM

➤ Using Virtual Payment Address (VPA):

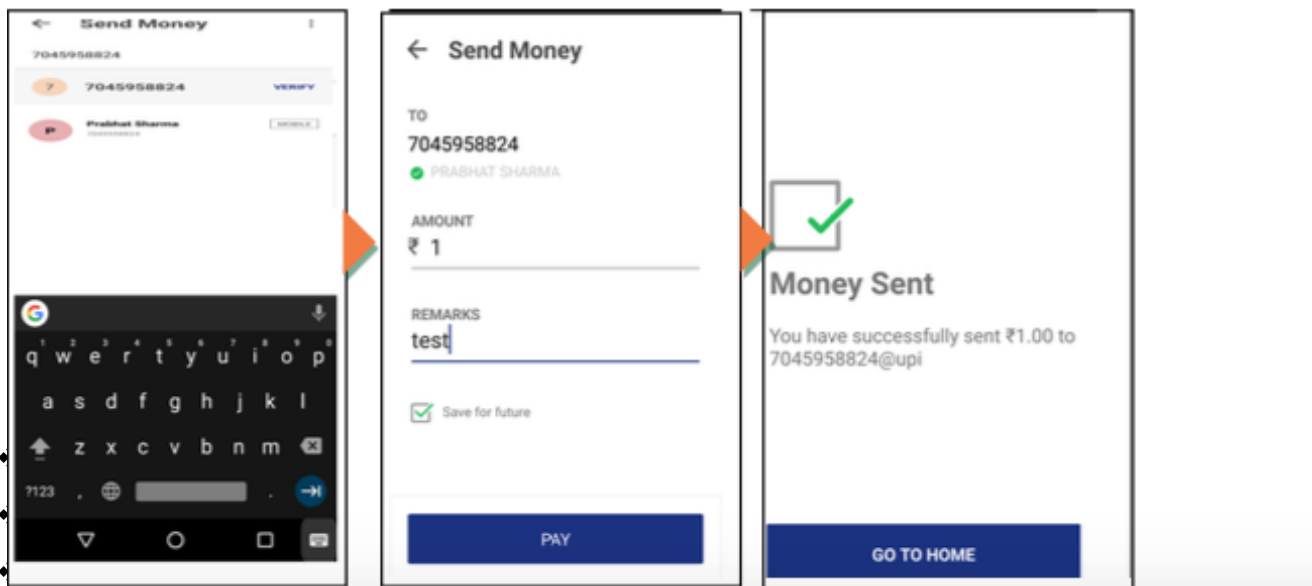
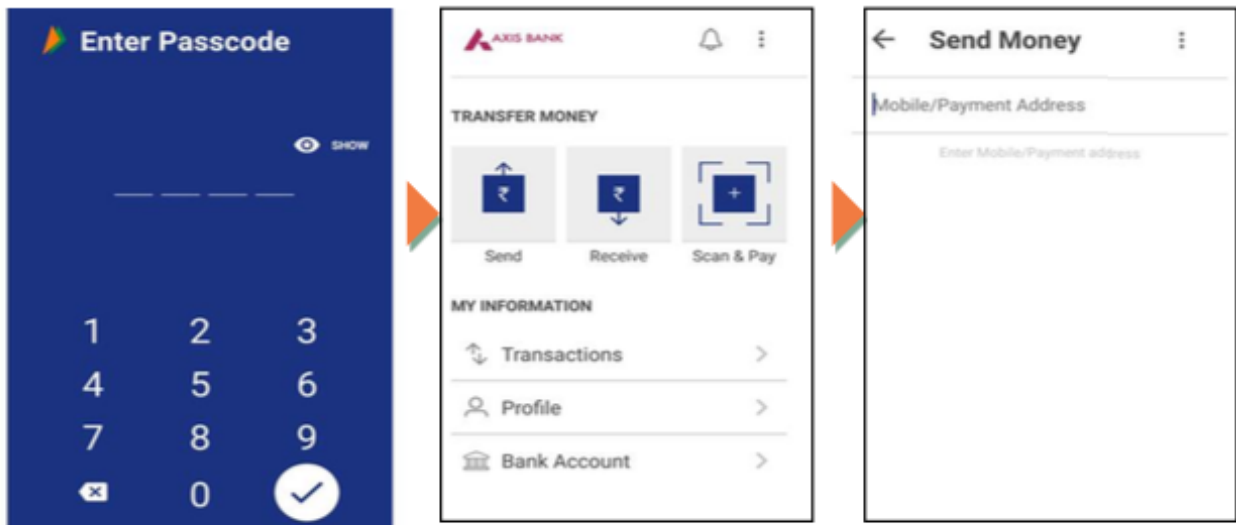
1. Customer opens the app on his/her smartphone and enters the app login passcode.
2. Select Send Money Option.
3. Customer initiates a SEND request
4. Customer enters the virtual payment address (VPA) & clicks on verify to check the name of the payee.
5. Name of the PAYEE is fetched from the Bank CBS. Customer enters the amount and remarks for the transaction and Clicks on PAY.
6. Customer can also check the details of the transaction from the dropdown in the UPI PIN entry page
7. UPI PIN entry page opens where customer enters his UPI PIN
8. Confirmation of money sent is shown to the customer
9. Customer gets a notification for the success of the transaction from the app



➤ Using Mobile Number:

1. Customer opens the app on his/her smartphone and enters the app login passcode.
2. Select Send Money Option.
3. Customer initiates a SEND request
4. Customer enters the mobile number & clicks on verify to check the name of the payee.
5. Name of the PAYEE is fetched, if user is registered on *99#/BHIM. Customer enters the amount and remarks for the transaction and Clicks on PAY.

