**Software Requirements specification**

**University wallet application**

**Version <1.0>**

[Revision history 1](#_30j0zll)

[1.Introduction 1](#_1fob9te)

[1.1 Purpose 1](#_3znysh7)

[1.2 Document Conventions 1](#_2et92p0)

[1.3 Intended Audience and Reading Suggestions 2](#_tyjcwt)

[1.4 Product Scope 2](#_3dy6vkm)

[1.5 References](#_1t3h5sf)

[2](#_1t3h5sf)

[2. Overall Description 2](#_4d34og8)

[2.1 Product Perspective: 2](#_2s8eyo1)

[2.2 Product Functions: 2](#_17dp8vu)

[2.3 User Classes and Characteristics: 2](#_3rdcrjn)

[2.4 Operating Environment: 3](#_26in1rg)

[2.5 Design and Implementation Constraints: 4](#_lnxbz9)

[2.6 User Documentation: 4](#_35nkun2)

[2.7 Assumptions and Dependencies : 4](#_1ksv4uv)

### [3. External Interface Requirements 5](#_2jxsxqh)

[3.1 User Interfaces 5](#_z337ya)

[3.2 Hardware Interfaces 7](#_3j2qqm3)

[3.3 Software Interfaces 7](#_1y810tw)

[3.4 Communications Interfaces](#_4i7ojhp)

[7](#_4i7ojhp)

[4. System Features 7](#_2xcytpi)

[5. Other Nonfunctional Requirements : 9](#_1ci93xb)

[Appendix 9](#_3whwml4)

[Glossary 20](#_2bn6wsx)

### **Revision history**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Date** | **Reason for changes** | **Version** |
| Maleeha | 16-5-2024 | Requirement changes | 1.1 |

## 

## 

## 

## **1.Introduction**

### 1.1 Purpose

The purpose of this document is to explain the mobile wallet application, its detailed requirements, constraints, and interfaces. The document is intended for users and potential system users such as developers and project managers.

### 1.2 Document Conventions

This document is based on IEEE standard for SRS.

### 1.3 Intended Audience and Reading Suggestions

* Student's Primary users who will perform transactions, maintain a record book and journal, and view transaction graphs.
* University Staff: Stakeholders such as cafe workers, accounts, stationery personnel, etc.
* Programmers: Those who will maintain or develop the application further.
* System Test Engineers: Those who will refer to the document for testing purposes.

### 1.4 Product Scope

The application focuses on student financial management, providing features for performing transactions within university premises, maintaining a journal, viewing a record book of transaction history, and linking with banks or third-party payment gateways for transactions, receiving, or loading money.

### 1.5 References

IEEE template

<https://dspmuranchi.ac.in/pdf/Blog/srs_template-ieee.pdf>

## **2. Overall Description**

### 2.1 Product Perspective:

The software is self-contained, created by students to help their peers manage finances on university premises, reducing the burden of tracking expenses and potentially aiding in savings.

### 2.2 Product Functions:

* **Journal:** Tracks where and when transactions occur and allows planning for balance usage.
* **Record book:** Logs transaction details including when, where, to whom, and balance received. Tracks remaining balance.
* **Pie chart:** Visualizes spending in a pie chart for easy understanding.
* **Profiles:** Users create profiles with unique usernames and passwords for authentication and authorization.

### 2.3 User Classes and Characteristics:

* **Primary user classes:**

1. **Student:** Manage balance, make payments for university services, track transaction history.
2. **University staff:** Accept payments, make transactions, check balances with basic technical expertise.

* **Secondary user classes:**

1. **University administrator:** Monitors usage, manages accounts, resolves issues.
2. **Development and Maintenance team:** Debugs issues, implements new features, performs maintenance.

### 2.4 Operating Environment:

The software operates on smartphones and smart devices with the following operating systems:

**Operating systems:**

1. **Android:**

* Android 6.0 (Marshmallow)
* Android 7.0/7.1 (Nougat)
* Android 8.0/8.1 (Oreo)
* Android 9 (Pie)
* Android 10-11
* Android 12 (latest version)

1. **iOS:**

* Versions 7 to 15.

1. **Windows:**

* Windows Mobile
* Windows Mobile, Phone 7, 8, 8.1.

1. **Watch OS:**

* Versions 1 to 5.

### 2.5 Design and Implementation Constraints:

* Requires devices with at least 4GB RAM and 32GB ROM.
* Screens smaller than 4.7 inches are not supported.
* Only works with university student IDs.
* Supports Urdu and English languages.
* Maintained by the university IT department.
* Regular backups are necessary.

