

# Software Requirement Specification (SRS) for MY BILL BUDDY

---

*Abhinav Kapoor(2009003)*

*Surbhi Jain(2009047)*

*Priyanshi Mittal(2009034)*

# Contents

---

1. Introduction
  - 1.1 Purpose
  - 1.2 Scope
  - 1.3 Definitions, Acronyms and abbreviations
  - 1.4 Overview
2. Requirements
  - 2.1 Essential Requirements
  - 2.2 Desirable Requirements
3. Use Cases
  - 3.1 Important Use Cases
  - 3.2 UML Case Diagram
4. Descriptions
  - 4.1 Non Functional Requirements
  - 4.2 Design and Implementation Constraints
5. Operating Environment
  - 5.1 Hardware Platform
  - 5.2 Operating System and other application
  - 5.3 User Interface Requirements
6. Data Flow Diagram
  - 6.1 DFD
  - 6.2 Data Dictionary

# 1. Introduction:

## 1.1 Purpose:

My Bill Buddy is intended for an easy, secure, instant and cashless transfer of small money amount to friends through mobile phones. This document is meant to delineate the features of My Bill Buddy, so as to serve as a guide to the developers on one hand and a software validation document for the prospective client on the other.

## 1.2 Scope:

We describe what features are in the scope of the software and what are not in the scope of the software to be developed.

### In Scope:

- a. Registration through IIIT-D mail account and phone number
- b. User authentication.
- c. Transfer of money through phone applet.
- d. Sending encrypted SMS to server when a person has to pay to the other, with the receiver's number and amount.
- e. Sending notifications from server to phone via SMS whenever there is a transaction.
- f. Sending monthly account balance notification via SMS.

### Out of Scope:

- a. Trustworthy bank body to take care of physical money transactions.
- b. To make sure the person who has to pay will surely pay the money through the application.

## 1.3 Definitions, Acronyms and Abbreviations

### Acronyms and Abbreviations:

- A) SRS: Software Requirements Specification
- B) SMS: Short Messaging Service

### Definitions

- A) User – It refers to the person who will register and use the phone application.
- B) Server – It refers to the system which has complete database of all the transactions and users and also has a SMS gateway attached to send and receive SMS.
- A) Bank – It refers to the secure body which takes care of physical transactions of money and updating of database whenever there is such transfer.

## 1.4 Overview:

The rest of this SRS is organized as follows: Section 2 gives an overall description of the software. It gives what level of proficiency is expected of the user, some general constraints while making the software and some assumptions and dependencies that are assumed. Section 3 gives specific requirements which

the software is expected to deliver. Functional requirements are given by various use cases. Some performance requirements and design constraints are also given. Section 4 gives some possible future extensions of the system.

## 2. Requirements

### 2.1 Essential Requirements:

- A) A Trustworthy Bank: A trustworthy body to act as a bank is a primary requirement for our project. For now we are implementing this project inside IIIT-D so the body needs to be recognized by the institute and the students should be able to trust it to a level that they can deposit their money here.
- B) A Dedicate GSM Device: A GSM receiver is needed at server to continuously listen for the incoming messages about transactions and account related enquiries.
- C) SMS system on Device: A SMS sending and receiving system is must at the user device. The applet will use this mode of communication for transaction requests and account enquiries.
- D) Java on Phone: The phone device should also be Java Enabled.
- E) Dedicated Server: A dedicated server at bank to record, process and manage the money and transactions.

### 2.2 Desirable Requirements:

- A) Notification via E-Mail: Notification of transactions can be sent via E-mail as well as SMS.
- B) Lending money: The user can lend money to his friend in emergency.
- C) Cloud Environment: Our server can be put on cloud and used as a Web App.

## 3. Use Cases

### 3.1 Important Use Cases:

Online Course Feedback System should support the following use cases:

Class of use cases	Primary Actors	Use cases	Description of use cases
Use cases to User	User	Registration	User registers through his/her IIIT-D mail account.
		Entering the number to whom	User starts the

		you have to pay and the amount.	application and enters the number of the person whom he has to pay and the amount to be paid.
		Depositing money	Depositing the money in the bank in the institute
		Collecting money	Withdrawing money from the bank
Use cases related to Bank	Bank	Safe Keeping	Keeping all the money at secure and safe place
		Deposit	Bank deposits the money in a person's account
		Withdraw	Bank withdraws the money from a person's account
		Check balance	Bank checks the current balance in a person's account
Use cases related to Server	Server	Send notifications	Send notifications to the person via SMS whenever there is a transaction in his/her account