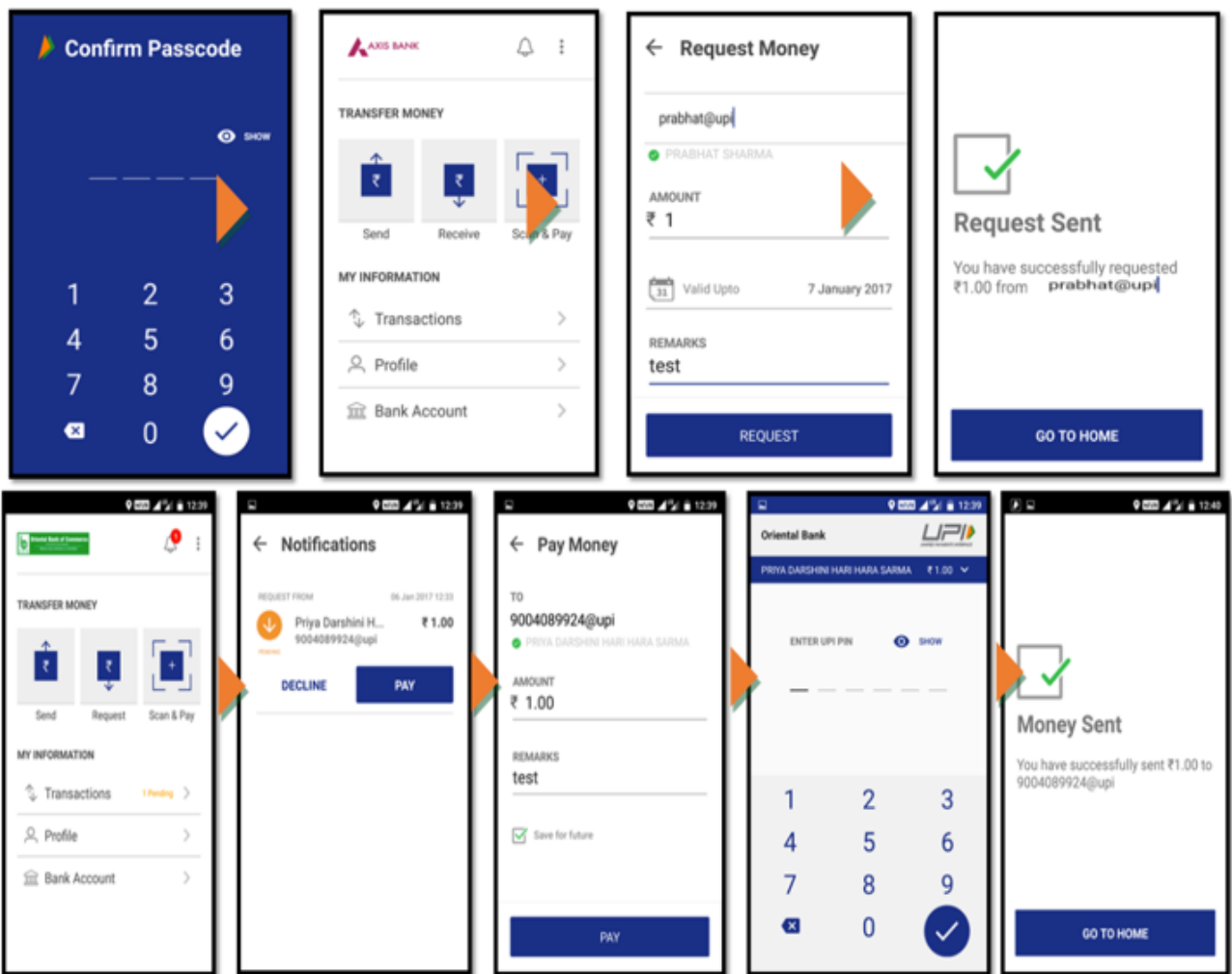
6. Customer can also check the details of the transaction from the dropdown in the UPI PIN entry page
7. UPI PIN entry page opens where customer enters his UPI PIN
8. Confirmation of money sent is shown to the customer
9. Customer gets a notification for the success of the transaction from the app.



❖ Collect money on BHIM

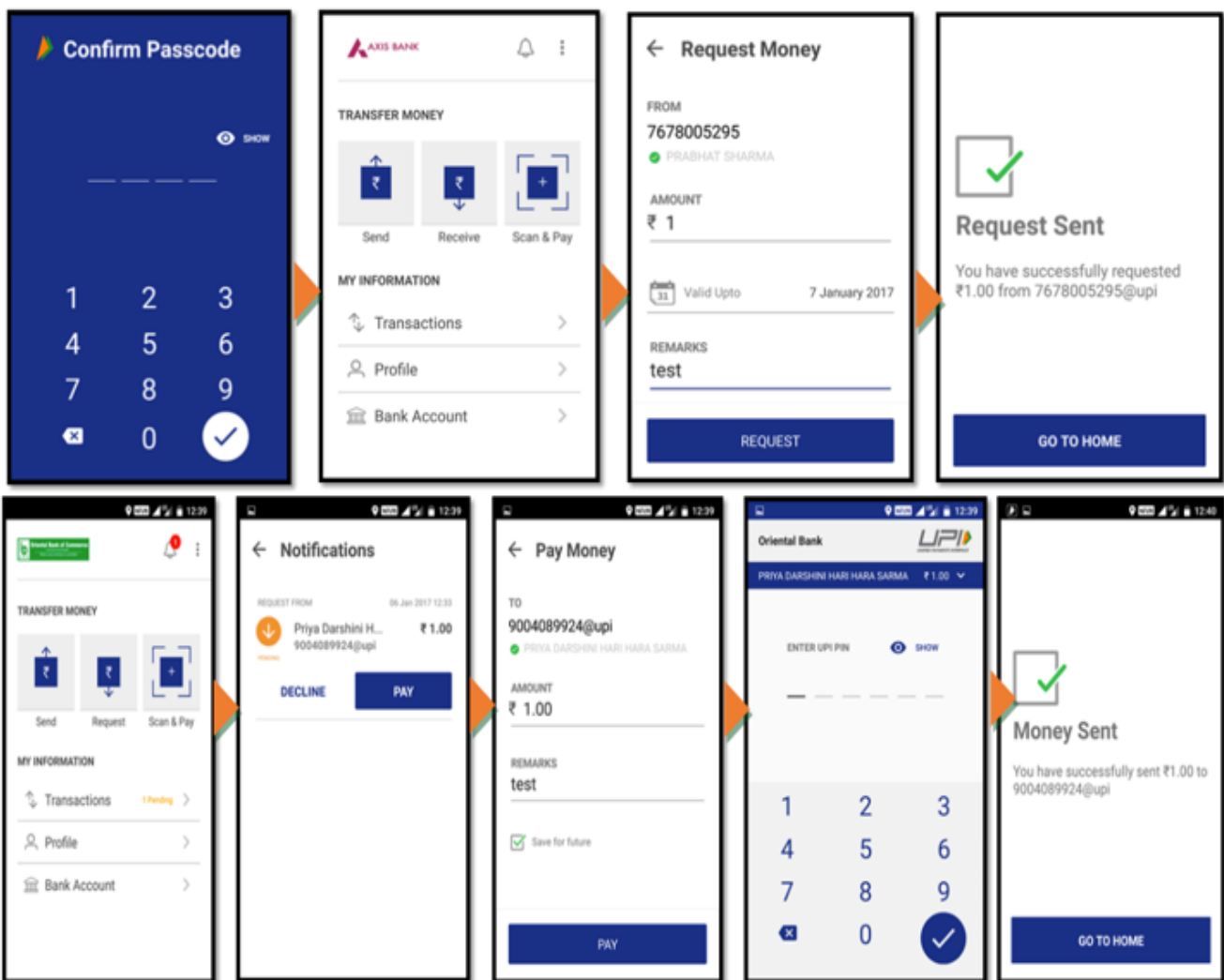
➤ Using Virtual Payment Address (VPA):