### 2.6 User Documentation:

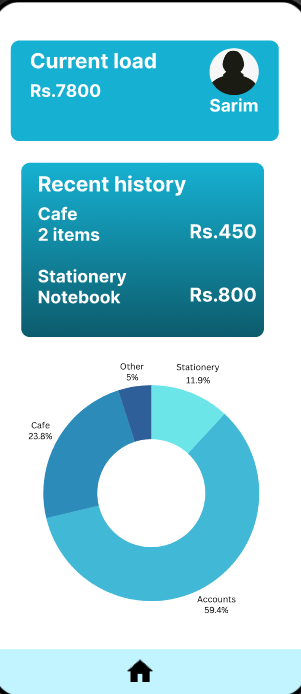
User manuals and tutorials will be provided.

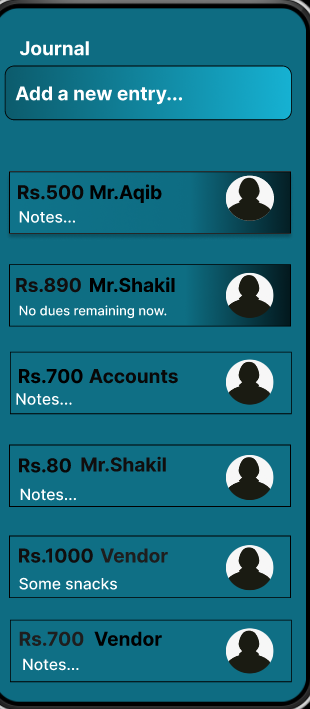
### 2.7 Assumptions and Dependencies:

* The university provides a secure API for student data access and payment processing.
* App functionality depends on the stability of university APIs.
* Maintenance and updates depend on the development team's availability and expertise.
* Must comply with university policies, data protection, and privacy laws.
* The development and maintenance budget must be approved and allocated by the university’s financial department.

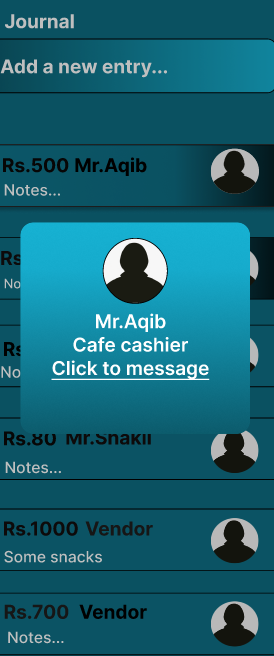
### **3. External Interface Requirements**

### 3.1 User Interfaces

1. **Homepage of wallet app showcasing current balance, name, recent transactions & month** **spendings in form of pie-chart.**



1. **Journal feature of wallet where the student will input his/her on-ground transactions**

****

1. **An overlay feature depicting the profiles of the senders/receivers.**

****

1. **Record book of student**

### 3.2 Hardware Interfaces

No specific hardware interfaces required.

### 3.3 Software Interfaces

* Database connection via MySQL/Heidi SQL/Oracle.
* Compatibility with Android and iOS mobile operating systems.

### 3.4 Communications Interfaces

* Requires an internet connection for online transactions and updates.
* Connects to SQL server for generating pie charts.
* Uses TCP protocol for online communications.

## **4.System Features**

4.1 App shall provide online transaction

4.1.1 Description and priority

The user should have a balance equal to or more than Rs/- 300 in his account in order to send money to someone online. User shall add receiver ID.

This requirement is of high priority.

**Risk:** User will not be able to perform online transaction if he will not have money less than Rs/- 300 due to which this app will not be helpful for him.

**Risk Rate:** 9

4.1.2 Functional Requirements

**R2:** The app should track at least 5 types of transactions, including purchases from campus stores, transfers between users, and withdrawals from linked bank accounts.

4.2 App shall provide current balance display

4.2.1 Description and priority

The current balance should be prominently displayed on the app's home screen providing users with real-time access to their financial status.

This requirement is of high priority.

4.2.2 Stimulus/Response Sequences

The user should login to the app in order to use this feature.

4.2.3 Functional Requirements

**R5:** Display the current balance prominently on the app's home screen. Ensure that users can view their balance within 1 second of opening the app.

4.3 User shall see stakeholder's profile

4.3.1 Description and priority

Profiles of university stakeholders involved in transactions should include relevant information such as name, position, contact details, and department affiliation, accessible within the app for user reference.

This requirement is of low priority.

**Risk:** If the university is wide with many stakeholders, then user will not be able to see all stakeholder profiles.

**Risk Rate:** 1

4.3.2 Stimulus/Response Sequences

The user should login to the app in order to see stakeholder's profile.

