EZPAY

**“SECURE AND EFFICIENT DIGITAL PAYMENT SOLUTION”**

# Business Requirements Document

## Contents

Project Overview

Stakeholders

Key Assumptions and Constraints

Business Flow Diagram

Requirements

Functional Requirements

Non-Functional Requirements

Security Requirements

Support & Maintenance Requirements

Use Cases

Use Case 1: User Management

Use Case 2: Payment Processing

Use Case 3: Transaction Management

Use Case 4: Notification System

Use Case 5: Security and Compliance Use Case 6: User Interface

Reference Links: .................................................................................................................................... 21

Version and Approvals

|  |  |  |  |
| --- | --- | --- | --- |
| VERSION HISTORY | |  |  |
| **Version** | **Date** | **Revised By** | **Reason for change** |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

## Project Details

|  |  |
| --- | --- |
| **Project Name** |  |
| **Project Type** |  |
| **Project Start Date** |  |
| **Project End Date** |  |
| **Project - Business Analyst** |  |
| **Project Head / Scrum Master** |  |

## Overview

**Project Overview**

### **Project Overview and Background**

EzPay aims to develop a secure and efficient digital payment application that allows users to make UPI payments and bank transfers, manage their transactions, and receive timely notifications. The app will provide a user-friendly interface accessible across various devices, ensuring a seamless payment experience for users while maintaining high security standards and regulatory compliance.

Key Features:

User registration and authentication

Profile management

UPI payments

Bank transfers

Transaction history and status tracking

Notification system for confirmations and reminders

Responsive design for multi-device accessibility

Robust security measures including data encryption and fraud prevention

Compliance with financial regulations

## Stakeholders

|  |  |  |
| --- | --- | --- |
|  | **STAKEHOLDERS** | **ROLE** |
| 1. | Product Owner/Manager | Product Owner/Manager: Responsible for overall product vision, strategy, and management. |
| 2. | Development Team | Responsible for building and maintaining the app using Java, Angular, Spring Boot, and Oracle database. |
| 3. | Quality Assurance Team | Responsible for testing and ensuring the quality of the application. |
| 4. | Customer Support Team | Handles user inquiries and provides assistance. |
| 5. | Users | Individuals using the app for digital payments and transaction management. |

## Key Assumptions and Constraints

### **Key Assumptions and Constraints**

|  |  |
| --- | --- |
|  | **Assumptions** |
| 1 | Users have access to smartphones or computers with internet connectivity. |
| 2 | Users have valid bank accounts and/or UPI IDs for making transactions. |
| 3 | The necessary APIs and integrations with banks and UPI services are available. |
| 4 | Users will provide accurate personal information during registration. |
| 5 |  |
| 6 |  |
|  | **Constraints:** |
| 1 | The app must comply with relevant financial regulations and data protection laws. |
| 2 | The system must integrate securely with existing banking and UPI infrastructures. |
| 3 | The app must maintain high performance and low latency even with a large user base. |
| 4 | Development must be completed within the specified timeframe and budget. |
| 5 | The app must comply with relevant financial regulations and data protection laws. |

Business flow Diagram

## Requirements

### Functional Requirements

|  |  |  |
| --- | --- | --- |
| **FUNCTIONAL REQUIREMENTS** | |  |
| **S No** | **REQUIRMENTS** | |
| 1 | User registration with secure authentication  User login with multi-factor authentication  Profile management for updating personal information  Password recovery mechanism | |
| 2 | UPI payment functionality  Bank transfer functionality  Support for multiple payment methods | |
| 3 | Transaction Management  View and search transaction history  Track transaction statuses in real-time | |
| 4 | Notification System  Send transaction confirmations  Send payment reminders for upcoming due dates | |
| 5 | Security and Compliance  Implement data encryption for all sensitive information  Fraud detection and prevention mechanisms  Ensure compliance with financial regulations | |
| 6 | User Interface  Responsive design for multi-device compatibility  Intuitive navigation and user-friendly interface | |

### Non-Functional Requirements

|  |  |  |
| --- | --- | --- |
| **NON-FUNCTIONAL REQUIREMENTS** | |  |
| **AVAILABILTY** | The application should have 99.9% uptime. | |
|  |  | |
| **USABILITY** | The user interface should be intuitive and require minimal training for users. | |
| **CAPACITY** | The system should be easily scalable to accommodate growing user base. | |
| **SECURITY** | All data transmissions should be encrypted using industrystandard protocols.  The system should comply with PCI DSS standards for handling financial data. | |
| **Performance** | The application should process transactions within 3 seconds.  The system should support at least 10,000 concurrent users. | |

### Security Requirements

|  |  |  |  |
| --- | --- | --- | --- |
| **USER ACCESS / SECURITY REQUIREMENTS** | |  | |
| **Serial No.** | **Requirements** | | **Priority** |
| 1 | Implement end-to-end encryption for all data transmissions. | | Highest |
| 2 | Use secure authentication mechanisms, including multi-factor authentication. | | Highest |
| 3 | Regular security audits and penetration testing. | | Highest |
| 4 | Implement fraud detection algorithms to identify and prevent suspicious activities. | | Highest |
| 5 | Ensure compliance with data protection regulations (e.g., GDPR, if applicable). | | Highest |

### Support & Maintenance Requirements

|  |  |  |  |
| --- | --- | --- | --- |
| **SUPPORT AND MAINTENANCE REQUIREMENTS** | |  | |
| **Serial No.** | **Requirements** |  | **Priority** |
| 1. | Provide 24/7 customer support through multiple ch chat, email, phone).  Regular system updates and patches to address secu vulnerabilities. | annels (e.g., rity | High |
| 2. | Maintain comprehensive system logs for auditing a troubleshooting purposes.  Implement a robust backup and disaster recovery pl | nd an. | High |

Use Cases

### **Prioritization method for analyzing the user story**

**Technique used:** MoSCoW

By providing a framework within which each demand is assessed in relation to other needs, this technique Prioritizes requirements.

