
Software Requirements Specification

for

Banking Web App

Version 1.0 approved

Prepared by:

Ifra Ejaz (21L-7508)

Abdullah Awan (21L-7713)

Syed Ali Hassan (21L-5274)

Sultan Sheikh (21L-5170)

31st March, 2024.

Table of Contents

Table of Contents	ii
Revision History	ii
1. Introduction.....	1
1.1 Purpose.....	1
1.2 Document Conventions	1
1.3 Intended Audience and Reading Suggestions	1
1.4 Product Scope	2
2. Overall Description.....	2
2.1 Product Perspective.....	2
2.2 Product Functions	2
2.3 User Classes and Characteristics	2
2.4 Operating Environment.....	6
2.5 Design and Implementation Constraints	7
3. External Interface Requirements.....	8
3.1 User Interfaces	8
3.2 Hardware Interfaces	13
3.3 Software Interfaces.....	14
4. System Features.....	15
4.1 System Feature 1.....	Error! Bookmark not defined.
4.2 System Feature 2	
4.3 System Feature 3.....	
4.4 System Feature 4.....	
4.5 System Feature 5.....	
4.6 System Feature 6.....	
4.7 System Feature 7.....	
4.8 System Feature 8.....	
4.9 System Feature 9.....	
5. Other Nonfunctional Requirements.....	17
5.1 Performance Requirements	25
5.2 Security Requirements.....	25
5.3 Software Quality Attributes	25
Appendix A: Glossary	Error! Bookmark not defined.
Appendix B: Analysis Models.....	26

Revision History

Name	Date	Reason For Changes	Version
Ifra Ejaz, Abdullah Awan, Syed Ali Hassan, Sultan Sheikh	31 st March, 2024	Initial Draft Created	1.0

1. Introduction

1.1 Purpose

The purpose of this document is to provide a comprehensive specification for the Banking Web Application. This includes detailing the software's functional and non-functional requirements to guide its development and ensure it meets the needs of its users.

1.2 Document Conventions

- **Use cases identifiers:** Each requirement is prefixed with a unique identifier (e.g., UC-01 for the first login requirement), facilitating easy reference and traceability.
- **Glossary references:** Terms that are defined in the Glossary section are marked with a number in square braces. Readers are encouraged to consult the Glossary for definitions and clarifications of technical terms and acronyms used in this document.

1.3 Intended Audience and Reading Suggestions

1.3.1 Intended Audience

- **Developers:** Implement the requirements specified in this document. Developers should pay special attention to the detailed requirements sections for each feature, as well as the "System Architecture"
- **Project Managers:** Individuals overseeing the project's progress and ensuring that deliverables meet the stipulated requirements.
- **Quality Assurance Testers:** Professionals who will create and execute test cases based on the requirements to ensure the application's functionality and performance. QA testers should focus on the "Functional Requirements" section

1.3.2 Reading Suggestions

- All readers are encouraged to begin with the "Introduction" section to gain context and understanding of the application's purpose, scope, and definitions.
- Proceed to the "Overall Description" for a high-level view of the application, including its functionality, user needs, and constraints.
- Based on the reader's role, follow the suggested sections above to focus on areas most relevant to your needs.
- The "Glossary" section at the end of the document provides definitions of technical terms and acronyms used throughout the SRS, which may be helpful for all readers.

1.4 Product Scope

The scope of the banking web application encompasses the development and deployment of a secure, intuitive, and comprehensive online platform that allows users to conduct a variety of financial transactions and account management activities. The application will cater to individual users seeking to manage their personal banking needs efficiently.