4.3.3 Functional Requirements

**R14:** Gather specific details for stakeholder profiles, including name, contact details, and role.

**R15:** Ensure that stakeholder profiles are visible within the app and include information for at least 10 stakeholders.

4.4 The user shall view record book

4.4.1 Description and priority

The user shall be able to check the record of their online transactions.

This requirement is of high priority.

**Risk:** If there are more than 5 types of transactions performed by the user, then user will not be able to see all of them.

**Risk Rate:** 9

4.4.2 Stimulus/Response Sequences

The user should click on the record checking tab to view the record book.

4.4.3 Functional Requirements

**R2:** The app should track at least 5 types of transactions, including purchases from campus stores, transfers between users, and withdrawals from linked bank accounts.

4.5 The user shall view pie chart

4.5.1 Description and priority

Users shall be able to view their transactions graphically on a pie chart.

This requirement is of high priority

**Risk:** If the categories are more than 5, then the user will not see all of them on the pie chart.

**Risk Rate:** 9

4.5.2 Stimulus/Response Sequences

The user shall click on view pie chart tab in to view pie chart.

4.5.3 Functional Requirements

**R9:** The app should generate a pie chart visualization of user spending, using transaction data categorized by department or spending type, with a minimum of 5 categories represented.

**R10:** A triadic color scheme should be used to enhance comprehension.

4.6 The user shall share his wallet

4.6.1 Description and priority

Users should have the option to share their wallet with friends or university mates, with configurable permissions to restrict access to certain features or transaction types.

This requirement is of high priority.

**Risk:** Loss of privacy and the user may have some security issues.

**Risk Rate:** 9

4.6.2 Stimulus/Response Sequences

The user should login on app in order to use this feature.

4.6.3 Functional Requirements

**R17:** Users should have the option to share their wallet with up to 5 friends or university mates, with configurable permissions to restrict access to certain features or transaction types.

4.7 App shall provide educational content

4.7.1 Description and priority

The app should provide articles, videos, quizzes, and interactive tools focused on improving financial literacy, reducing stress, and promoting responsible spending habits among students. (These help you understand how to manage your money better, like making budgets, saving, investing, and borrowing wisely.

This requirement is of low priority.

4.7.2 Stimulus/Response Sequences

The user should login to the app in order to see educational content.

4.7.3 Functional Requirements

**R1:** Provide at least 50 educational articles, 20 tutorials, and 30 interactive quizzes covering topics such as budgeting, saving, investing, and responsible borrowing.

4.8 App shall provide a journal

4.8.1 Description and priority

Journal should keep track of where and when did you made a transaction. It keeps track and students can also decide how they want to use their balance.

This requirement is of high priority.

4.8.2 Stimulus/Response Sequences

Users should enter the amount and the receiver's name to whom he is sending money to use this feature.

* + 1. Functional Requirements

**R18:** The journal system shall maintain a record of all transactions made by the student, including the date, time, and location of each transaction.

**R19:** The system shall also provide a feature for students to view their transaction history and track their balance usage.

**R20:** The system shall enable students to create a plan for their balance usage, including setting budget limits and tracking their progress towards those limits.

**R21:** The journal system shall provide a one-click options on interface for students to easily access and manage their transaction history and balance usage plan.

**5.Other Nonfunctional Requirements:**

1. **Performance:**

Users can view their balance within 3 seconds.

1. **Security:**

Includes encryption and measures like fingerprint/pattern authentication.

1. **Scalability**:

Supports up to 100 concurrent users.

1. **Reliability**:

Ensures 95% availability.

1. **Capacity:**

Handles up to 200 transactions per minute.

1. **Data integrity**:

Maintains less than 2% error rate in transactional records.

