# **SOFTWARE REQUIREMENTS SPECIFICATIONS**

## **Use case diagram**

Attached

## **Use case descriptions.**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Use Case Number | 1 | | | | | | | |
| Name | Login | | | | | | | |
| Brief Description | This use case allows a user to Login | | | | | | | |
| Actors | Group Member / System Administrator / Account Manager | | | | | | | |
| Preconditions | N/A | | | | | | | |
| Flows of Events | | | | | | | | |
| Basic Flows |  | | | | |  | | |
| Step | User Actions | | | | | System Response | | |
| 1 | The user opens the application on his/her device. | | | | | The application pops up the Login screen | | |
| 2 | The user enters his login credentials | | | | | The system validates and checks if user exits in the system and presents an appropriate screen | | |
| Alternate Flows |  | | | | |  | | |
| 1 | 1. In case the user enters a wrong username, the system responds with user doesn’t exist. 2. Incase the user enters a correct username but with a wrong password, system responds with "incorrect password" message. | | | | | | | |
| Post Conditions | User Logged in successful message is displayed | | | | | | | |
| Business Rules | Every user should have a specific role type i.e. Member, Account Manager (Is also a member), System Admin | | | | | | | |
| Use Case Number | 2 | | | | | | | |
| Name | Logout | | | | | | | |
| Brief Description | This use case allows a user to Logout | | | | | | | |
| Actors | Group Member / System Administrator / Account Manager | | | | | | | |
| Preconditions | User should be logged in. | | | | | | | |
| Flows of Events | | | | | | | | |
| Basic Flows |  | | | | |  | | |
| Step | User Actions | | | | | System Response | | |
| 1 | The user clicks on the logout button | | | | | The system closes the current session and exits | | |
| Post Conditions | Message confirming user Logout success | | | | | | | |
| Business Rules | N/A | | | | | | | |
| Use Case Number | 3 | | | | | | | |
| Name | Make Deposit | | | | | | | |
| Brief Description | This use case allows a member to make a monthly contribution to the group account | | | | | | | |
| Actors | Group Member | | | | | | | |
| Preconditions | A member should have logged in | | | | | | | |
| Flows of Events | | | | | | | | |
| Basic Flows |  | | | | |  | | |
| Step | User Actions | | | | | System Response | | |
| 1 | Member selects make deposit option | | | | | System presents a payment form | | |
| 2 | Member fills the payment form and clicks on submit button | | | | | System validates input and updates Database | | |
| Alternate Flows |  | | | | |  | | |
| 1 | 1. If user provides an amount that is less than the minimum deposit, system responds with " Please provide an amount bigger than $XXX". 2. If User provides wrong card details, system responds with "Please provide correct card details" 3. If user provides wrong mobile number, system responds with "Invalid mobile number". 4. If user provides a number/card with insufficient funds to what he wants to deposit to the App, system responds with "Account has insufficient funds" | | | | | | | |
| Post Conditions | Successful deposit message | | | | | | | |
| Business Rules | Deposited amount should be greater than minimum set deposit | | | | | | | |
| Use Case Number | 4 | | | | | | | |
| Name | View Statement | | | | | | | |
| Brief Description | This use case enables a member to view their financial statement | | | | | | | |
| Actors | Group Member | | | | | | | |
| Preconditions | User should have logged-in | | | | | | | |
| Flows of Events | | | | | | | | |
| Basic Flows |  | | | | |  | | |
| Step | User Actions | | | | | System Response | | |
| 1 | Member clicks on view statement option | | | | | System allows member to choose a range (Start date to end date) | | |
| 2 | Member chooses range and submits | | | | | System retrieves a statement as per request | | |
| Alternate Flows |  | | | | |  | | |
| 1 | If user provides invalid data "No Statement Found" message is displayed. | | | | | | | |
| Post Conditions | User's Statement successfully retrieved message is displayed. | | | | | | | |
| Business Rules |  | | | | | | | |
| Use Case Number | 5 | | | | | | | |
| Name | Second Member | | | | | | | |
| Brief Description | This use case allows a group member to stand surety of another group member taking out a loan | | | | | | | |
| Actors | Group Member | | | | | | | |
| Preconditions | 1. Group Member has logged in to the system 2. Group Member has been selected to be a seconder | | | | | | | |
| Flows of Events | | | | | | | | |
| Basic Flows |  | | | | | |  | |
| Step | User Actions | | | | | | System Response | |
|  | The group member selects the approval tab | | | | | | The system displays all the pending approvals for the group member | |
|  | The group member selects the member from the approval list | | | | | | The system updates the database and notifies the member seeking approval | |
| Alternate Flows | | | | | | | | |
| 1. In case a member does not want to second another member, he clicks the reject button and the system notifies the member seeking a loan. | | | | | | | | |
| Post Conditions |  | | | | | |  | |
| 1. Group Member successfully seconds another member whenever approval is made 2. A message alert is displayed to the group member confirming approval | | | | | | | | |
| Business Rules |  | | | | | |  | |
| 1. A member can only second another member if he currently has no running loan 2. If approval is not made within 3 months from the date of submission, the system automatically cancels the request and notifies the member seeking a loan | | | | | | | | |
| Use Case Number | | 6 | | | | | | | |
| Name | | Withdraw Interest | | | | | | | |
| Brief Description | | This use case allows a member to withdraw accrued interest on savings | | | | | | | |
| Actors | | 1. Group Member | | | | | | | |
| Preconditions | | 1.Group Member has logged in to the system 2.There is accrued interest on the members account | | | | | | | |
| Flows of Events | | | | | | | | | |
| Basic Flows | |  | |  | | | | | |
| Step | | User Actions | | System Response | | | | | |
|  | | The group member selects tab to withdraw interest | | The system displays the interest accrued by the member and asks how much the member wishes to withdraw | | | | | |
|  | | The group member enters the amount they want to withdraw | | System forwards payment request to payment subsystem and updates the database | | | | | |
| Alternate Flows | |  | |  | | | | | |
| 1. If the member enters a value of withdrawal more than they are entitled to, the system should inform the member that they can’t withdraw more than they are entitled to and asks them to re-enter a value | | | | | | | | | |
| 1. If Member has no interest available, system should inform member that they have zero interest | | | | | | | | | |
| Post Conditions | |  | |  | | | | | |
| 1. Interest withdrawn successfully from the members account whenever a member has accrued interest and has entered a valid amount. | | | | | | | | | |
| Business Rules | |  | |  | | | | | |
| 1. Interest on member accounts is paid annually | | | | | | | | | |
| Use Case Number | | 7 | | | | | | | |
| Name | | Quit group | | | | | | | |