### 3.2 UML Case Diagram:

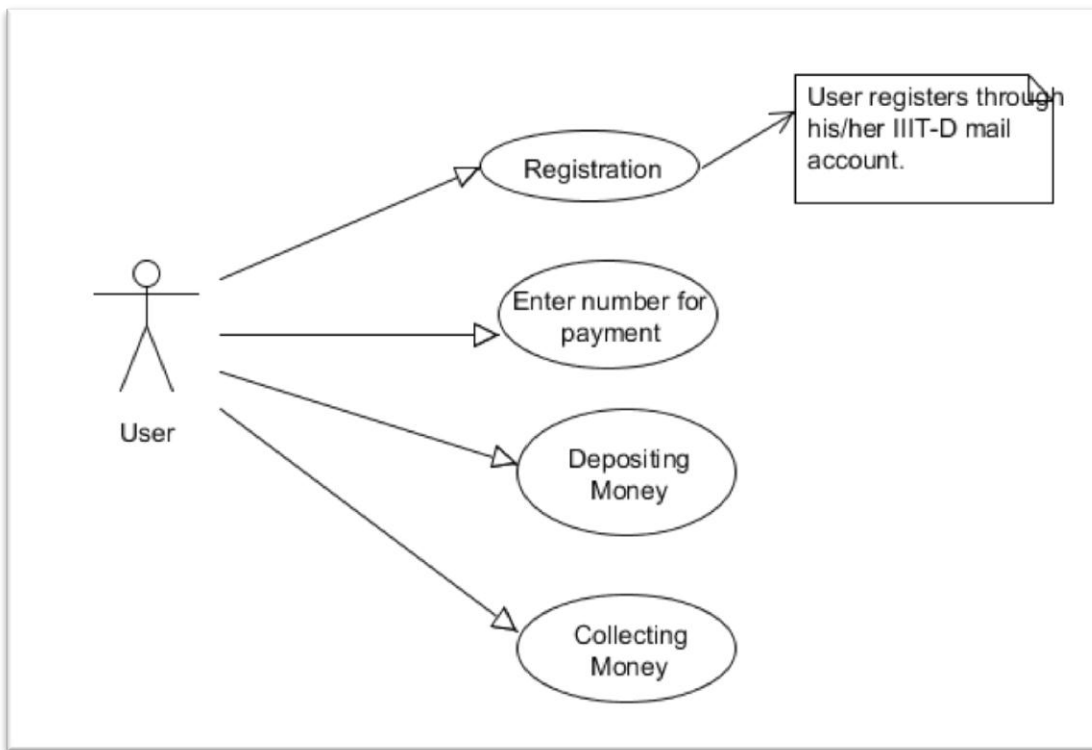


Figure 1 User Use Cases

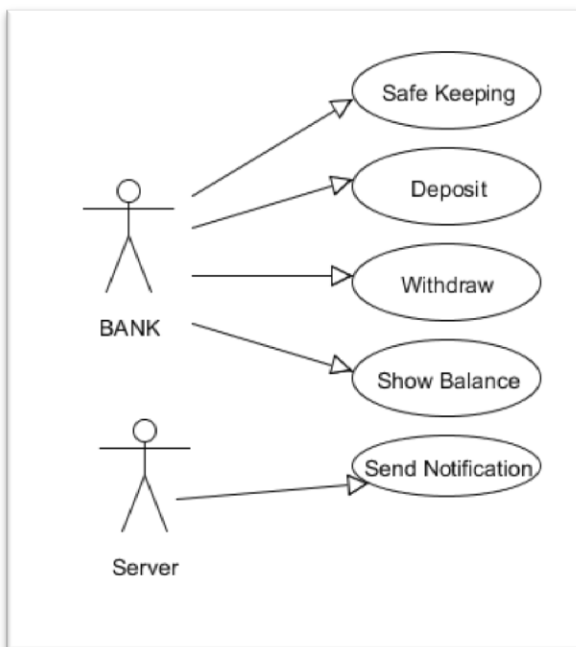


Figure 2 : Bank Use Cases

## 4. Descriptions

### 4.1 Non Functional Requirements:

#### Security

A) Encryption: The conversation between the applet and server monitoring the records will be encrypted.

B) Database Security: The database maintained at the server will be made secure from external attacks.

#### Performance

A) The time between sending request for transfer and receiving conformation from server will be minimized, even though it will depend a lot on the network provider.

B) The database will be maintained to provide fast response for searching and other operations.