For example: Is it necessary or desirable? Is it a feature that could make the product better or a perfect aspect for the future?

1. ***Must Have –*** Without this, a use-case cannot be delivered before the deadline. Without this, there would be no sense in completing the initiative by the deadline; similarly, if the solution weren't deployed by the planned date, there would be no point in doing so either. For use cases designated as "Must have," it is illegal and unsafe to use without it. It is not possible to present the Business Case without it.

1. ***Should Have –*** Crucial but not necessary, and while it could be challenging to omit, the solution is still a plausible option. You might need to temper expectations, deal with certain bottlenecks, and overcome it, finish the paperwork, or discover a resolution that is already in place. One can discern between a Should Have and a Could Have by looking at the degree of difficulty caused by it not being satisfied, in terms of corporate value or the number of people affected.

1. ***Could Have –*** Desirable or wanted yet not of paramount importance. Less of an influence if omitted. In essence, it specifies a need that is deemed preferable but not essential.

1. ***Won’t Have –*** Indicates a requirement that stakeholders have acknowledged but won't be executed in the current version however it may be taken into consideration in the future.

#### Use Case 1: User Management

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case ID:** | 1 |  |  |
| **Use Case Name:** | User Management |  |  |
| **Created By:** | ChandraSekhar | **Last Updated By:** | ChandraSekhar |
| **Date Created:** |  | **Date Last Updated:** |  |

##### User story 1.1 : User Registration and Login

|  |  |
| --- | --- |
| **Actors** | User, System |
| **Description** | User registers and logs into the EzPay application. |
| **Preconditions** | ser has a valid email address and mobile number. User has access to the EzPay application. |
| **Postconditions** | User account is created and stored in the database.  User is logged into the application. |
| **Normal Course** | 1. User opens the EzPay application. 2. User selects "Register" option. 3. User enters required information: name, email, mobile number, and password. 4. System validates the entered information. 5. System sends a verification code to the user's mobile number. 6. User enters the verification code. 7. System creates the user account and displays a success message. 8. User proceeds to login page. 9. User enters registered email/mobile and password. 10. System validates the credentials and grants access to the application. |
| **Alternative Courses** | 1. If the email or mobile is already registered, the system displays an error message. 2. If the verification code is incorrect, the system allows the user to request a new code. 3. If login credentials are incorrect, the system displays an error message and allows retry. |
| **Frequency of Use** |  |
| **Assumptions** |  |
| **Notes and Issues** |  |

##### User Story 1.1: Wireframe Login Credentials

Screen Verification

OTP Verification

Error in Screen verification

Invalid Credentials

Forgot Password

Reattempt with Wrong Credentials

##### User story 1.2: Profile Management

|  |  |
| --- | --- |
| **Actors** | User, System |
| **Description** | User updates their profile information. |
| **Preconditions** | User is logged into the EzPay application. |
| **Postconditions** | User's profile information is updated in the database. |
| **Normal Course** | 1. User navigates to the "Profile" section. 2. System displays current profile information. 3. User selects "Edit Profile" option. 4. User modifies desired information (e.g., name, address, profile picture). 5. User selects "Save Changes". 6. System validates the updated information. 7. System saves the changes and displays a success message. |
| **Alternative Courses** | If any entered information is invalid, the system displays an error message and allows correction |
| **Exceptions** |  |
| **Frequency of Use** |  |
| **Assumptions** |  |

##### User Story 1.2: Wireframe

Sign-up Page

Account Created successfully.

User Id already Exist.

##### User story 1.3: Password Recovery

|  |  |
| --- | --- |
| **Actors** | User, System |
| **Description** | User recovers forgotten password |
| **Preconditions** | User has a registered account in the EzPay application |
| **Postconditions** | User's password is updated in the database. |
| **Normal Course** | 1. User selects "Forgot Password" on the login screen. 2. User enters registered email or mobile number. 3. System verifies the entered information. 4. System sends a password reset link to the user's email. 5. User clicks on the reset link. 6. User enters a new password and confirms it. 7. System validates the new password. 8. System updates the password and displays a success message. |
| **Alternative Courses** | If the entered email/mobile is not registered, the system displays an error message.  If the new password doesn't meet security requirements, the system prompts for a different password. |
| **Exceptions** |  |
| **Frequency of Use** |  |
| **Assumptions** |  |

##### User Story 1.3: Wireframe

Sign-up Page

Account Created successfully.

User Id already Exist.

#### Use Case 2: Payment Processing

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case ID:** | 2 |  |  |
| **Use Case Name:** | Payment Processing |  |  |
| **Created By:** | ChandraSekhar | **Last Updated By:** | ChandraSekhar |
| **Date Created:** |  | **Date Last Updated:** |  |

##### User story 2.1: UPI Payments

|  |  |  |
| --- | --- | --- |
| **Actors** | User, System, UPI Service | |
| **Description** | User makes a payment using UPI. | |
| **Preconditions** | User is logged into the EzPay application. User has a registered UPI ID. | |
| **Postconditions** | Transaction is recorded in the system. User's account balance is updated. | |
| **Normal Course** | 1. | User selects "Make UPI Payment" option. |
|  | 2. | User enters recipient's UPI ID and payment amount. |
|  | 3. | User enters a note for the transaction (optional). |
|  | 4. | User confirms the payment details. |
|  | 5. | System initiates the UPI transaction. |
|  | 6. | UPI Service processes the payment. |
|  | 7. | System receives confirmation from UPI Service. |
|  | 8. | System displays transaction success message to the user. |
| **Alternative Courses** | 1. | If the UPI ID is invalid, the system displays an error message. |
|  | 2. | If the payment fails due to insufficient funds, the system notifies the user. |
| **Assumptions** |  |  |
| **Notes** |  |  |