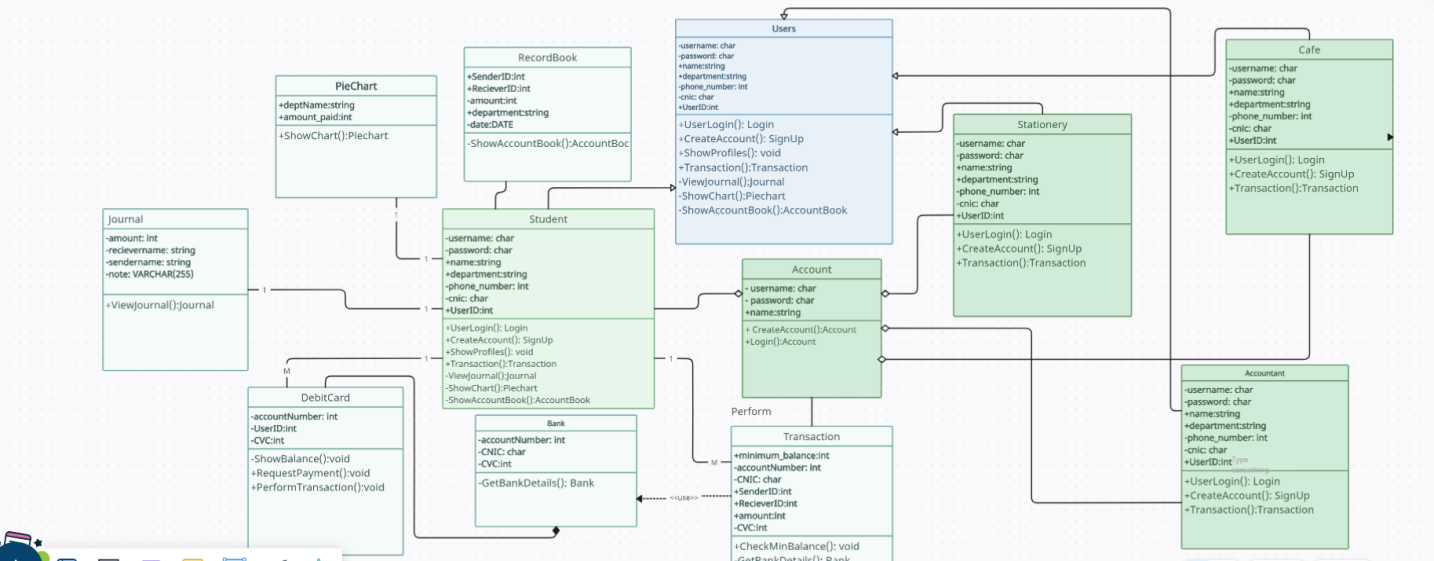
1. **Resource utilization**:

Requires devices with at least 4GB RAM and 32GB ROM. Not supported on screens smaller than 4.7 inches.