1. Customer opens the app and enters the app login and passcode. Select Receive Money Option and initiates a request transaction.
2. Customer enters virtual payment address (VPA) to collect money and clicks on verify to check the name of payer
3. PAYER name is shown to the requester and he enters the amount and remarks
4. Confirmation of Request Sent is given to the initiator.
5. Initiator is notified once PAYER accepts the request for money



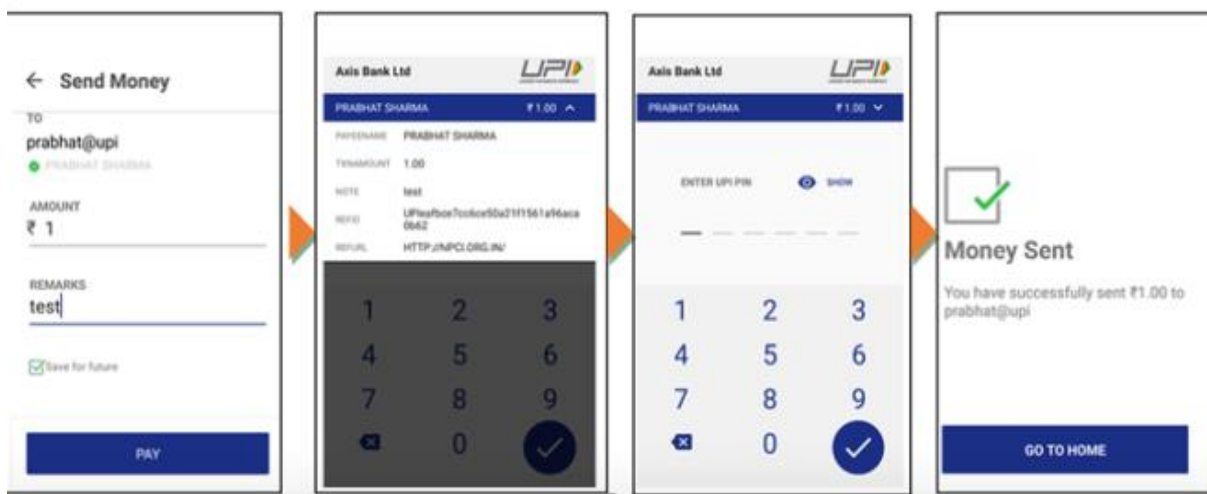
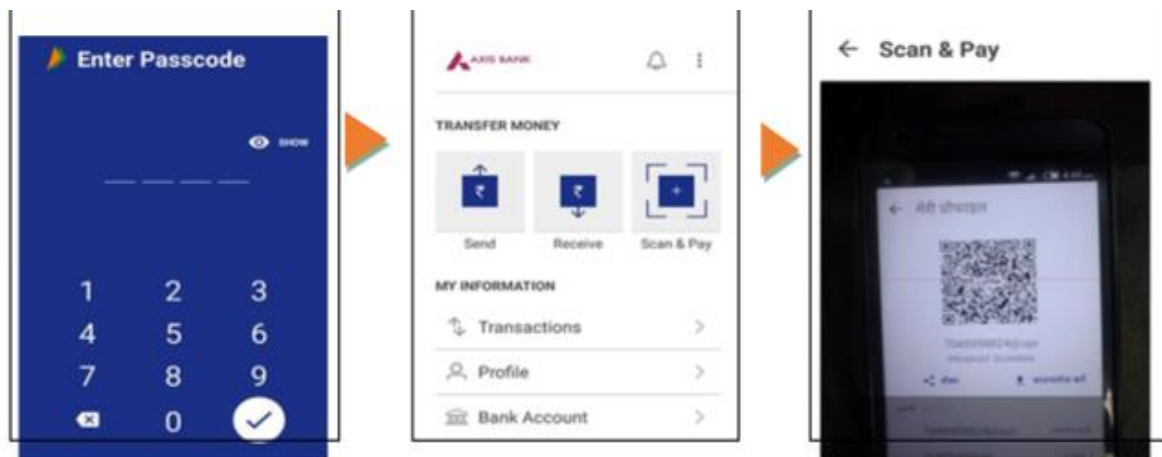
➤ Using Mobile Number:

1. Customer opens the app and enters the app login and passcode. Select Receive Money Option and initiates a request transaction.
2. Customer enters Mobile no to collect money and clicks on verify to check the name of payer
3. PAYER name is shown to the requester and he enters the amount and remarks
4. Confirmation of Request Sent is given to the initiator.
5. Initiator is notified once PAYER accepts the request for money