##### User Story 2.1: Wireframe

Home Page

Bills Overview

Debt Payments Bill Overview

House Rent Bill Overview

##### User story 2.2: Bank Transfers

|  |  |
| --- | --- |
| **Actors** | User, System, Banking Service |
| **Description** | User initiates a bank transfer. |
| **Preconditions** | User is logged into the EzPay application.  User has added and verified their bank account in the app. |
| **Postconditions** | Transfer is recorded in the system.  User's account balance is updated. |
| **Normal Course** | 1. User selects "Bank Transfer" option. 2. User enters recipient's bank details (account number, IFSC code). 3. User enters transfer amount and purpose of payment. 4. User confirms the transfer details. 5. System initiates the bank transfer. 6. Banking Service processes the transfer. 7. System receives confirmation from Banking Service. 8. System displays transfer success message to the user. |
| **Alternative Courses:s** | 1. If the bank details are invalid, the system displays an error message. 2. If the transfer fails due to any reason, the system notifies the user with the reason. |
| **Notes** |  |

##### User Story 2.2: Wireframe

Add bill details.

Successfully New Bill added

Upcoming/Overdue Bills

Overdue Bills

Payment screen

Upcoming Bills

Payment Screen

#### Use Case 3: Transaction Management

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case ID:** | 3 |  |  |
| **Use Case Name:** | Transaction Management |  |  |
| **Created By:** | ChandraSekhar | **Last Updated By:** | ChandraSekhar |
| **Date Created:** |  | **Date Last Updated:** |  |

##### User story 3.1: View Transaction History

|  |  |
| --- | --- |
| **Actors** | User, System |
| **Description** | User views their transaction history. |
| **Preconditions** | User is logged into the EzPay application. |
| **Normal Course** | 1. User navigates to "Transaction History" section. 2. System retrieves user's transaction data from the database. 3. System displays a list of transactions, including date, amount, recipient/sender, and status. 4. User can filter transactions by date range, type, or status. 5. User can select a specific transaction to view more details. |
| **Alternative Courses** | If there are no transactions, the system displays a message indicating no transaction history. |
| **Assumptions** |  |
| **Notes** |  |

User Story 3.1: Wireframe

##### User story 3.2: Track Transaction Status

|  |  |
| --- | --- |
| **Actors** | User, System |
| **Description** | User tracks the status of a specific transaction. |
| **Preconditions** | User is logged into the EzPay application. User has made at least one transaction. |
| **Normal Course** | User navigates to "Transaction History" section.  User selects a specific transaction.  System retrieves detailed status information for the selected transaction.  System displays the current status (e.g., initiated, processing, completed, failed) and any relevant details. |
| **Alternative Courses:** | If the transaction status cannot be retrieved, the system displays an error message. |
| **Postconditions** | User views the current status of their transaction. |

##### User Story 3.2: Wireframe

Cancel Scheduled Payment

Modify Scheduled Payment

#### Use Case 4: Notification System

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case ID:** | 4 |  |  |
| **Use Case Name:** | Notification System |  |  |
| **Created By:** | ChandraSekhar | **Last Updated By:** | ChandraSekhar |
| **Date Created:** |  | **Date Last Updated:** |  |

##### User story 4.1: Transaction Confirmations

|  |  |
| --- | --- |
| **Actors** | User, System |
| **Description** | System sends transaction confirmation to the user. |
| **Preconditions** | User has completed a transaction. User has enabled notifications. |
| **Normal Course** | 1. System processes a completed transaction. 2. System generates a confirmation message with transaction details. 3. System sends the confirmation to the user via push notification and email. 4. User receives and views the confirmation. |
| **Assumptions** |  |
| **Notes** |  |

##### User Story 4.1: Wireframe

Payment Progress

Payment Overview

Payment History

##### User story 4.2: Payment Reminders

|  |  |
| --- | --- |
| **Actors** | User, System |
| **Description** | System sends payment reminders to the user. |
| **Preconditions** | User has scheduled payments or recurring bills in the system. User has enabled reminders. |
| **Normal Course** | 1. System identifies upcoming payments based on user's schedule. 2. System generates reminder messages for payments due within the next 3 days. 3. System sends reminders to the user via push notification and email. 4. User receives and views the reminders. |
| **Assumptions** |  |
| **Notes** |  |

##### User Story 4.2: Wireframe

Payment Progress

Payment Overview

Payment History

#### Use Case 5: Security and Compliance

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case ID:** | 5 |  |  |
| **Use Case Name:** | Security and Compliance |  |  |
| **Created By:** | ChandraSekhar | **Last Updated By:** | ChandraSekhar |
| **Date Created:** |  | **Date Last Updated:** |  |

##### User story 5.1: Data Encryption

|  |  |
| --- | --- |
| **Actors** | System, Security Module |
| **Description** | System ensures all sensitive data is encrypted. |
| **Preconditions** | System has access to encryption algorithms and keys. |
| **Normal Course** | 1. User inputs sensitive data (e.g., payment information, personal details). 2. System identifies the sensitive data. |
|  | 1. Security Module encrypts the data using appropriate encryption algorithms. 2. System stores or transmits the encrypted data. 3. When data needs to be accessed, Security Module decrypts it for authorized use. |
| **Assumptions** |  |
| **Notes** |  |

##### User Story 5.1: Wireframe

Payment Progress

Payment Overview

Payment History

##### User story 5.2: Fraud Prevention

|  |  |
| --- | --- |
| **Actors** | System, Fraud Detection Module |
| **Description** | System detects and prevents fraudulent activities. |
| **Preconditions** | Fraud Detection Module is active and updated with latest fraud patterns. |
| **Normal Course** | 1. User initiates a transaction or account activity. 2. System passes the activity details to the Fraud Detection Module. 3. Fraud Detection Module analyzes the activity against known fraud patterns. 4. If no fraud is detected, the activity proceeds normally. 5. If potential fraud is detected, the system blocks the activity and alerts the user. |
| **Alternative Courses** | For high-risk activities, system may require additional verification from the user. |
| **Notes** |  |