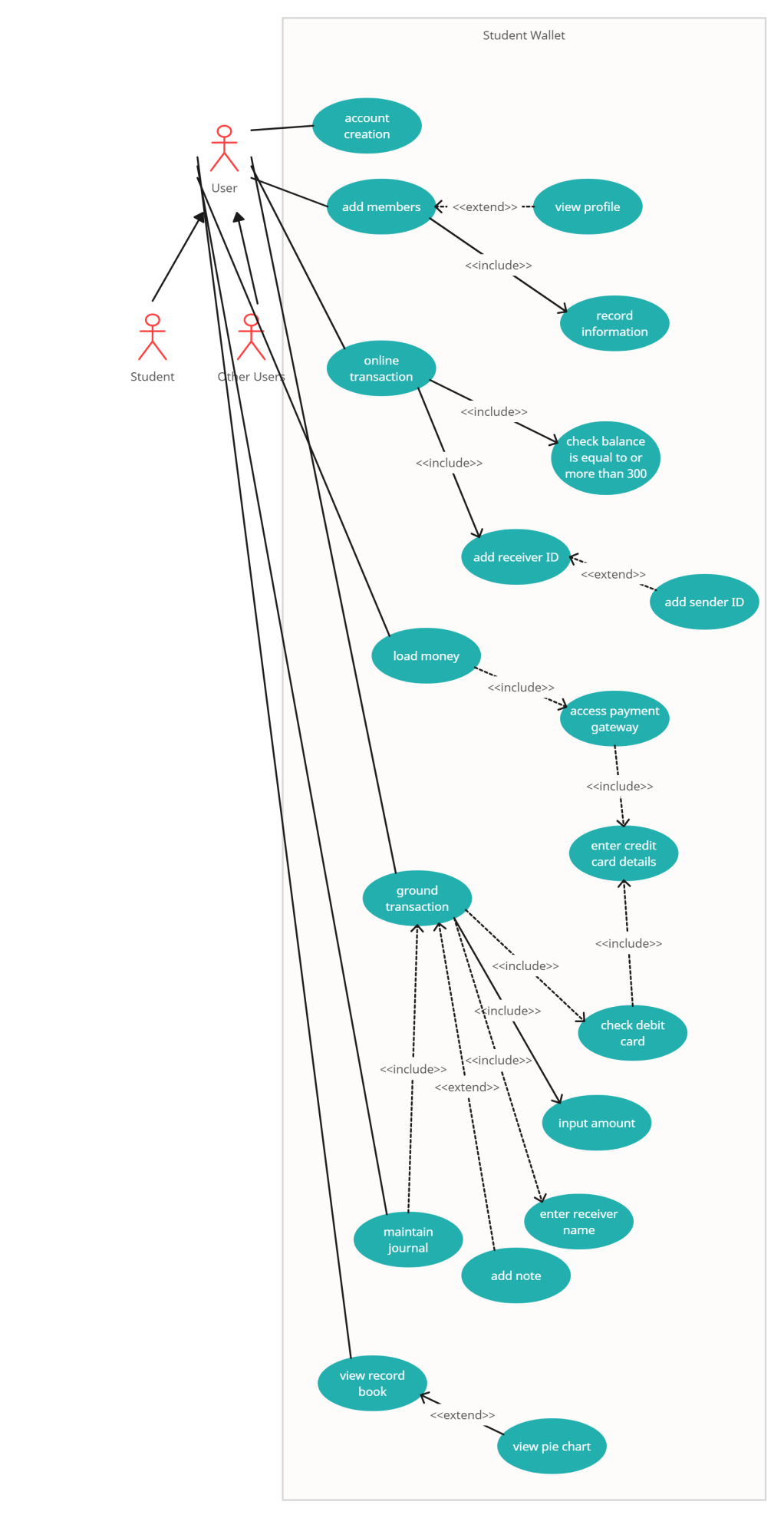
### **Appendix**

**Diagram -1**

**CLASS DIAGRAM:**



**DIAGRAM 2**



Use Case Description:

|  |  |
| --- | --- |
| **Use Case ID** | UC-1.1 |
| **Use Case Name** | Account creation |
| **Actors** | Students and other users |
| **Description** | Users will enter their username and password due to which their account will be created, and they can use the app. |
| **Trigger** | Username and password. |
| **Preconditions** | Users will enter username and password. |
| **Post conditions** | An account will be created. |
| **Normal Flow** | Users will enter their username and password due to which their account will be created, and they can use the app. |
| **Alternative Flow** | Create Account |
| **Exception** | In the step of normal flow, if the user enters wrong password, then user will not be able to access the app due to which he/she will reset the password and try gain. |
| **Includes** | Add username and add password |
| **Special Requirements** | Add username and add password |
| **Assumptions** | The user must understand English. |
| **Notes and issues** | - |

|  |  |
| --- | --- |
| **Use Case ID** | UC-2.1 |
| **Use Case Name** | Add members |
| **Actors** | Students and other users |
| **Description** | Users can add all other members so he/she can easily view their profile before the transaction and their information will be recorded. |
| **Trigger** | Users will add members |
| **Preconditions** | Users will add members |
| **Post conditions** | Users can send balance to any other of his/her members or can pay his/her member’s dues. |
| **Normal Flow** | Users can add all other members so he/she can easily view their profile before the transaction and their information will be recorded. |
| **Alternative Flow** | - |
| **Exception** | - |
| **Includes** | Adding members. |
| **Special Requirements** | Add members. |
| **Assumptions** | - |
| **Notes and Issues** | - |

|  |  |
| --- | --- |
| **Use Case ID** | UC-3.1 |
| **Use Case Name** | Online Transaction |
| **Actors** | Students and other users |
| **Description** | Users will be able to perform transactions online from their account. |
| **Trigger** | Must login account first. |
| **Preconditions** | Login account |
| **Post conditions** | Users will transact money. |
| **Normal Flow** | Users will be able to perform transactions online from their account. |
| **Alternative Flow** | Ground transaction. |
| **Exception** | If there are less than 300 moneys in User’s account, then the user will not be able to send money online to anyone. In this case ground transactions will be performed. |
| **Includes** | Minimum money in account to send someone should be 300 rupees |
| **Special Requirements** | Minimum money in account to send someone should be 300 rupees |
| **Assumptions** | Minimum money in account to send someone should be 300 rupees |
| **Notes and Issues** | - |

|  |  |
| --- | --- |
| **Use Case ID** | UC-4.1 |
| **Use Case Name** | Load money |
| **Actors** | Students and other users |
| **Description** | Users can load money using their bank account or any other third-party payment gateway. |
| **Trigger** | Enter account number, CNIC, name and CVC. |
| **Preconditions** | Enter account number, CNIC, name and CVC. |
| **Post conditions** | Users will be able to transact money online. |
| **Normal Flow** | By entering account number, CNIC, name and CVC then users can load money using their bank account or any other third-party payment gateway. |
| **Alternative Flow** | Ground transaction. |
| **Exception** | If there are less than 300 moneys in the user’s account, then the user will not be able to send money online to anyone. In this case ground transactions will be performed. |
| **Includes** | Minimum 300 rupees in account. |
| **Special Requirements** | Minimum 300 rupees in account. |
| **Assumptions** | - |
| **Notes and Issues** | - |