2. Overall Description

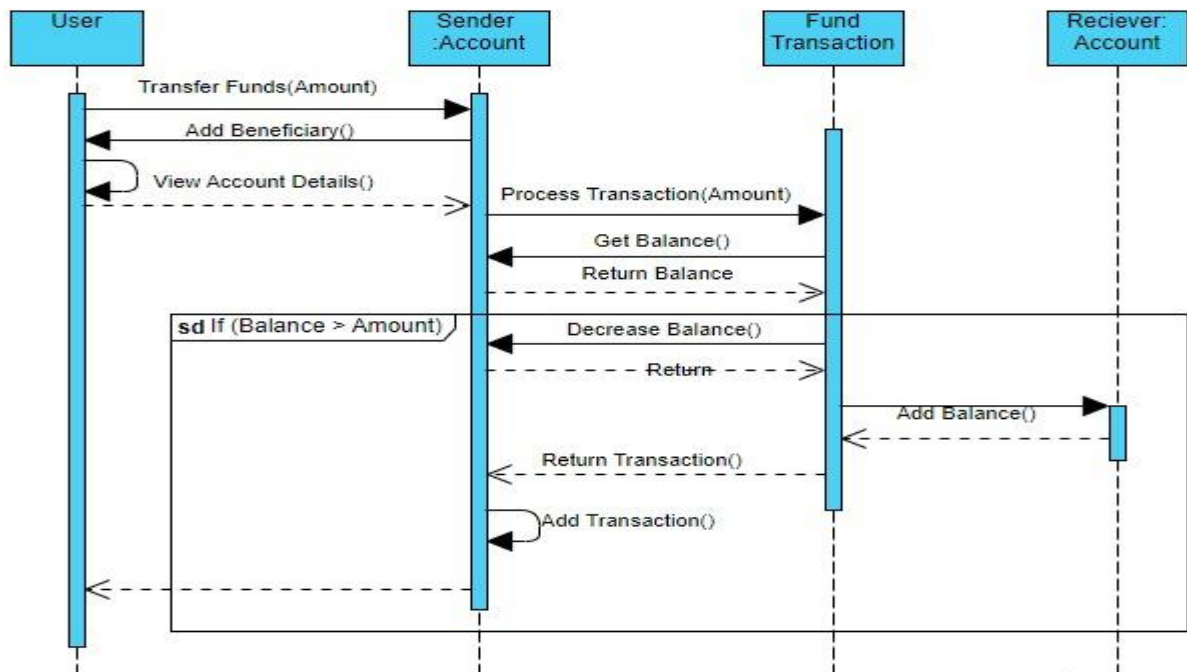
2.1 Product Perspective

This product is conceived as a new, self-contained entity, designed to meet specific needs within its target domain. It is not a continuation of or a replacement for any existing system, nor is it a member of a pre-existing product family.