##### User Story 5.2: Wireframe

Payment Progress

Payment Overview

Payment History

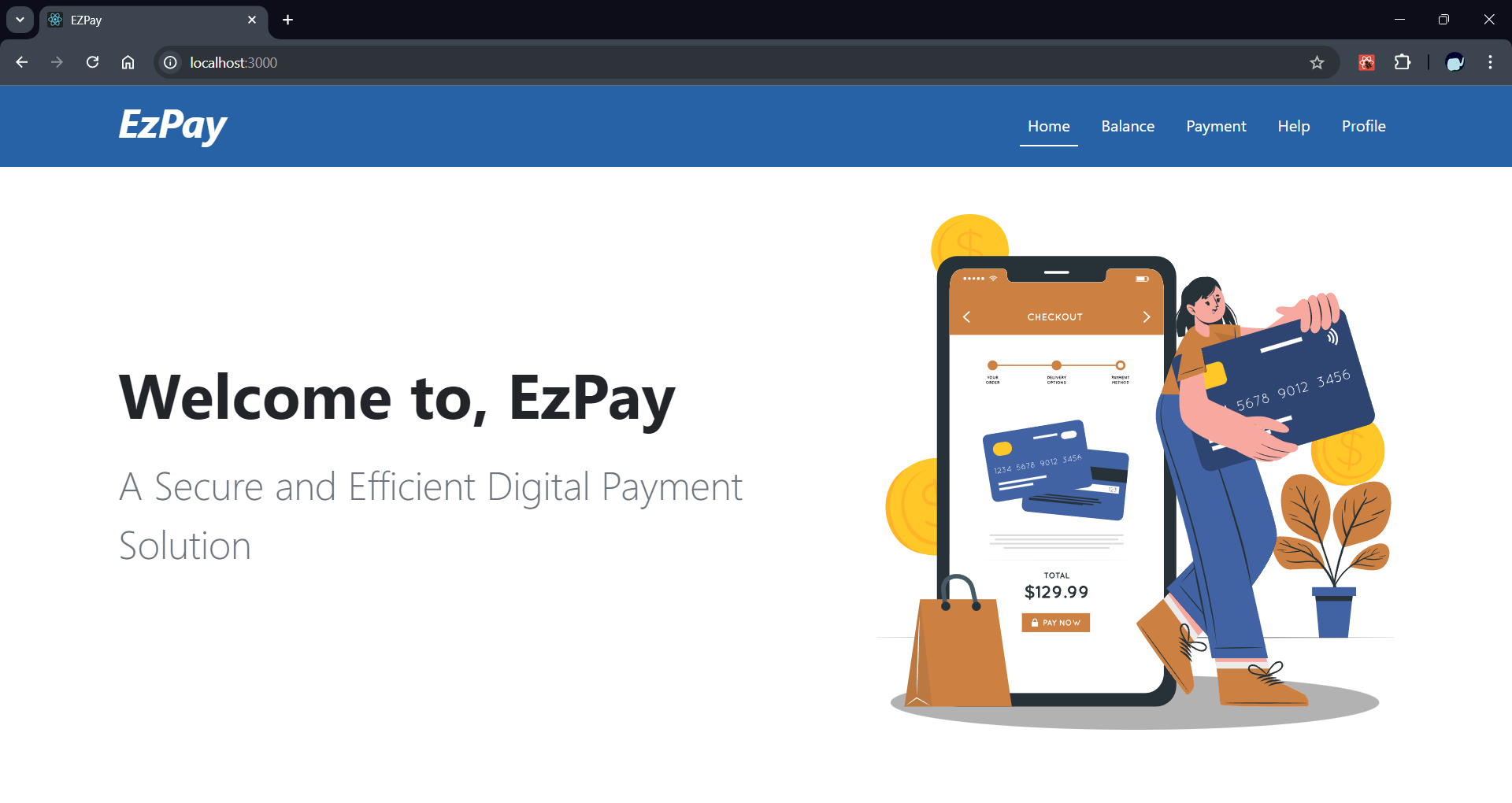
#### Use Case 6: User Interface

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case ID:** | 6 |  |  |
| **Use Case Name:** | User Interface |  |  |
| **Created By:** | ChandraSekhar | **Last Updated By:** | ChandraSekhar |
| **Date Created:** |  | **Date Last Updated:** |  |

##### User story 6.1: Responsive Design

|  |  |
| --- | --- |
| **Actors** | User, System |
| **Description** | User accesses the application on different devices. |
| **Preconditions** | ser has access to devices with different screen sizes (e.g., smartphone, tablet, desktop). |
| **Normal Course** | 1. User accesses the EzPay application on a device. 2. System detects the device type and screen size. 3. System adjusts the layout and UI elements to fit the screen size. 4. User interacts with the responsive interface. |
| **Assumptions** |  |
| **Notes** |  |

##### User Story 6.1: Wireframe



The landing page :

User Dashboard - Balance Page

A screenshot of a computer

Description automatically generated

User Dashboard - Payment Page

A screenshot of a computer

Description automatically generated

The Help page displays all the tickets raised by the current user. For each ticket, users have three options: resolving the ticket, chatting with a ChatBot for assistance, or deleting the ticket.

A screenshot of a computer

Description automatically generated

On clicking the ‘Create Ticket’ Button, a modal will open up pre-filled with the current user id. Once the user enters their complaint, it will be added to the list.

A screenshot of a computer

Description automatically generated

On clicking the chat button, the Support Chat modal will open up. The user can then chat with the ChatBot Assisstant.

A screenshot of a computer

Description automatically generated

The filter option allows the user to sort the tickets based on its status.

A screenshot of a computer

Description automatically generated

If the userId is invalid, an error message will be displayed.