|  |  |
| --- | --- |
| **Use Case ID** | UC-6.1 |
| **Use Case Name** | Maintain Journal |
| **Actors** | Students and other users |
| **Description** | Users can maintain their journal as well on ground transaction. |
| **Trigger** | Debit or credit card |
| **Preconditions** | Entering the amount, receiver name. |
| **Post conditions** | Users can maintain journal. |
| **Normal Flow** | Users can maintain their journal as well on ground transaction by entering amount and receiver name. |
| **Alternative Flow** | - |
| **Exception** | If user forgets his/her debit card pin, then he/she will contact bank. |
| **Includes** | Users should have a debit or credit card |
| **Special Requirements** | Users should have a debit or credit card |
| **Assumptions** | - |
| **Notes and Issues** | - |

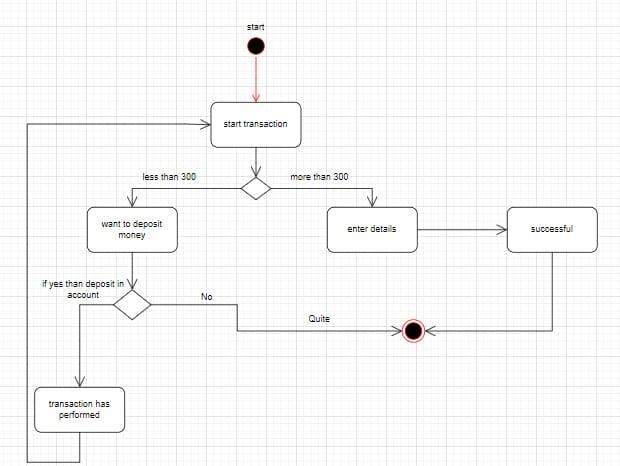
|  |  |
| --- | --- |
| **Use Case ID** | UC-7.1 |
| **Use Case Name** | Review recorded transaction |
| **Actors** | Students and other users |
| **Description** | Users will be able to check the record of their transactions online. |
| **Trigger** | - |
| **Preconditions** | Just clicking on the record checking tab. |
| **Post conditions** | Record of amount paid and the sender/receiver's name, department with date. |
| **Normal Flow** | Users will be able to check the record of their transactions online by just clicking on the record checking tab. |
| **Alternative Flow** | Graphical representation |
| **Exception** | - |
| **Includes** | - |
| **Special Requirements** | - |
| **Assumptions** | - |
| **Notes and Issues** | - |

|  |  |
| --- | --- |
| **Use Case ID** | UC-8.1 |
| **Use Case Name** | View pie chart |
| **Actors** | Students and other users |
| **Description** | Users will be able to view their transactions graphically on a pie chart. |
| **Trigger** | - |
| **Preconditions** | Clicking on the tab |
| **Post conditions** | Graphical representation. |
| **Normal Flow** | Users will be able to view their transactions graphically on a pie chart by just clicking on the tab. |
| **Alternative Flow** | Record checking tab |
| **Exception** | - |
| **Includes** | - |
| **Special Requirements** | - |
| **Assumptions** | - |
| **Notes and Issues** | - |