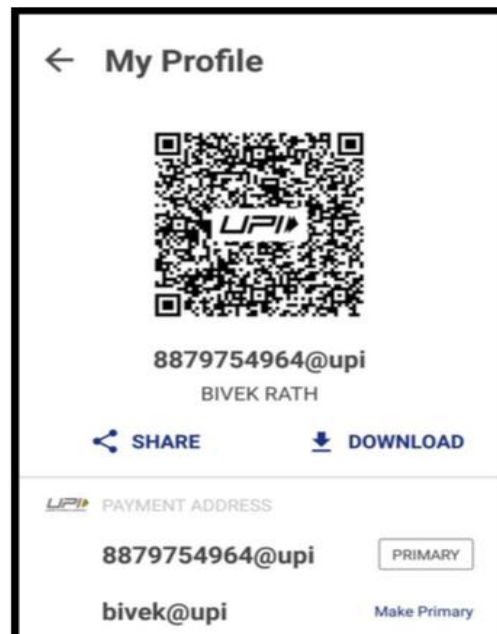


❖ Scan & Pay

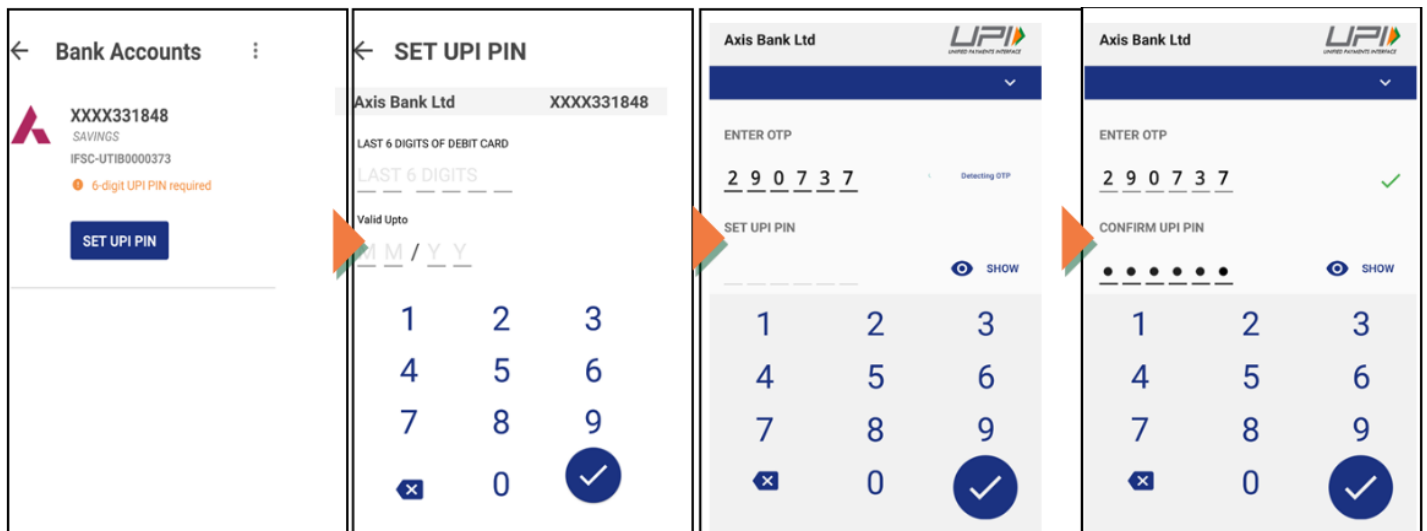
1. Customer opens the app on his/her smartphone and enters the app login passcode.
2. Customer can also choose to Scan & Pay through a QR.
3. Application opens a QR scanner which then populates the details like any other PAY transaction and on entering the UPI PIN, transaction can be completed
4. Name of the PAYEE is fetched from the Bank CBS. Customer enters the amount and remarks for the transaction and Clicks on PAY.
5. UPI PIN entry page opens where customer enters his UPI PIN
6. Confirmation of money sent is shown to the customer.
7. Customer gets a notification for the success of the transaction from the app



❖ My Profile



❖ Reset/Change UPI PIN on BHIM:



2.3 User Characteristics

As fast as it can get! All payments over BHIM are linked to your bank account and transaction can be completed within few seconds. There are no charges for making transaction through BHIM. Note: Your bank might however levy a nominal charge as UPI or IMPS transfer fee, which is not under our control.

To enable transfers directly using your bank account, your bank needs to be live on UPI (Unified Payment Interface) platform. All the banks, which are currently live on UPI, have been listed in the BHIM app.

You can set your UPI PIN by going to Main Menu -Bank Accounts -Set UPI-PIN for the selected account. You will be prompted to enter the last 6 digits of your Debit/ATM card along with the expiry date. You will then receive an OTP which you will enter and set your UPI-PIN. Note: UPI PIN is not the same as MPIN provided by your bank for mobile banking.

Currently, BHIM supports linking of one Bank only. At the time of account set-up, you can link your preferred bank account as the default account. In case you want to link another bank account, you can go to Main menu, choose Bank Accounts and select your default account. Any money that is transferred to you using your mobile number or payment address will be credited into your default account.

At the time of registration, you will be providing us with the Debit card details and with the use your mobile number registered to your bank account, we will pre-fetch the details from your bank. All the information exchange happens over secure banking networks and we don't store it, your information is safe! Every transaction through your bank account will be recorded by the Bank and you can view all your past transactions in the 'Transaction History' section on the BHIM App.