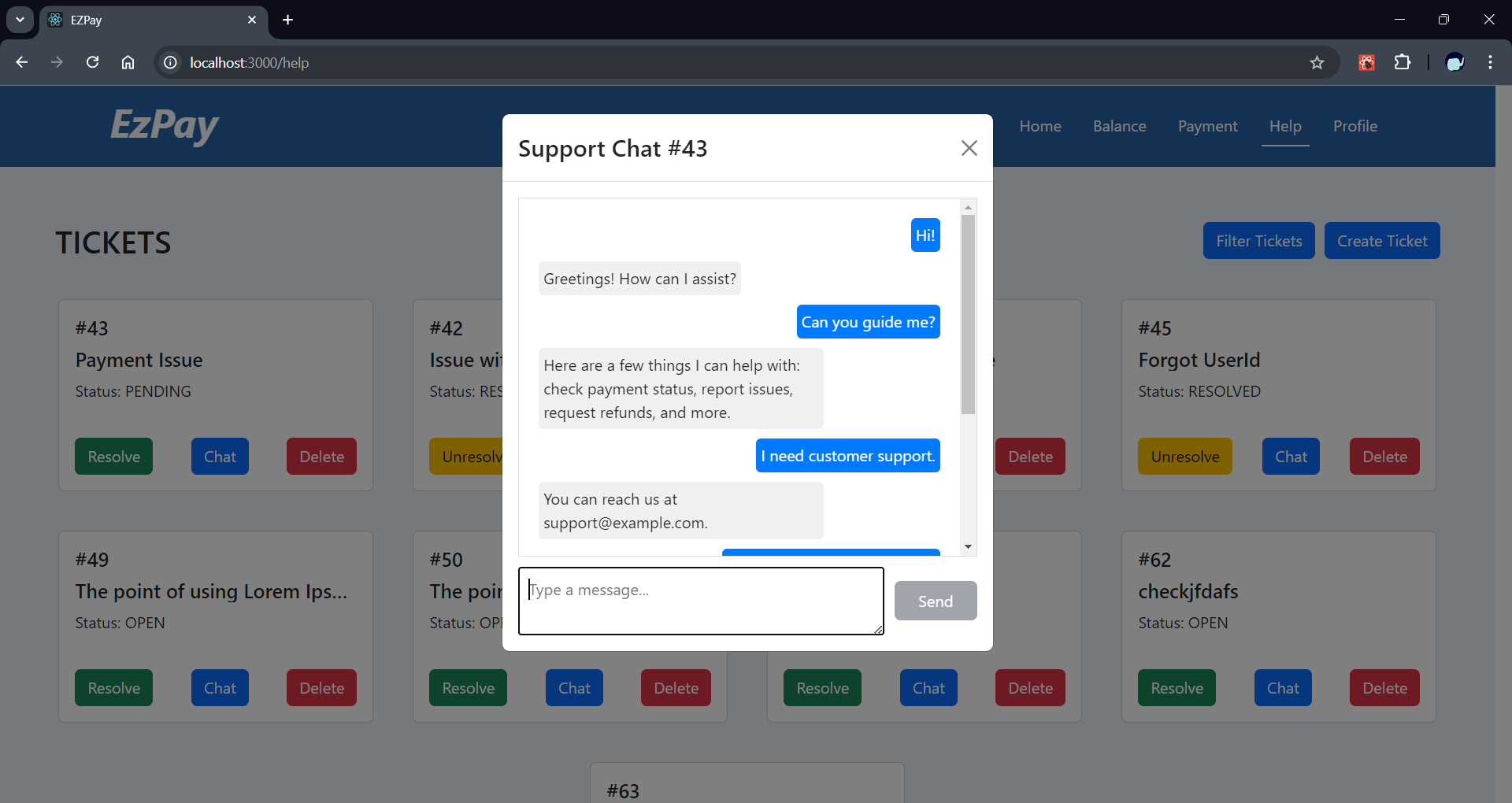
A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

Chat Modal on clicking the chat button. This allows the user to send and receive messages from the AI Chatbot



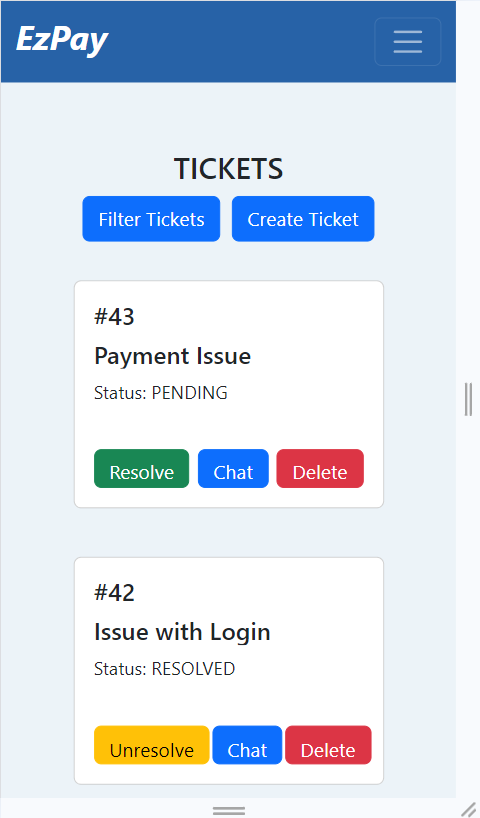
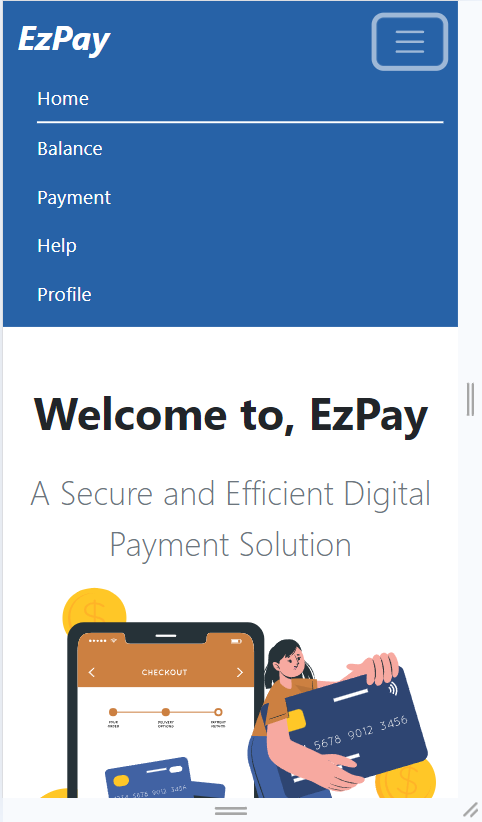
A screenshot of a computer