| Brief Description | | This use case allows a member to make a request leave a group. | | | | | | | |
| Actors | | 1. Group Member 2. Account Manager | | | | | | | |
| Preconditions | | 1.Group Member has logged in to the system | | | | | | | |
| Flows of Events | | | | | | | | | |
| Basic Flows | | | | | | | | | |
| Step | | User Actions | | | System Response | | | | |
|  | | The group member selects the quit group button | | | The system displays group exit form for member to input feedback | | | | |
|  | | The group member fills the quit form and selects submit | | | The system requests user to confirm if they really want to quit the group | | | | |
|  | | Group member clicks confirm button to quit | | | Form is sent to Account manager for approval | | | | |
| Alternate Flows | |  | | |  | | | | |
| 1. Member changes his mind and does not confirm quitting; the system clears the fields then closes the quit form. | | | | | | | | | |
| Post Conditions | |  | | |  | | | | |
| 1. Quit form submitted successfully to account manager for approval | | | | | | | | | |
| Business Rules | |  | | |  | | | | |
| 1. If the request is not approved by the Account manager within 3 months from the date of submission, the request is automatically approved. 2. Group Member has no loan running currently | | | | | | | | | |
| Use Case Number | | 8 | | | | | | | |
| Name | | Loan Application | | | | | | | |
| Brief Description | | This use case allows a member to apply for a loan | | | | | | | |
| Actors | | 1. Group Member 2. Account Manager | | | | | | | |
| Preconditions | | 1.Group Member has logged in to the system | | | | | | | |
| Flows of Events | | | | | | | | | |
| Basic Flows | |  | |  | | | | | |
| Step | | User Actions | | System Response | | | | | |
|  | | The group member selects tab to apply for a loan | | The system displays information about the member and the amount of loan a member is eligible to get.  The system then asks the user to proceed | | | | | |
|  | | The group member selects proceed | | System displays form for application for a loan | | | | | |
|  | | Group member fills the form which includes seconders and clicks submit | | System validates fields in the form and sends approval requests to the group members entered as seconders | | | | | |
| Alternate Flows | |  | |  | | | | | |
| 1. If the member enters a loan amount more than what he is entitled to, the system in this case displays an information message to the member and asks the member to enter a lower amount | | | | | | | | | |
| 1. Sureties are not valid as per business rules, the system in this case displays an information message and asks the member to select other sureties. | | | | | | | | | |
| Post Conditions | |  | |  | | | | | |
| 1. Loan application form submitted for processing | | | | | | | | | |
| Business Rules | |  | |  | | | | | |
| 1. Group member can take a loan of up to twice his savings 2. Savings for sureties should be able to cover the amount by which the loan value exceeds the members savings | | | | | | | | | |
| Use Case Number | | | 9 | | | | | | |
| Name | | | Generate Report | | | | | | |
| Brief Description | | | This use case allows a member to generate a report for a selected period. | | | | | | |
| Actors | | | System Administrator, Account Manager | | | | | | |
| Preconditions | | | User has logged in to the system | | | | | | |
| Flows of Events | | | | | | | | | |
| Basic Flows | | |  | | | | |  | |
| **Step** | | | **User Actions** | | | | | **System Response** | |
| 1 | | | The group member selects the generate report tab and chooses a period range | | | | | The system displays a summary of all the transactions executed by the member in that period | |
| 2 | | | The member can now choose to print or email the statement | | | | | The system responds by either sending a mail or provide options for printing the statement | |
| Post Conditions | | | The system records this event for persistence either in a logFile or in the DB | | | | | | |
| Business Rules | | |  | | | | |  | |
| Use Case Number | | | 10 | | | | | | |
| Name | | | Member management CRUD | | | | | | |
| Brief Description | | | This use case allows the Account Manager to create , read and update member details | | | | | | |
| Actors | | | Account Manager | | | | | | |
| Preconditions | | | Account Manager has logged in to the system | | | | | | |
| Flows of Events | | | | | | | | | |
| Basic Flows | | |  | | | | |  | |
| 10.1 Create Member | | | | | | | | | |
| **Step** | | | **User Actions** | | | | | **System Response** | |
| 1 | | | The manager initiates the create member command | | | | | The system presents a form with fields including email, dateOfBirth, dateOfJoing, Names and Address | |
| 2 | | | The manager fills the form and then requests the system to save member details | | | | | The system responds by validating input and checking if this member already exists. Then finally adds member to that account | |
| Post Conditions | | | 1. The system displays a toast or message confirming to the admin the member has been successfully added.  2.The system records this event for persistence either in a logFile or in the DB2. | | | | | | |
| Business Rules | | | The member must have a unique ID | | | | | | |
| 10.2 Edit Member | | | | | | | | | |
| **Step** | | | **User Actions** | | | | | **System Response** | |
| 1 | | | The manager requests to view a list of members | | | | | The system presents a list of all members | |
| 2 | | | The manager then chooses one member whose details he desires to edit | | | | | The system then presents a form with existing member details | |
| 3 | | | The manager then edits the member information as he wishes and clicks the save button | | | | | The system then validates input and overwrites member info with updated info | |
| Post Conditions | | | 1. The system displays a toast or message confirming to the admin the member has been successfully updated.  2.The system records this event for persistence either in a logFile or in the DB2. | | | | | | |
| Business Rules | | | The member unique ID cannot be edited | | | | | | |
| 10.3 Read/View Member | | | | | | | | | |
| **Step** | | | **User Actions** | | | | | **System Response** | |
| 1 | | | The manager initiates the view member command | | | | | The system presents a statement with members details | |
| Post Conditions | | | 1.The system records this event for persistence either in a logFile or in the DB2. | | | | | | |
| Business Rules | | | The member must have a unique ID | | | | | | |
| 10.4 Approve Member Exit Request | | | | | | | | | |
| Preconditions | | | 1.Account Manager has logged in to the system 2.There should be a request from member to exit | | | | | | |
| Step | | | User Actions | | | | | System Response | |
| 1 | | | The manager clicks on disable button | | | | | The system responds by changing the status to inactive | |
| Post Conditions | | | 1. The system displays a toast or message confirming to the manager that the member has been disabled  2. The sends an email or SMS to all members about his exit 3. The system records this event for persistence either in a logFile or in the DB2. | | | | | | |
| Business Rules | | | The member must have cleared all his outstanding loans | | | | | | |
|  | | | | | | | | | |