**Diagram 3**

**ACTIVITY DIAGRAM**

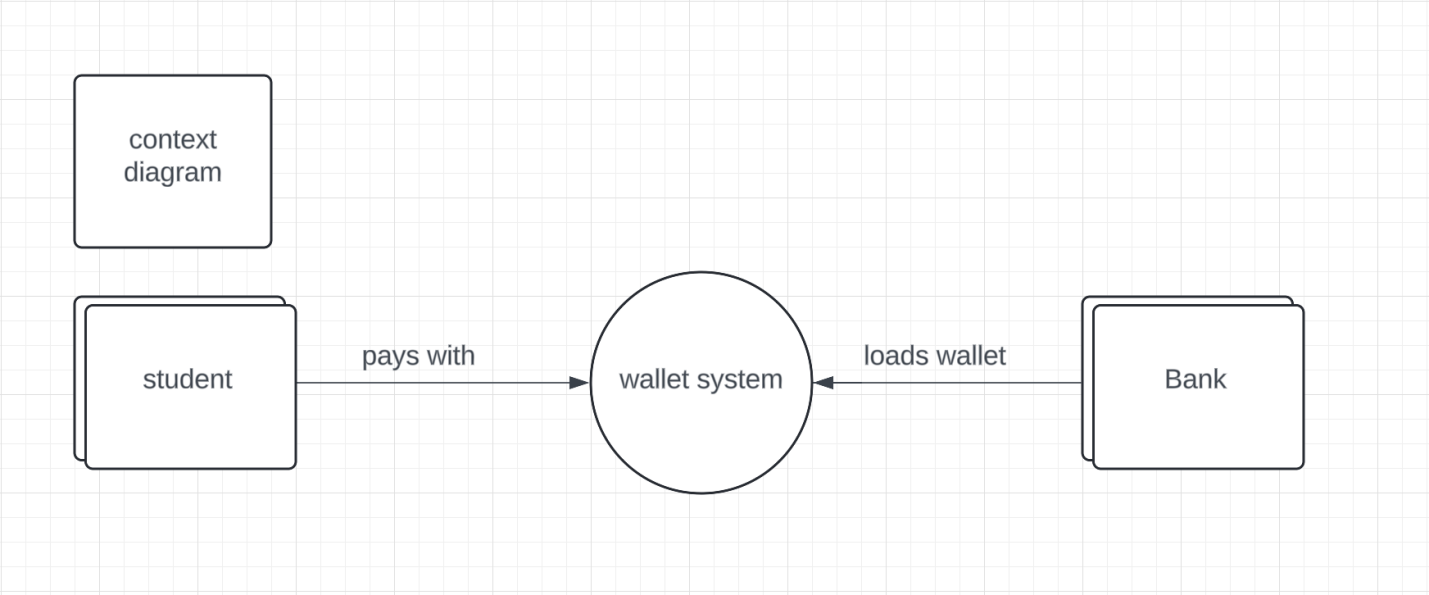
**Decomposing transaction scenario when the user’s load is less than minimum balance.**