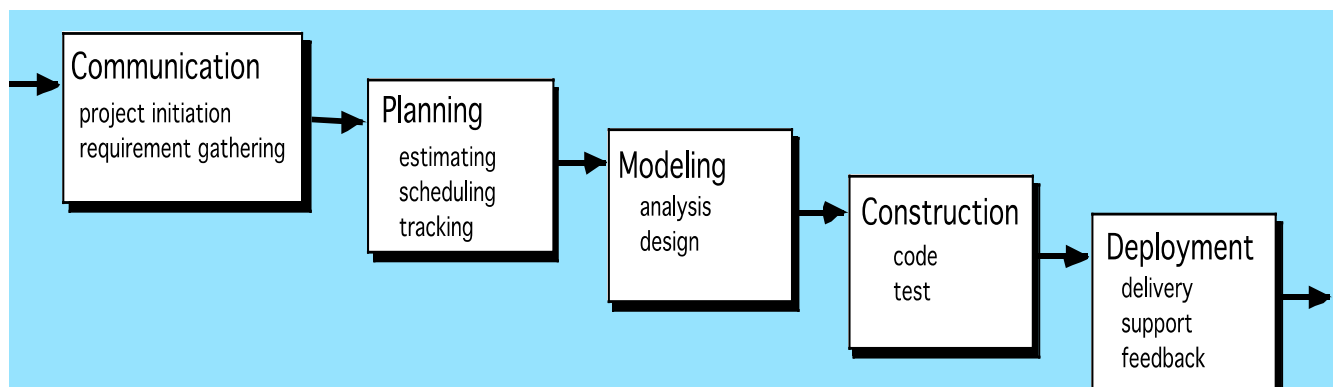
2.4 Non-functional requirements

1. Secure access of confidential data (user's details).
2. 24 X 7 availability.
3. Flexible service based architecture will be highly desirable for future extension
4. Simplicity of interface.
5. Usability
6. Interportability
7. Availability
8. Reliability
 - a. backup & recovery
 - b. Data migration
 - c. maintenance
9. Security

3.0 Processing Model

The Waterfall Model

Diagram:



Description:

The Waterfall model is a sequential design process, used in software development process, in which process is seen as flowing steadily downwards through the phases of conception , initiation , analysis , design , construction , testing, production and maintenance.

It was first process model to be introduced and referred as a linear-sequential life cycle model. It is very simple and understand to use.

Waterfall approach was the first SDLC model to be used widely in software engineering to ensure success of the project. In waterfall model, the whole process of software development is divided into separate phases.

Phases of waterfall model:

1. Communication:

All possible requirements of the system to be developed are captured in this phase and documented in a requirement specification doc. Here in this phase developer needs to communicate with the user to ask their specification and requirements.

Here some requirement specification of the BHIM app:

- Secure access of confidential data (user's details).
- Flexible service based architecture will be highly desirable for future extension
- Simplicity of interface.

2. Planning:

The requirement specification from first phase are studied in this phase and system design is prepared.

Planning of the BHIM app:

- To create a user-friendly online interface for citizens for transaction and, reduce the distance and time barrier between citizens and banks.
- To create an online platform where people can send or request money via mobile phone direct.
- To encourage the citizens to actively participate in BHIM app to cashless money transaction.

3. Modeling:

There are some issues, which come up in the client environment. To fix those issues we need to analysis the modules of our product. If some module cannot satisfy customers satisfaction, then again those modules are send to planning phase and delivered again with some extra features.

Modeling in the BHIM app:

If user has some question, regarding to some issue on the app then one extra module added called FAQs. Here user can give satisfied answer of the question. So to add this module first we analysis the requirement of the user.

4. Construction:

With inputs from system design, the system is first developed in small program called units, which are integrated in the next phase. Each unit is developed and tested for its functionality, which is referred to as unit testing.

Construction in the BHIM app:

- Users should be able to create new account, login to their existing accounts.
- Authenticated users should be able to send or request money using mobile phone directly via UPI pin.
- Authenticated users should be able to check bank balance.

5. Deployment:

Once the functional and non-functional testing is done, the product is deployed in the customer environment and giving feedback called Deployment.

Product should support in customer environment. Maintenance is done to deliver these changes in the customer environment.

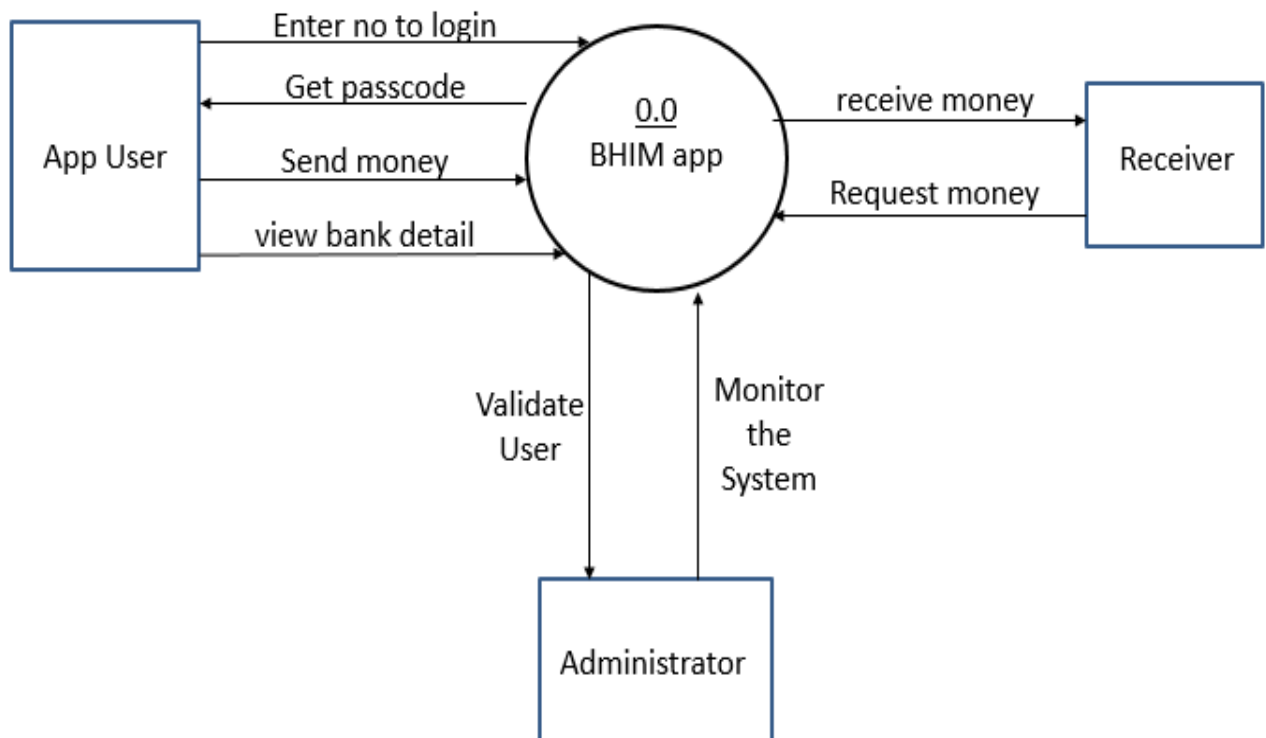
Deployment in the BHIM app:

After the product is deployed to, the Gov. it is necessary that public and various sector of the Gov. are uses it and give feedback. The Maintenance is also necessary by managing database of the various users.

