1. INTRODUCTION

Microfinance services are provided to unemployed or low-income individuals because most of them have limited financial resources. They do not have enough income to do business with traditional financial institutions. This application is intended to provide microfinance facilities to such individuals and groups.

1.1. PURPOSE:

The purpose of this document is to provide a detailed description of an online microfinance system for small help groups, organizations or wherever applicable. Some basic financial services are provided.

1.2. SCOPE:

The software product will:

1) Allow the user to login or sign up to the system with email and password.
2) View and update personal details.
3) Create, view and manage savings accounts.
4) Allow the users to withdraw and deposit money.
5) Create and manage term deposits.
6) Create and manage loans.
7) Provide automated loan repayment.
8) Transfer money.

The software product will not:

1) Allow to login with 3rd party systems like google, facebook etc.
2) Provide debit or credit card facilities.
3) Provide share holding facilities.
4) Allow a user to take more than one loan at a time

1.3 DEFINITIONS, ACRONYMS AND ABBREVIATIONS

1.4. REFERENCES

This application has been prepared on the basis of discussion with team members, faculty members and also taking information from the following books and websites:

Books:

Fundamental of Software Engineering By Rajiv Mall
Software Engineering Seventh Edition Ian Sommerville

1.5 OVERVIEW

The remaining sections of this document provide a general description, including characteristics of the users of this project, the product's features, the functional and non-functional requirements of the product and the assumptions.

2. OVERALL DESCRIPTION

This section gives an overview of the whole system. The system and its subsystems are introduced with their basic functionality. It will also describe the stakeholders that will use the system and what functionality is available for each type. At last, the constraints and assumptions for the system will be presented.

2.1. PRODUCT PERSPECTIVE

This product is independent and self-contained.

2.2 USER CHARACTERISTICS

There are two types of users for this product, Admins and Customer. Each of these types has their own requirements.

Customers: Customers are the saving account holders.

Admin group: Can view and make internal changes to the system. It consists of:

a) Management: Consists of the management team of the client of the software system.
b) Collector: The collector will need to approve any withdrawal or deposit transactions.

2.3 CONSTRAINTS

A few constraints the system should follow are:

a) When creating a new savings account, mandatory fields must be filled.
b) Minimum balance criteria should be fulfilled before withdrawal is allowed.
c) Before granting a loan the specified criterias should be fulfilled.

2.4 ASSUMPTIONS

a) We assume that the personal details are given when a new saving account is created.
b) Savings interest rate and minimum balance requirement is shown when the user is creating a new account.
c) Balance is automatically incremented with respect to savings interest rate.
d) Automated loan repayment is disabled automatically if sufficient balance is not available.
e) Recipient of a transfer is assumed to be a user of the system.
f) Admin group members can not be customers of the system.
g) The collector will verify the amount manually and approve the transaction during deposit or withdrawal.

3 SPECIFIC REQUIREMENTS

3.1 USE CASE DIAGRAM

3.2 FUNCTIONAL REQUIREMENTS

- **The system shall allow the user to sign up:** New users will need to sign up to the system by providing name, email ID and password. The user will have to login to the system once the sign-up is complete.
- **The system shall allow the user to login:** Each Customer will have their Email Id and password. This page will require both of these attributes for them to access their account. The provided email id along with the password is matched with the database for authentication.
- **The system shall allow the members of the admin group to login:** Each member will have a designated ID and a password which they will need to provide to login to the system.
- **The system shall provide provisions**
- **to create savings account:**
    a) Users need to fill the form which will be verified by administration. If any logical inconsistencies are found, the user will need to re-enter correct data.
    b) An automated account ID will be generated after successful submission.
    c) The user will need to deposit an amount equal or greater than minimum balance required , to the collector. She will need to show the transaction ID and deposit the money to the collector. Then the account balance will be updated accordingly.
- **The system shall provide provisions to view savings account details:** The user will be able to view account balance, term deposits, loans taken, transaction history of the last 30 days.
- **The system shall provide provisions to manage savings account details:**
  The user will be able to request for withdrawal and deposit of money from/to her account. Minimum balance criteria should be fulfilled before withdrawal is allowed. A transaction ID is generated for the request. She will need to show the transaction ID and deposit the money to the collector. Then the account balance will be updated accordingly.
- **The system shall provide provisions to create term deposit:** The user will be able to create term deposit by specifying amount and time period of the deposit.
- **The system shall provide provisions to manage term deposit:** The user can break the term deposit prematurely or renew the deposit.
- **The system shall provide provisions to create a new loan:**
  The following criterias need to be fulfilled to get a loan:
    a) User needs to specify her annual income of the last financial year which should be acceptable.
    b) Past loans have to be cleared.
    c) The user needs to be a member for at least a year.
    d) The user will be given tenure options for loan repayment of the loan.
- **The system shall provide provisions to manage loan:**
  The user will be able to change the loan repayment method. She will be able to repay the remaining loan amount at once. The user can repay every installment at any time before due by depositing the amount in the savings account.

- **The system shall provide provisions for automated loan repayment:** Loan is automatically repaid on the due date.
- **The system shall provide provisions to transfer money:** User can send money to another account in the same system provided sufficient balance is available. User needs to provide the account number of the recipient.
- **The system shall provide provisions to manage an account by the management team:** A member of the management team can view account details, term deposits, loan details, transaction history. They can also manage term deposits and loans.
- **The system shall provide provisions for a collector to approve a deposit/withdrawal transaction.**

3.3. NON-FUNCTIONAL REQUIREMENTS:

Non-functional requirement specifies criteria that can be used to judge the operation of a system, rather than specific behaviours (stated by the functional requirements).

Performance requirement:

Database operations should be small and fast. Transactions in turn will also be faster.

Logical database requirement:

We shall store the following information:

- Login credentials
- User profile details
- User account details
- Term deposits by the user
- Loan details
- Transaction details

Security requirement:

a) The system shall automatically logout all customers after a period of inactivity.
b) The interface shall never display the user's password. It shall always be shown in special characters representing typed characters.
c) The systems back-end features shall only be accessible to authenticated administrators.
d) Transactions shall be either completed entirely or not at all.
e) Data shall be stored in a secure environment.
f) Sensitive data should be encrypted before being sent over insecure connections like the internet.

Software quality requirement:

Availability: The system should be available at all times, except when the server is in maintenance. In case of a hardware failure or database corruption, a replacement page will be shown. Also in this case, backup of the database should be retrieved from the server. Then the services will be resumed.

Correctness: Only valid transactions by the system shall be allowed.

Maintainability: Database shall maintain correct information about the user. Operations should be isolated in nature.Maintenance will be easier when operations don't overlap. Software should be modular in nature so that maintenance can be done efficiently.

Reliability: Backup of the database is continuously maintained and updated to reflect the most recent changes. Reliability of the system depends on the reliability of individual modules.

Portability: The application is HTML and scripting language based. The end-user part is fully portable and any system using any web browser (with the most recent version supporting HTML5) should be able to use the features of the system, including any hardware platform that is available or will be available in the future