## **Glossary**

1. Account Manager: A person who is a member of a group, tracks the financial position of a group, manages group members, approves financial transactions on behalf of the group and generates summary reports about the financial position of the group.
2. Deposit: Amount of money deposited by a group member on the savings account of the group.
3. Group: A group of people that come together with a common goal of saving money with the aim of improving their financial position.
4. Group Member: A person that belongs to one or more savings groups.
5. Interest: This is a payment from the group to a group member of an amount of money above the savings of the group member.
6. Loan: A sum of money offered to a group member from the savings account of the group that is expected to be paid back with interest
7. Report: Summary of financial transactions of the group or an individual member.
8. Seconder: A group member who stands surety of another group member that is taking a loan from the group.
9. System Administrator: A person who operates and maintains the SenteAPP.
10. Savings Account: An account in a bank all the deposits of the group members are safely stored and where withdrawals are made from in case a group members loan is approved.

## **Supplementary Specifications**

The purpose of this document is to define non-functional requirements of this SenteApp system. This Supplementary Specification lists the requirements that are not readily captured in the use cases of the use-case model. The Supplementary Specifications and the use-case model together capture a complete set of requirements on the system.

* Scope: This Supplementary Specification applies to the SenteApp System. This specification defines the non-functional requirements of the system; such as reliability, usability, performance, and supportability as well as functional requirements that are common across several use cases. (The functional requirements are defined in the Use Case Specifications.).
* Reliability: The main system must be running 95% of the time.
* Performance: The system shall support up to 15 simultaneous users against the central database
* Security: None
* Design Constraints: Our system shall interact with a payment, email and Short message subsystems