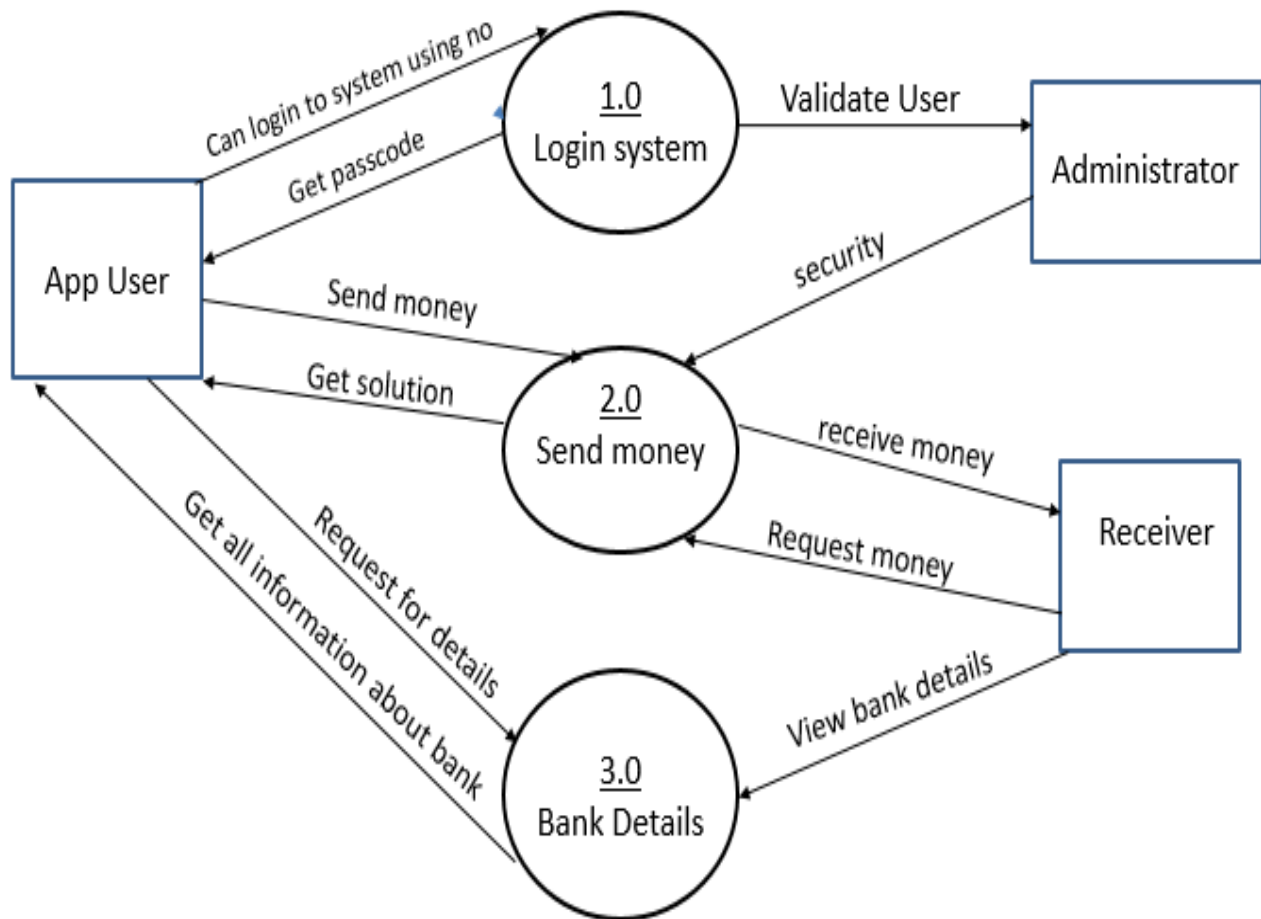
4.0 Various Diagrams

❖ Data Flow Diagram(DFD)