Description automatically generated

A screenshot of a chat

Description automatically generated

A person holding a credit card

Description automatically generatedA screenshot of a ticket form

Description automatically generated

Mobile View of the App

A screenshot of a support ticket

Description automatically generatedA screenshot of a support ticket

Description automatically generatedA screenshot of a phone

Description automatically generated

A screenshot of a ticket

Description automatically generatedA screenshot of a computer

Description automatically generated

A screenshot of a phone

Description automatically generatedA screenshot of a phone

Description automatically generated

A screenshot of a phone

Description automatically generatedA screenshot of a phone

Description automatically generated

A screenshot of a login page

Description automatically generatedA screenshot of a login page

Description automatically generated

A screenshot of a computer screen

Description automatically generatedA screenshot of a login page

Description automatically generated

A screenshot of a message box

Description automatically generated

A person holding a large cellphone

Description automatically generatedA screenshot of a social media account

Description automatically generatedA screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generated

##### User story 6.2: Easy Navigation

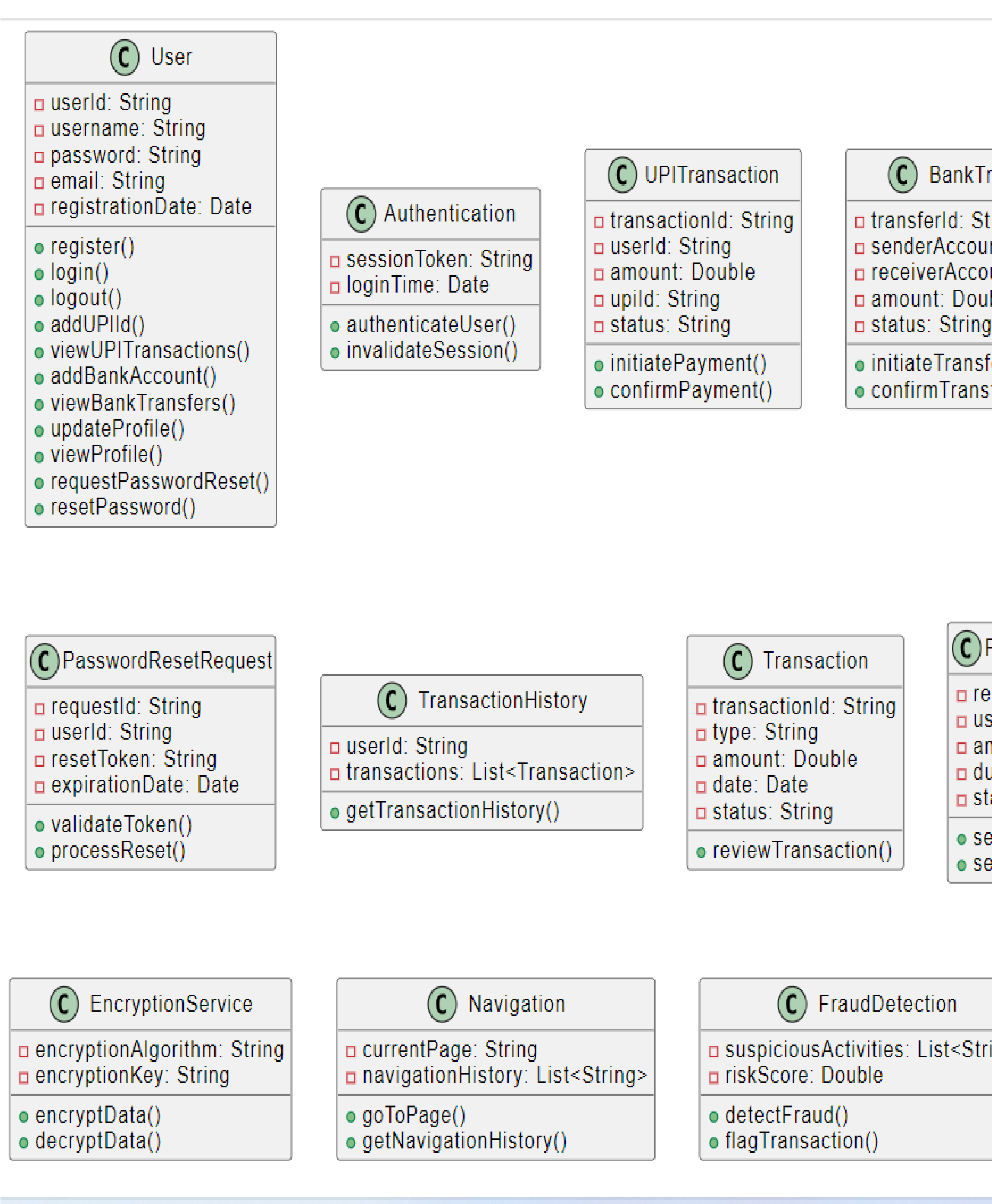
|  |  |  |
| --- | --- | --- |
| **Actors** | User, System | |
| **Description** | User navigates through the application intuitively. | |
| **Preconditions** | User is logged into the EzPay application. | |
| **Normal Course** | 1. | User accesses the main dashboard of the application. |
|  | 2. | System displays clear menu options and quick action buttons. |
|  | 3. | User selects desired function (e.g., make payment, view history). |
|  | 4. | System navigates to the selected function seamlessly. |
|  | 5. | User can easily return to previous screens or the main dashboard. |
| **Alternative Courses** | 1. | System provides a search function for users to quickly find specific features. |
|  | 2. | System offers contextual help or tooltips for complex features |
| **Notes** |  |  |