### 4.2 Design and Implementation Constraints:

- A) Continuously running server.
- B) Balance problem in GSM modem for sending and receiving messages.
- C) Java enabled phone not available with everyone.
- D) Only small amount of money can be transfer possible.
- E) Completely secure banks.

## 5. Operating Environment:

### 5.1 Hardware Platform:

- A) GSM modem and a SIM card.
- B) PC with decent performance (>2.0 GHz dual core, >2 GB RAM, >70GB secondary space).
- C) Phone Device (capable of sending SMS).

### 5.2 Operating Systems and other Applications:

- A) Continuously running server with operating system Windows/Linux and a GSM modem connected to it.
- B) Java support on phone device.

### 5.3 User Interface Requirements:



#### Welcome Screen:

This screen contains the logo and 1 line description including a welcome message. It has a two buttons:

1. New User: It will be used by a new user to create a new account. It will take the user to new account creation screen.
2. Existing user: It will take the user to a new screen where he will be asked to enter his/her password to use the application assuming he is already a registered user.
3. Exit: To Exit the applet.



### Sign up Screen:

This screen will allow user to register using his IIIT-D mail account. He will be asked for his IIIT-D email address, password, name, phone number, and a password for future application usages. The buttons are:

1. DONE: After entering the details the user will press this to create a new account. If the details are incorrect he will be asked to reenter the details.
2. BACK: It will allow the user to go back to main screen without creating an account.

### Existing User Screen:

This screen welcomes an already registered user and contains a password field for authenticating the transactions. The user has to enter the password in the field and then select the operation he/she wishes to perform. The options are:

1. Make Payment: This option will first confirm the password. If incorrect password is provided, user will be asked to re-enter the password. After authentication a new screen will appear for payment.
2. Check Balance: This option also authenticates the user first, generating error message if required. After authentication, the user is shown a new screen with his account details.

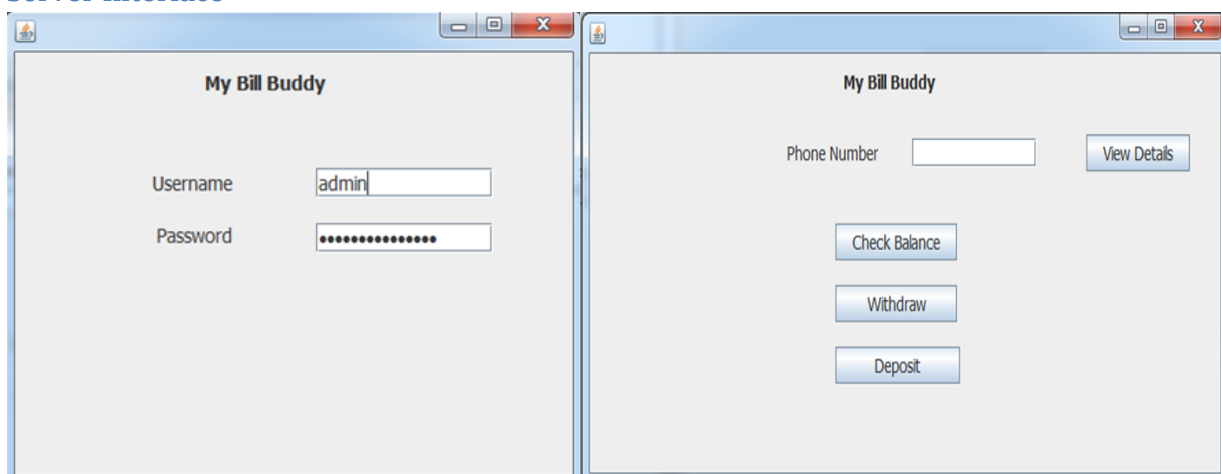
### Make Payment Screen:

It assumes that the user is authenticated. It asks for the phone number of the person to make payment to. Also he should be a registered user of MY BILL BUDDY too. The next field is to enter the amount to be paid. It must be less than your account balance. After pressing the ok button the application will send the details to server and wait for confirmation. Any error if happens will be reported or a success notification will be sent to the user and also to the person getting the payment.

### Account Balance Screen:

It displays the current account balance of the user.

### Server Interface



The image displays two side-by-side screenshots of a web application titled "My Bill Buddy".