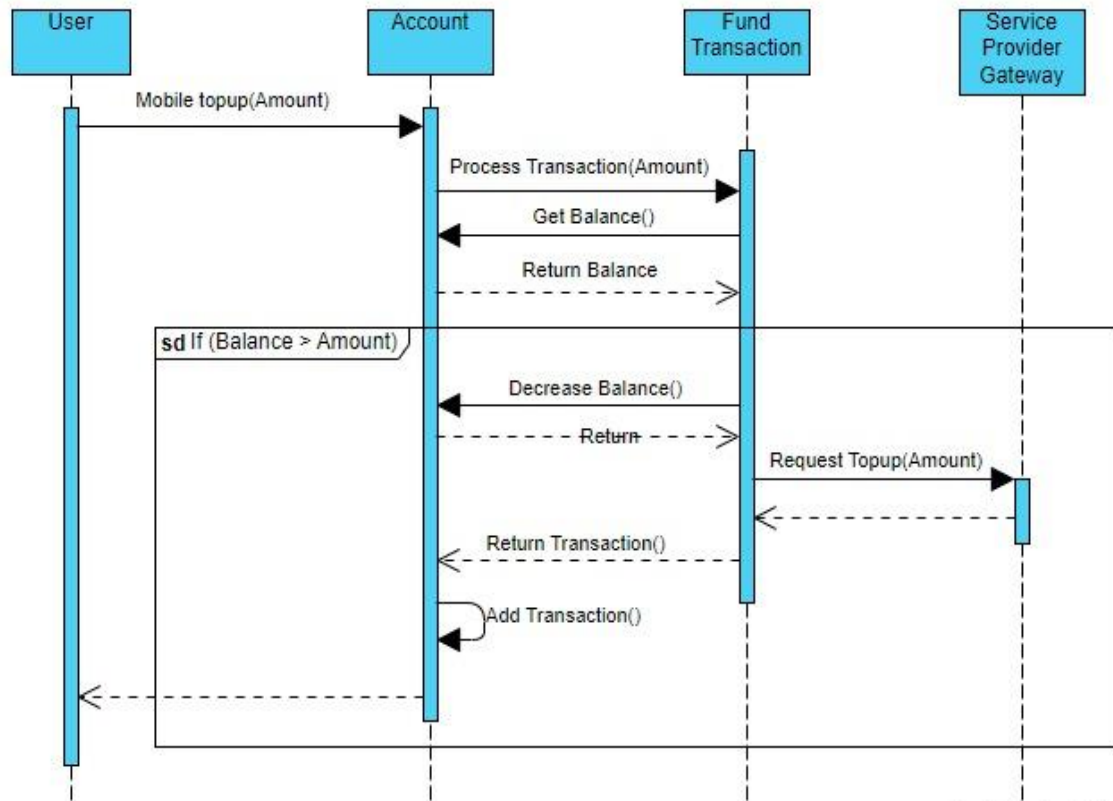
2.2 Product Functions

The major functions that will be performed by this application are described below along with sequence of flow of some of them:

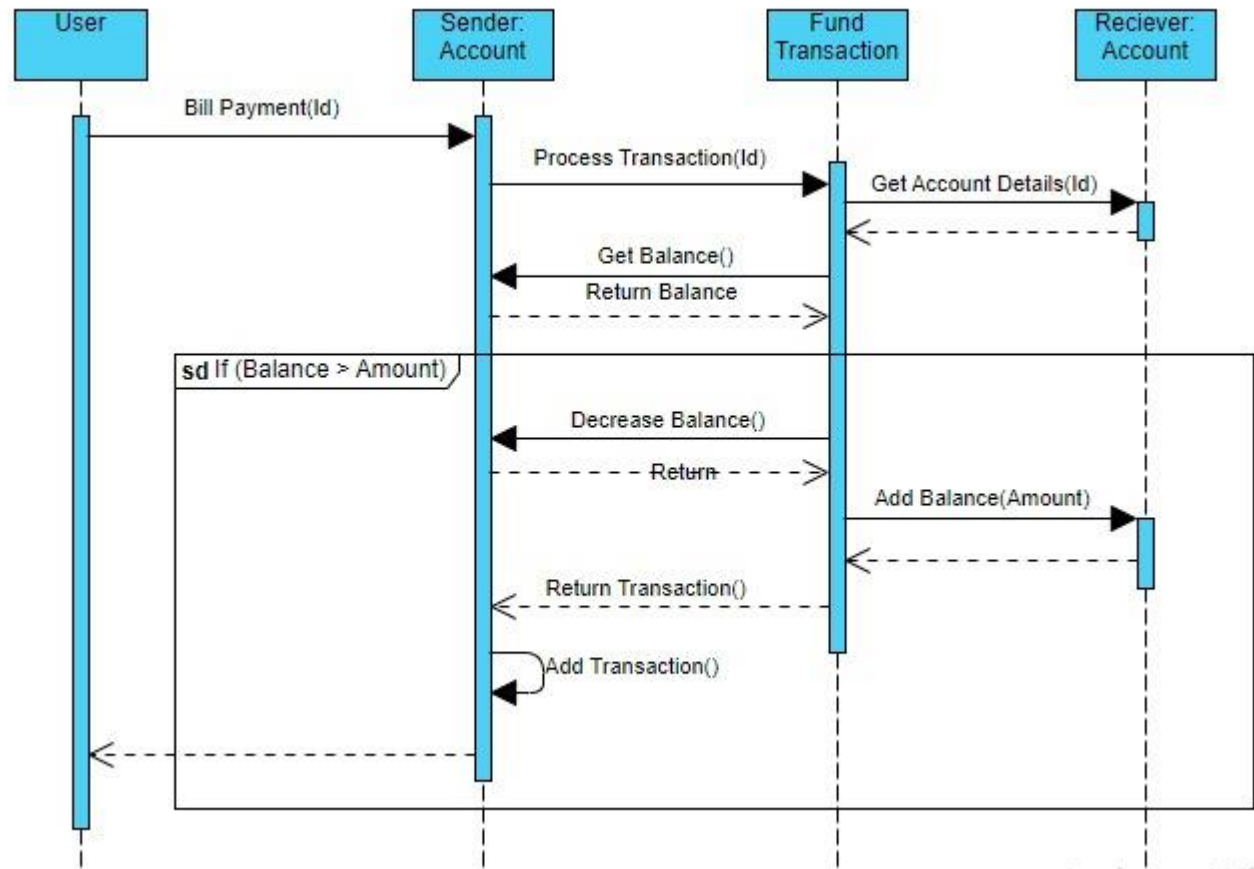
- **Transfer Funds**



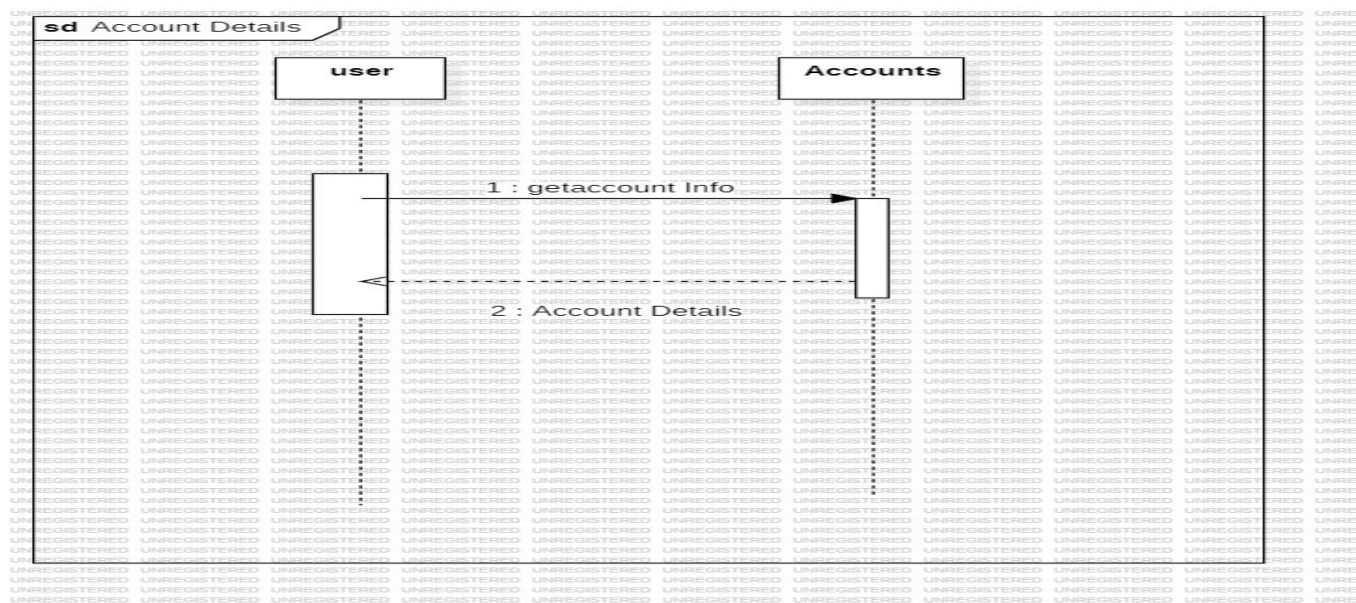
- **Mobile Top-Up:**



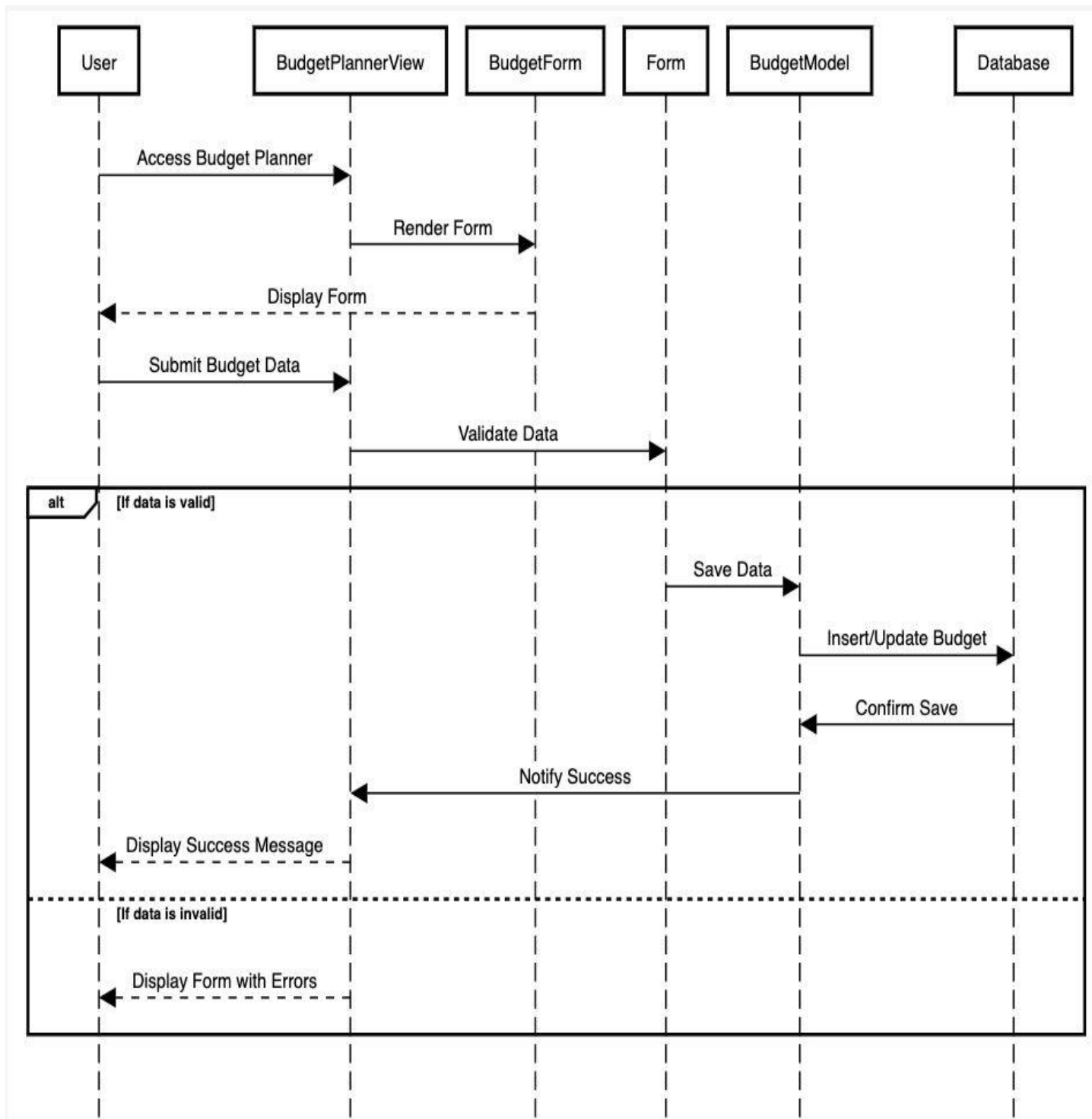
• Bill Payments:



• Get account Details:



- Plan Budget:



2.3 User Classes and Characteristics

2.3.1 User Classes

- **Regular Users:** This class consists of individuals who will use the application on a daily basis. They require a user-friendly interface with streamlined workflows to perform routine tasks efficiently.
- **Technical Users:** Users with a higher level of technical expertise, such as IT professionals or developers, who may interact with the application to configure settings, manage integrations, or utilize advanced features.
- **Administrative Users:** Individuals responsible for managing the application at an organizational level. This includes overseeing user access, ensuring data security, and generating reports.

2.3.2 Characteristics

- **Frequency of Use:** Regular users demand an efficient and streamlined experience.
- **Subset of Product Functions Used:** Technical users may delve into advanced features, whereas regular users might only use a core set of functionalities.
- **Technical Expertise:** Varies from minimal (regular and occasional users) to high (technical users).
- **Security or Privilege Levels:** Administrative users have higher access levels for managing the system, unlike other users who have restricted access based on their needs.