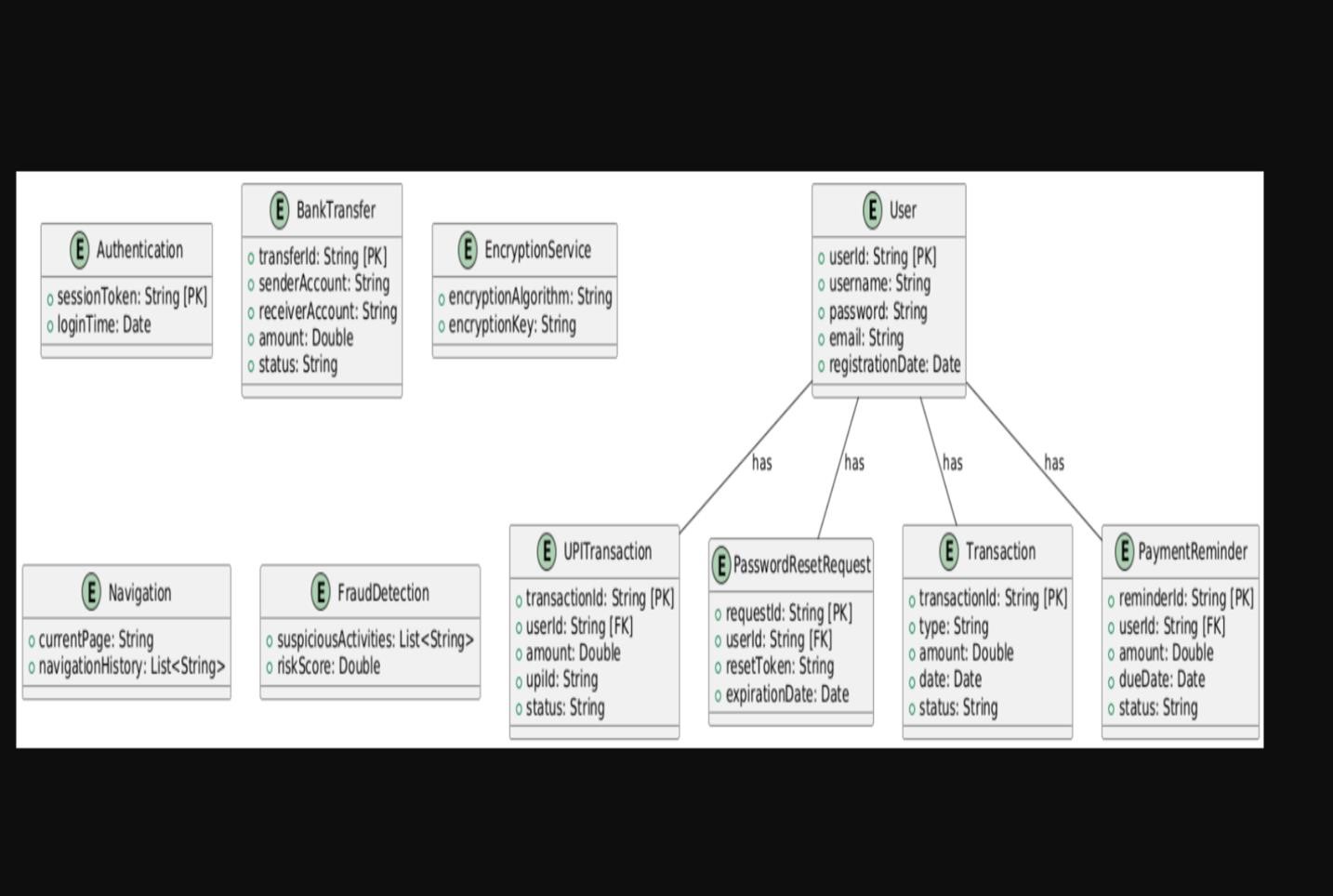
##### User Story 6.2: Wireframe

Reference Links:

Project Diagram

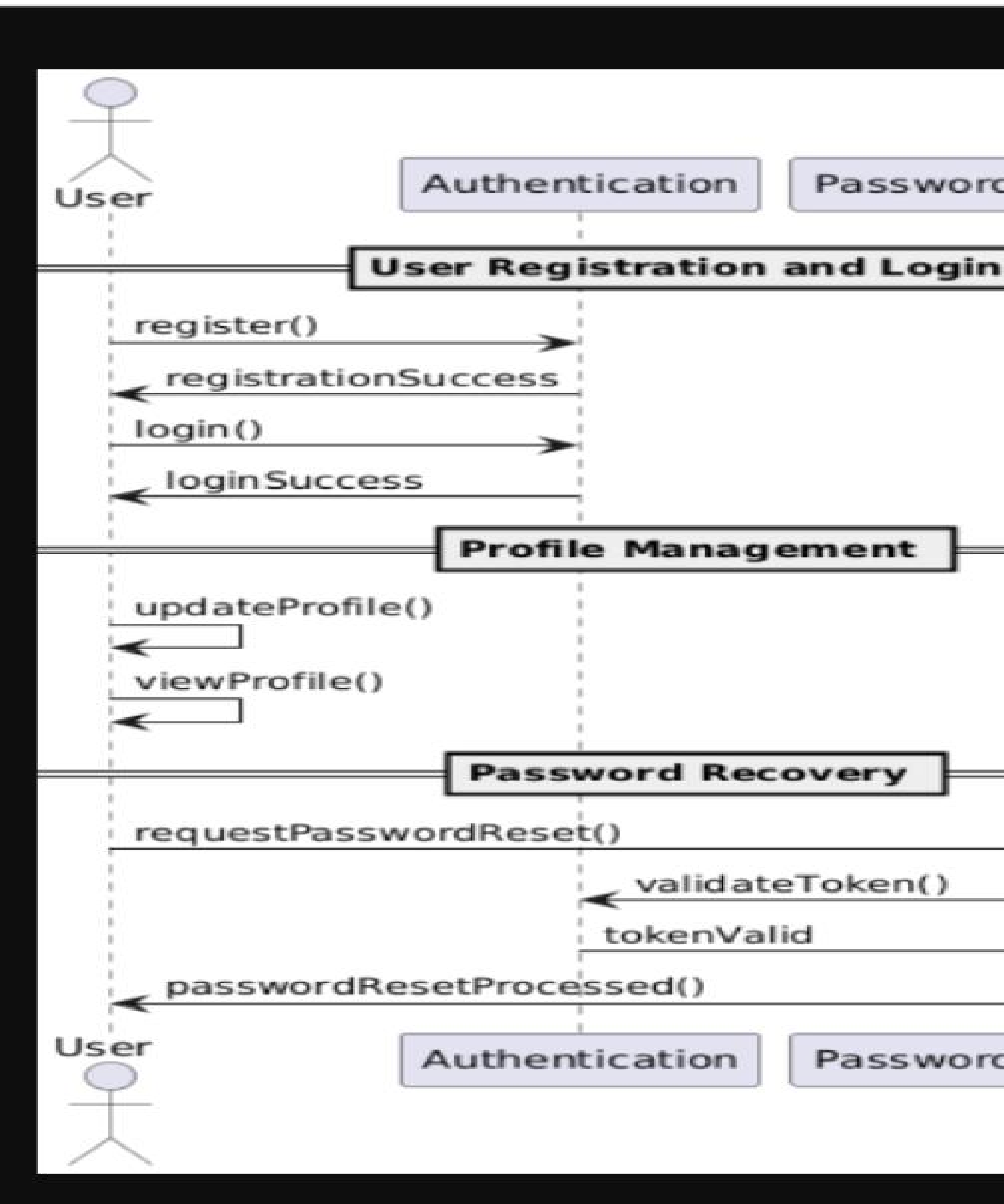
## Class Diagram

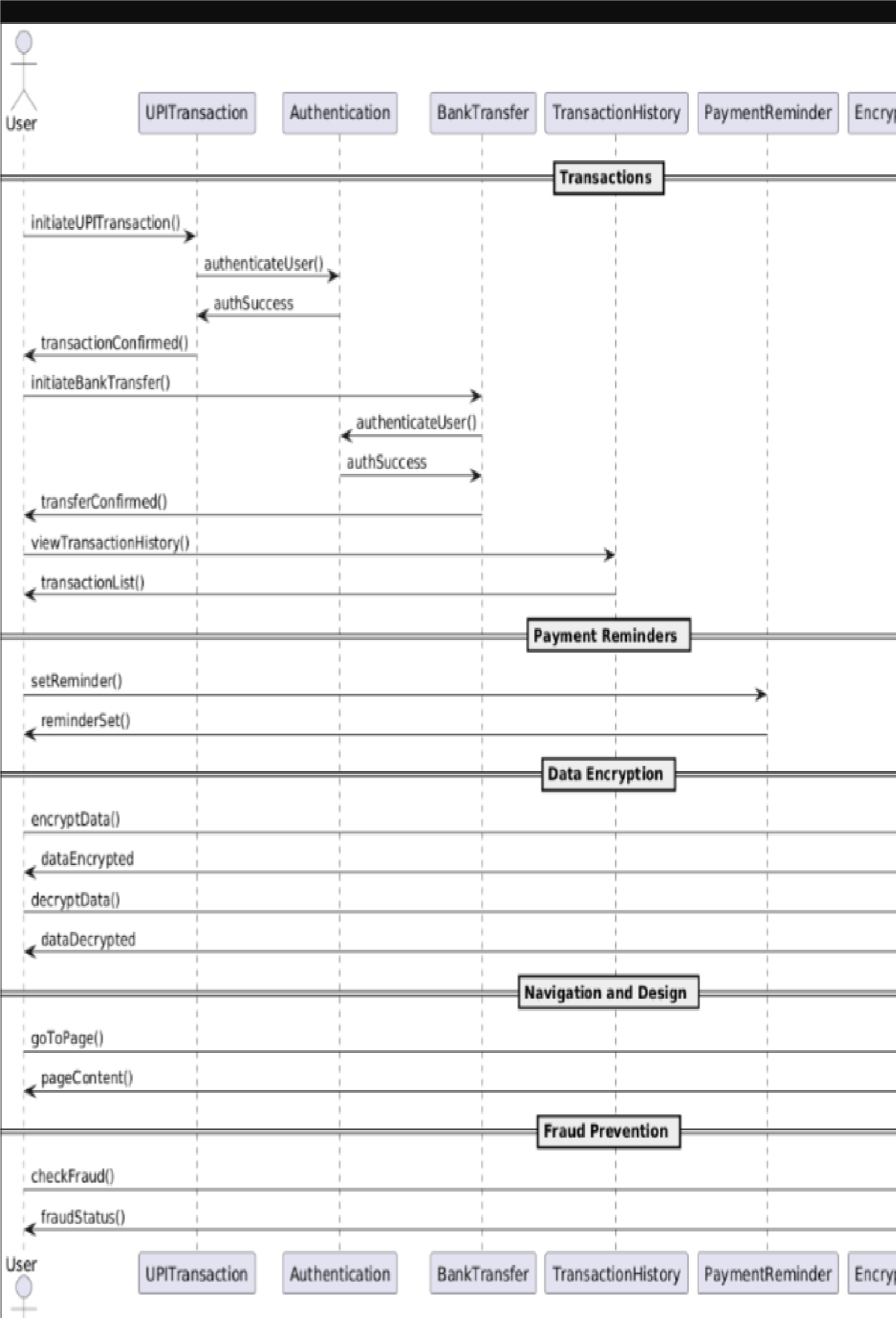
ER Diagram



•

Sequence Diagram





Architecture Diagram