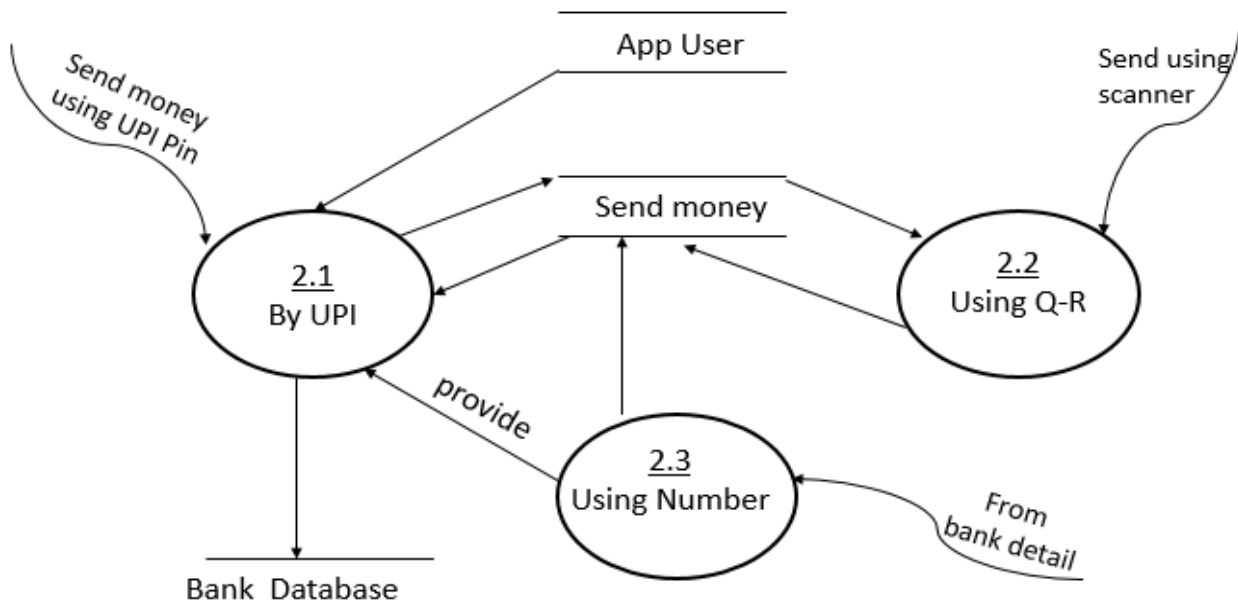
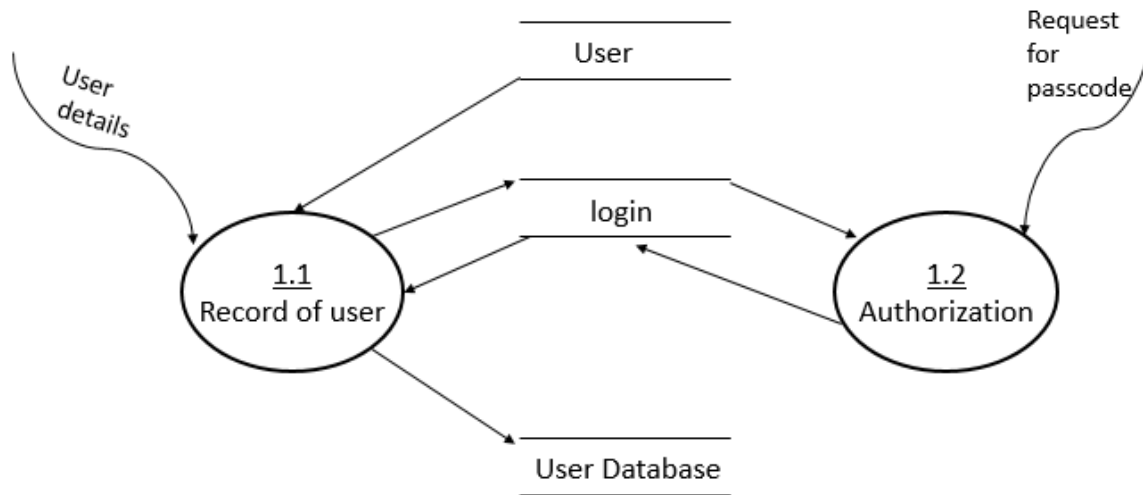
➤ Level 0 DFD