2.3.3 Prioritization

While all user classes are important, **regular users** and **administrative users** are often considered most critical for the success of the product. Regular users are the backbone of daily operations and user engagement, while administrative users ensure the smooth functioning and security of the application. However, the importance of each user class may vary based on the specific context and objectives of the product.

2.4 Operating Environment

Hardware Platform:

This software is particularly made as a web application designed for any operating system. The banking web application is designed to be highly scalable and performant, supporting a wide range of devices including desktop computers and mobile phones.

Operating System and Versions

The application will be compatible with the following operating systems, ensuring broad accessibility for users:

- Windows: Support for Windows 10 and newer.
- macOS: Support for macOS .
- iOS: Support for iOS .
- Android: Support for Android.

Web Browsers

Given the web-based nature of the application, compatibility with major web browsers is essential. The application will support the latest versions of:

- Google Chrome
- Mozilla Firefox
- Safari
- Microsoft Edge

2.5 Design and Implementation Constraints

2.5.1 Hardware Limitations

- The application must be optimized for performance on hardware with limited processing capabilities or memory, ensuring responsiveness across a wide range of devices.
- The application should be designed to perform well under various network conditions, especially in regions with poor connectivity.

2.5.2 Interfaces to other Applications

Integration with banking APIs[1], payment gateways, and identity verification services, which may limit choice or design based on the external APIs'[1] capabilities and constraints.

2.5.3 Specific Tools and Technologies

Constraints on the use of certain IDEs, version control systems, or deployment tools based on corporate licenses or preferences.

2.5.4 Parallel Operations

The application must support multiple users performing transactions simultaneously without data inconsistency or performance degradation, which can limit architectural choices.

2.5.5 Security Considerations

- **Authentication and Authorization:** The need for robust security measures, such as two-factor authentication or biometric verification, can influence the user authentication flow and data access controls.
- **Data Encryption:** Requirements for encrypting data at rest and in transit to protect sensitive user information.

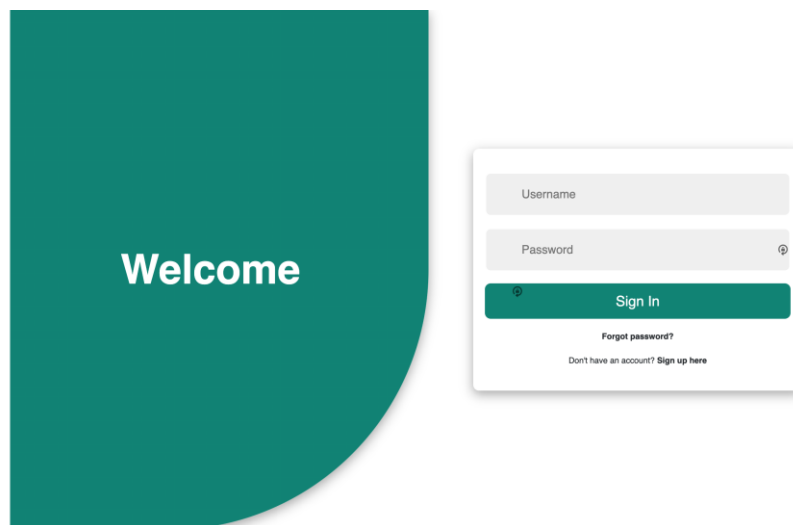
3. External Interface Requirements

3.1 User Interfaces

In this section, we delve into the external interface requirements, focusing specifically on user interfaces (UIs). Additionally, we will detail the conventions for standard buttons, functions such as help features, keyboard shortcuts, and the presentation of error messages, all of which are integral to fostering a seamless and efficient user interaction.