The left screenshot shows a login form with the following elements:

- Username:
- Password:

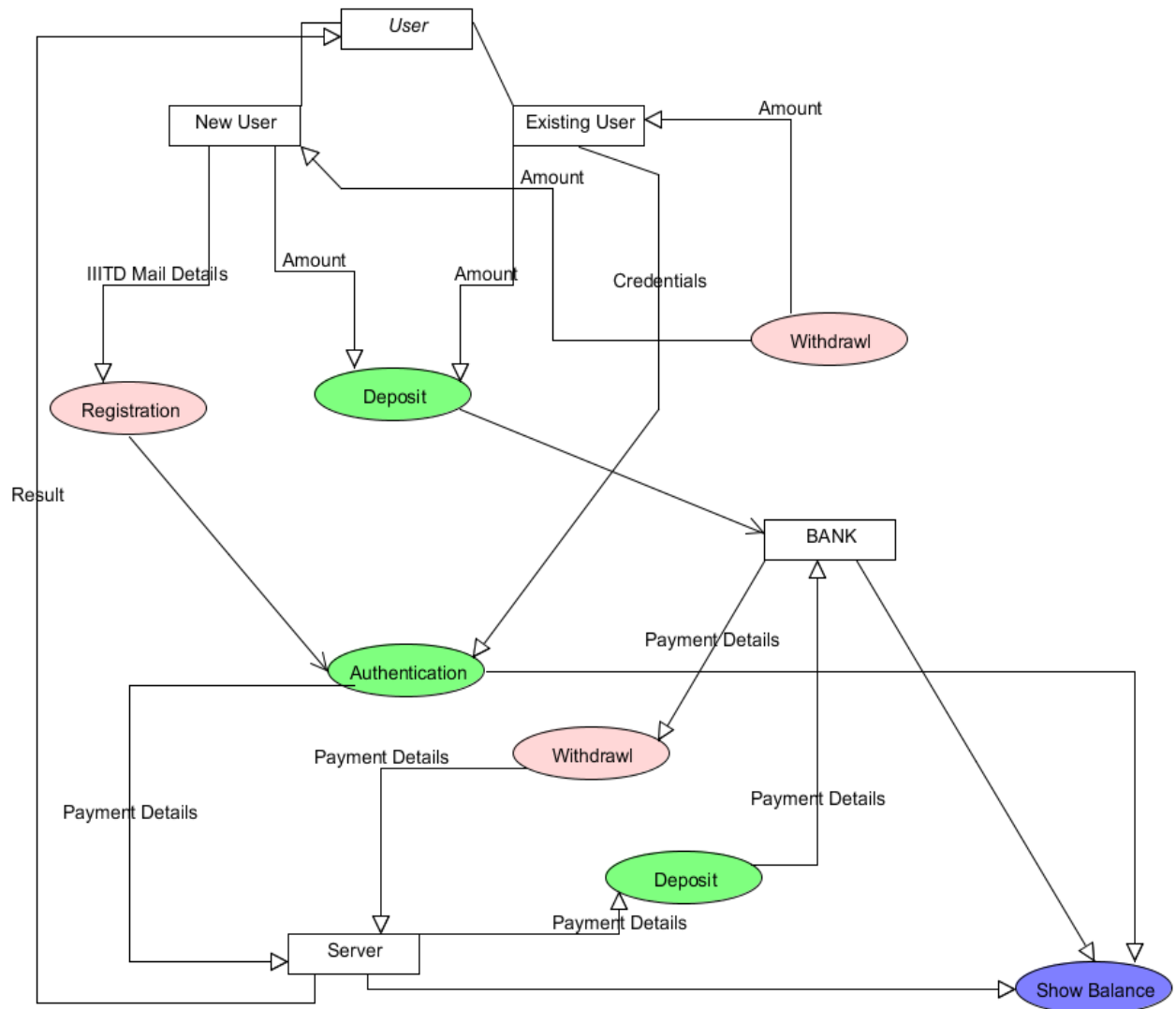
The right screenshot shows the main interface with the following elements:

- Phone Number:
- View Details:
- Check Balance:
- Withdraw:
- Deposit:

The Server will be to manage the account of the users. They will deposit and withdraw the money at the bank. The server is also present at bank. On the server the administrator will deposit and withdraw money for users using ADMIN authentication. These transactions are physical money transfers

## 6. Data Flow Diagram

### 6.1 DFD:



## 6.2 Data Dictionary:

- A) IIITD\_Mail\_Details = IIITD\_Email\_Address + Password + Name + Phone\_Number + Application\_Password
- B) Amount = [ Deposit\_Money | Withdrawal\_Money ]
- C) Credentials = Application\_Password
- D) Payment\_Details = Phone\_Number + Amount
- E) Result = Notification\_String + [ Success | Failure ]