****

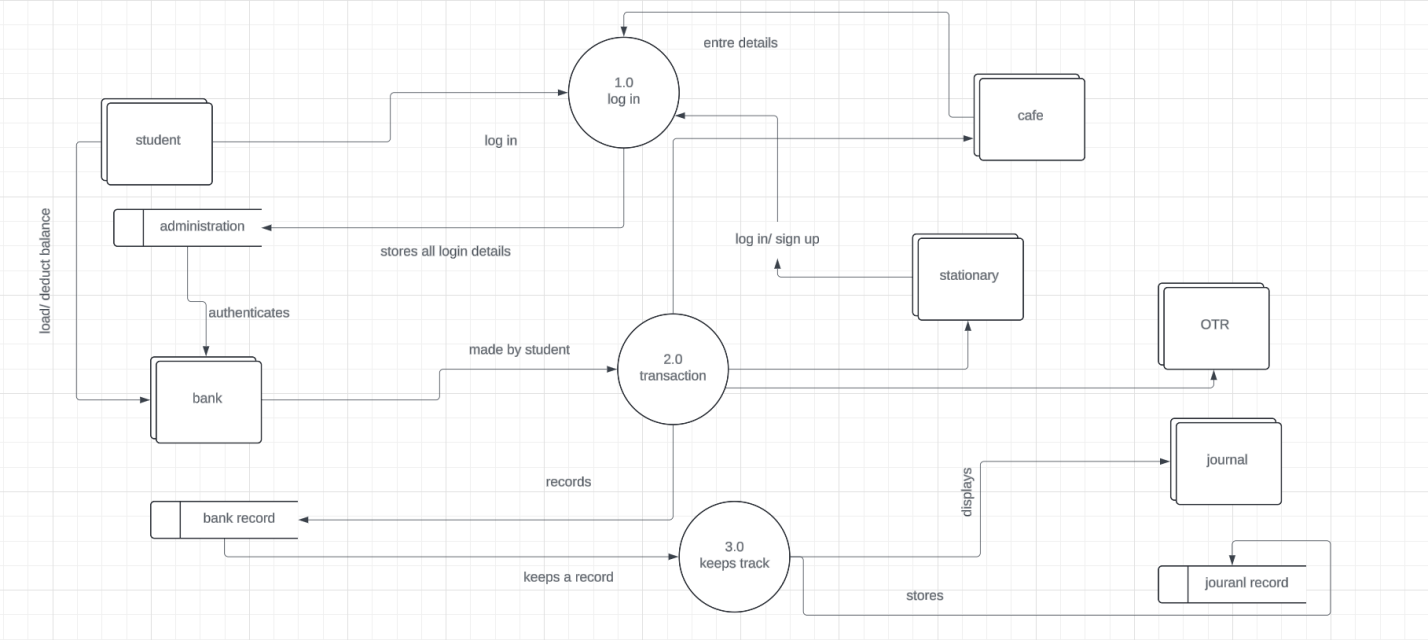
**Diagram 4**

**DATA FLOW DIAGRAM**

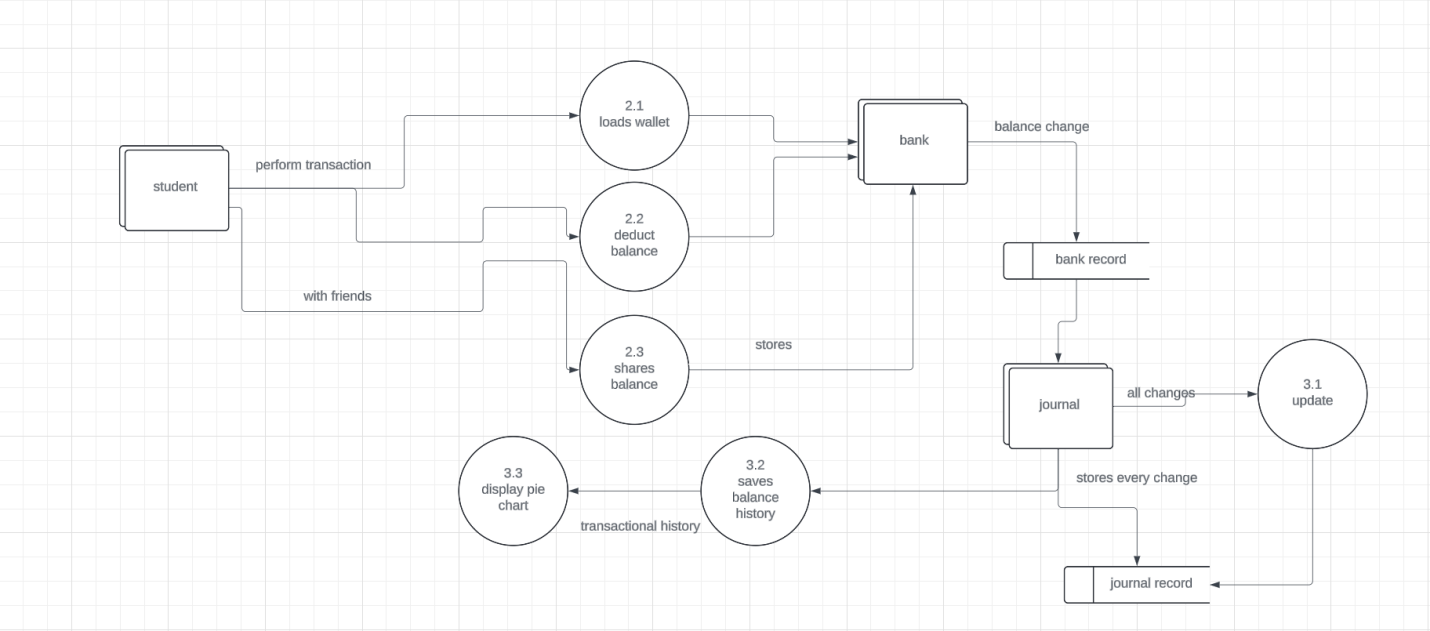
**Context diagram:**

****

**Level 1 diagram:**

****

**Level 2 diagram:**

****

### **Glossary**

|  |  |
| --- | --- |
| **Term** | **Definition** |
| Programmers | Who writes code to develop an application |
| Operating environment | An operating environment is the place where users run application software or programs |
| Transaction Gateway: | A system that helps process payments securely and efficiently, like a doorway for money transfers |
| RAM | Random access memory (RAM) is the hardware in a computing device that provides temporary storage for the operating system |
| ROM | Read-only memory, or ROM, is a type of computer storage containing non-volatile, permanent data that, normally, can only be read, not written to |
| Database | Collection of all the information attained by this system |
| API | An application programming interface is a way for two or more computer programs or components to communicate with each other. |
| SQL/Heidi SQL/Oracle server | relational database management system |
| Android operating server | operating system that primarily runs on smartphones and tablets. |
| IOS | Apple iOS is the operating system for iPhone, iPad, and other Apple mobile devices. |
| Real-time access | Having immediate access to information or systems, without delay, like instant access. |
| Configuration | The process of setting up or arranging something, like a computer or system, to work in a specific way. |
| Concurrent | Happening or done at the same time, like multiple tasks or processes running simultaneously. |
| Debug | To find and fix errors or mistakes in a system, program, or code, like troubleshooting. |
| Watch OS | Operating system of apple that extends the functionality of apple |