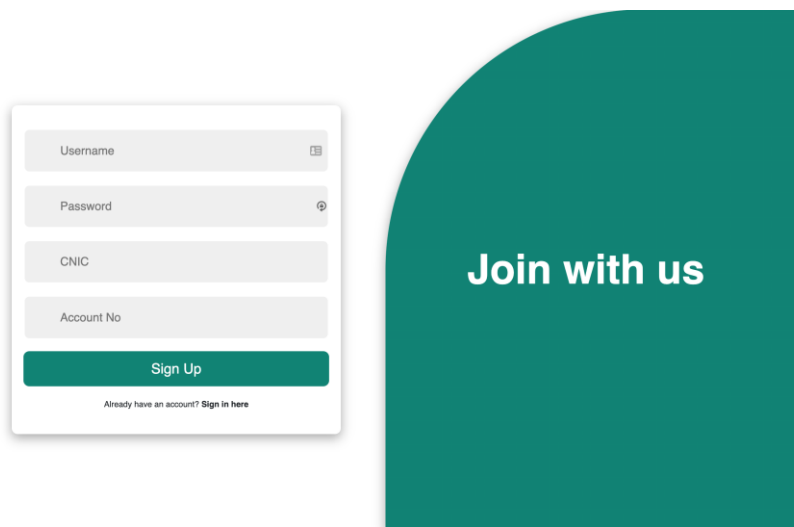
3.1.1 Login Interface

- **Appearance:** A clean and modern design with a welcoming "Welcome" message.
- **Functionality:** Allows existing users to enter their username and password to log in.
- **Navigation:** Offers a link for users who have forgotten their password and an option to sign up for new users.
- **Constraints:** The username and password fields must adhere to predefined security standards.
- **Standard Elements:** A 'Sign In' button, links for password recovery, and a redirect to the signup page.
- **Error Handling:** Displays error messages in a discreet but clear manner when login fails.



3.1.2 Signup Interface

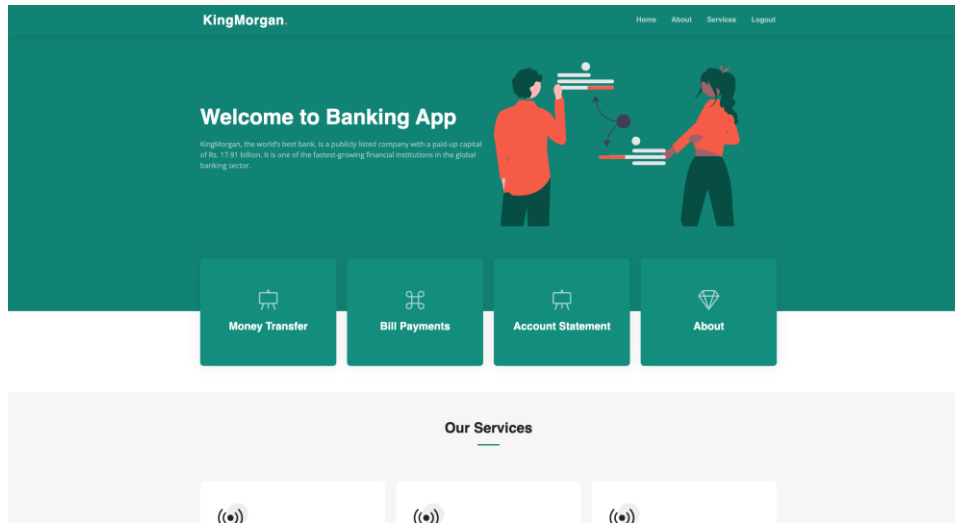
- **Appearance:** Consistent with the login page, with a "Join with us" heading, promoting a seamless user experience.
- **Functionality:** New users can create an account by entering a username, password, CNIC, and account number.
- **Navigation:** Includes a link back to the sign-in page for existing users.
- **Constraints:** All fields are required and must validate against specific format requirements.
- **Standard Elements:** A 'Sign Up' button and a link to switch to the login page for existing users.



3.1.3 Dashboard Interface

The dashboard is the main interface of our web application which will allow users to switch between different interfaces.

- **Appearance:** A vibrant and inviting layout with a quick overview of the banking app's features.
- **Functionality:** Directs the user to various banking services like Money Transfer, Bill Payments, and Account Statements.
- **Navigation:** A header with navigation links to Home, About, Services, and Logout.
- **Constraints:** The dashboard is structured to highlight essential services and information, maintaining a balance between functionality and simplicity.
- **Standard Elements:** Service icons, a logout link, and additional resource links in the footer.