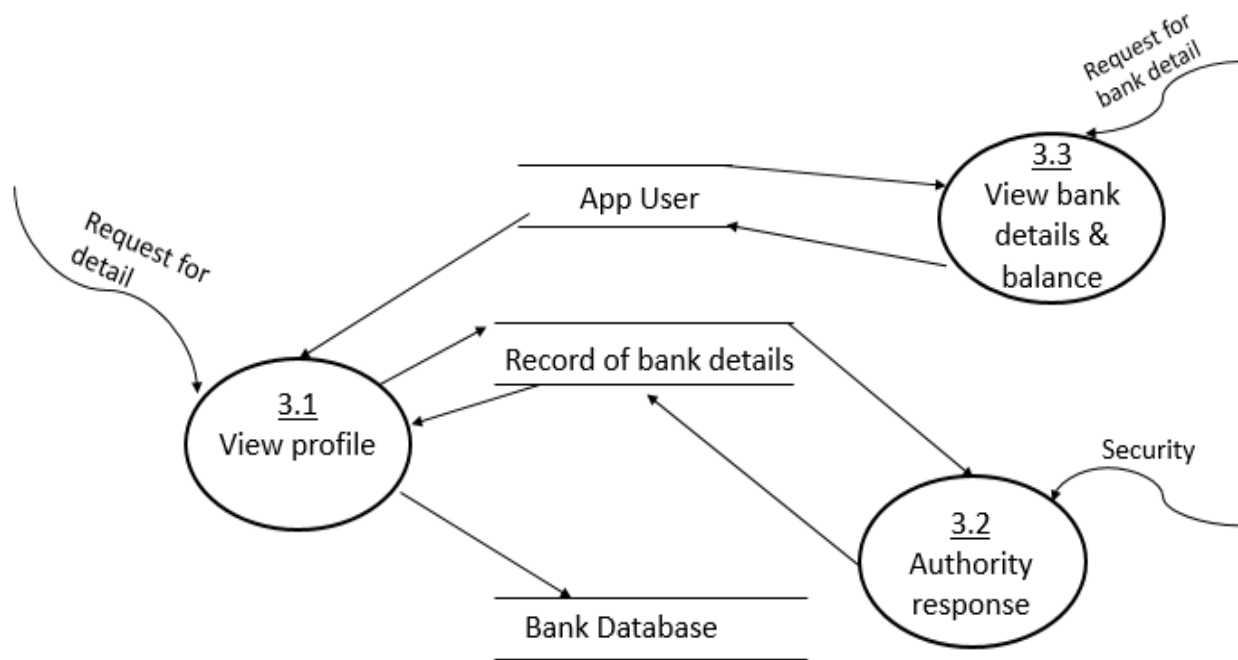


➤ Level 1 DFD

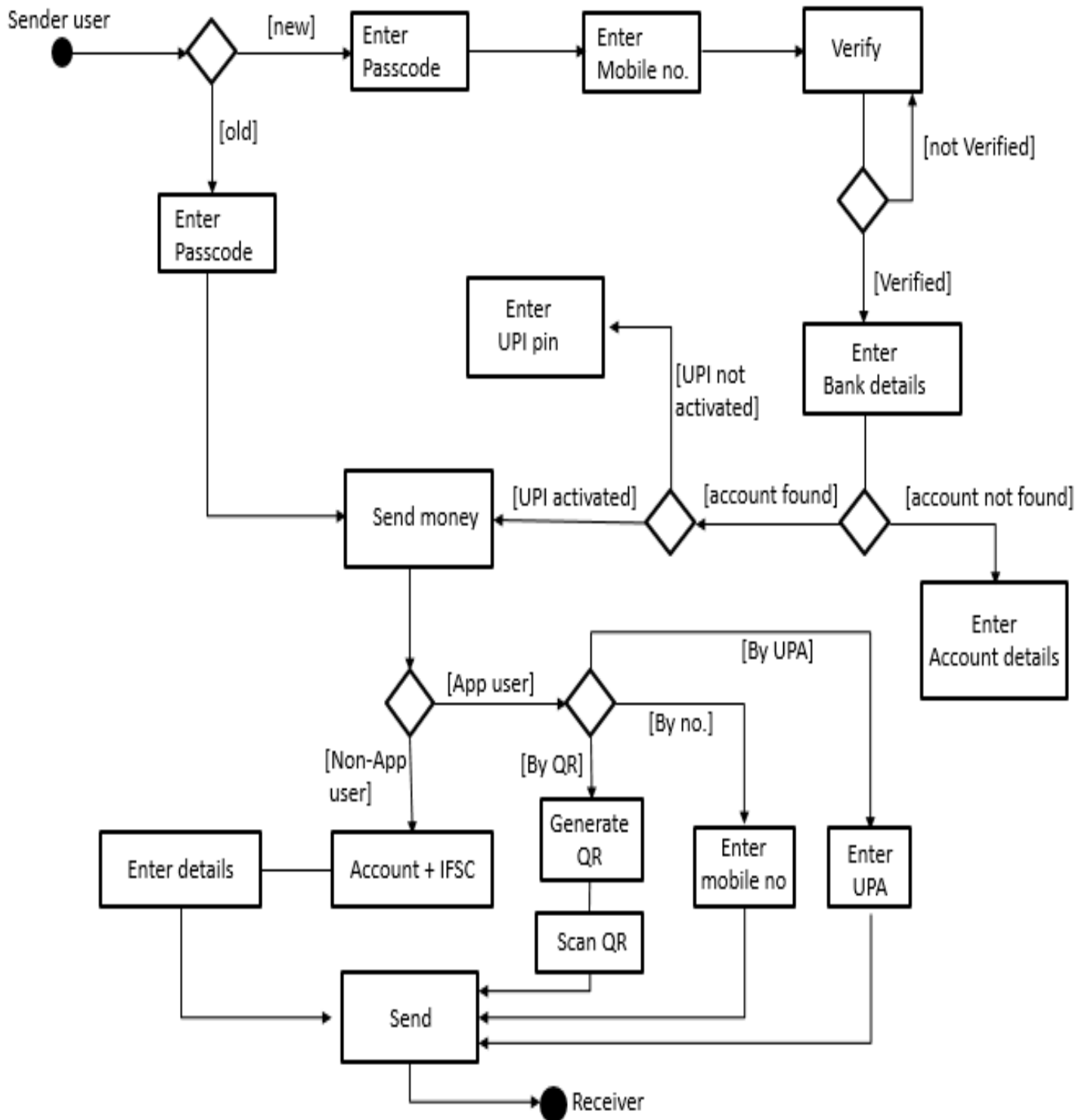


➤ Level 2 DFD





❖ Sequence Diagram



5.0 Data Dictionary

➤ Table list

- User
- User_bank
- Bank
- UPI

➤ User

Entity	Constraint	Data type(Size)	Description
u_name	NOT NULL	Varchar(50)	User name
username	PRIMARY KEY	Varchar(50)	unique id
u_password	NOT NULL	Varchar(50)	User password
u_phn	NOT NULL	Number(10)	User contact

➤ User_bank

Entity	Constraint	Data type(Size)	Description
username	NOT NULL	Varchar(50)	unique id
password	NOT NULL	Varchar(50)	password
bank_name	FOREIGN KEY from u_phn	Varchar(50)	bank names if user have multiple bank acc.

➤ Bank

Entity	Constraint	Data type(Size)	Description
bank_name	NOT NULL	Varchar(50)	user's bank name
employee name	PRIMARY KEY	Varchar(50)	user name
balance		Number(12)	balance of user
no. of transaction		Number(12)	number of transactions

➤ UPI

Entity	Constraint	Data type(Size)	Description
username	PRIMARY KEY	Varchar(50)	unique id
bank_name	NOT NULL	Varchar(50)	user's bank name
UPI	NOT NULL	Number(6)	UPI number
u_phn	NOT NULL	Number(10)	user phone number

6.0 Unit Testing

Black-Box Testing

- Black Box means opaque object or box where we can't see the internal structure.
- In Black box testing, we only concentrate on Input and output.
- Black-box test design is usually described as focusing on testing functional requirements. So we normally test the functionality of software without going deep in to its code and structure.
- Techniques that is used in Black Box testing
 1. Boundary-value analysis
 2. Error guessing
 3. Race conditions
 4. Cause-effect graphing
 5. Syntax testing
 6. State transition testing
 7. Graph matrix
 8. Equivalence partitioning
- Best example of Black box testing is Search on Google. User just enters keywords and get the expected results in turn. End user don't worry about what is behind this screen that is employed to fetch these results.
- Technical background is not the necessity of Black-Box Tester.

White-Box Testing

- White-Box testing is also known as clear box testing, glass box testing, transparent box testing and structural testing.
- White-Box testing deals with the internal structure and the internal working rather than only functionality.
- For White-Box testing, programming background is must because this helps in creating test cases for white-box testing.
- White-box testing is applied for Unit testing, integration testing and sometime in it is also used for system testing.
- These are few techniques that are used in white box testing
 1. Code Coverage
 2. Segment coverage: This is done to ensure that all statement or each line of code has been executed
 3. Compound condition coverage and loop coverage: In this we test all the conditions and all the branching and loops in code
 4. Data Flow testing: we test all the intermediate steps, in this we test how sequential steps behave
 5. Path Testing: In this we test all the path that is defined in code