3.1.4 Bill Payment Interface

- **Functionality:** Allows users to view and select bill payment options.
- **Layout:** Bill payment options are displayed in a grid format with icons.
- **Navigation:** Consistent navigation to 'Our Services' and other sections as per the dashboard layout.
- **Styling:** Utilizes the site's color scheme and branding, with iconography for each bill payment option.

3.1.5 Fund Transfer Interface

- **Functionality:** Users can select a bank, input account number, and amount for money transfer.
- **Form Elements:** Dropdown for bank selection, input fields for account number and amount, and a submit button for initiating the transfer.
- **Navigation:** Consistent navigation to 'Our Services' and other sections as per the dashboard layout.
- **Feedback:** Displays messages for successful or unsuccessful transfers.
- **Styling:** A clean form design that is easy to read and interact with, utilizing branded colors.

3.1.6 Card Detail Interface

- **Functionality:** Displays card details such as card type, number, and expiry date.
- **Interactivity:** Includes a toggle switch to activate or deactivate cards.
- **Layout:** Card details are presented in a card-like format, mirroring the physical appearance of a bank card.

- **Navigation:** Consistent navigation to 'Our Services' and other sections as per the dashboard layout.
- **Styling:** The card layout design is modern, with colors that reflect the bank's branding.

3.1.7 Account Information Interface

- **Functionality:** Provides detailed information on the user's account including name, email, phone number, balance, account type, card number, and card expiry date.
- **Layout:** Information is presented in a list format with alternating background colors for readability.
- **Navigation:** Consistent navigation to 'Our Services' and other sections as per the dashboard layout.
- **Styling:** Adheres to a minimalistic style with a focus on clarity and ease of use

3.1.8 Mobile Top Up Interface

- **Functionality:** Enables users to add credit to mobile phone accounts.
- **Fields:** Mobile number entry, carrier selection, top-up amount.
- **Buttons:** 'Top Up Now' to initiate the transaction.
- **Navigation:** Consistent navigation to 'Our Services' and other sections as per the dashboard layout.
- **Feedback:** Confirmation message upon successful top-up or error message for transaction failure.
- **Styling:** In line with the application's theme, possibly utilizing mobile-related icons and a straightforward form design

3.1.9 Transaction Detail Interface

- **Functionality:** Presents detailed information about past financial transactions.
- **Layout:** A list or table view displaying date, transaction type, amount, recipient/payee, and status.
- **Navigation:** Links or buttons to filter transactions by type, date range, or amount.
- **Interactivity:** Clickable transaction items to expand and view more details.
- **Styling:** A clean, readable design with alternating row colors for ease of reading, and icons indicating the type of transaction

3.1.10 About Us Interface

- **Content:** Includes detailed information about the bank's values, history, and commitment to clients.

- **Multimedia:** Features images and video links to provide a rich user experience.
- **Layout:** Sections for 'About Us', 'Clients', 'Call to Action', 'Stats Counter', 'Testimonials', and 'Contact'.
- **Navigation:** Consistent navigation to 'Our Services' and other sections as per the dashboard layout.
- **Styling:** A mix of textual information and multimedia elements with a professional and engaging design.

GUI Standards and Style Guides:

- Follows the bank's brand guidelines with specific color palettes, typography, and imagery.
- Ensures consistency across all screens for elements like buttons, input fields, and navigational components.
- Adheres to responsive design principles, ensuring compatibility across a range of devices.

Error Message Display Standards:

- All error messages are displayed in a consistent format, using a color that stands out against the background for visibility.
- Provides clear instructions on how to resolve the issue indicated by the error.

Common Elements Across All Interfaces:

- **Navigation:** A consistent header and footer are present across all pages, providing a uniform user experience.
- **Accessibility:** Features are designed to be accessible, with clear labels and easy navigation.
- **Responsive Design:** All interfaces are built to be responsive across devices, maintaining usability and design integrity.

Standard buttons and functions:

- A help button on all pages
- A logout button accessible from every page once logged in
- Standardized back navigation on the dashboard

Screen layout constraints could include:

- Fixed-width side navigation on the dashboard
- Limiting the length of input fields to prevent layout issues

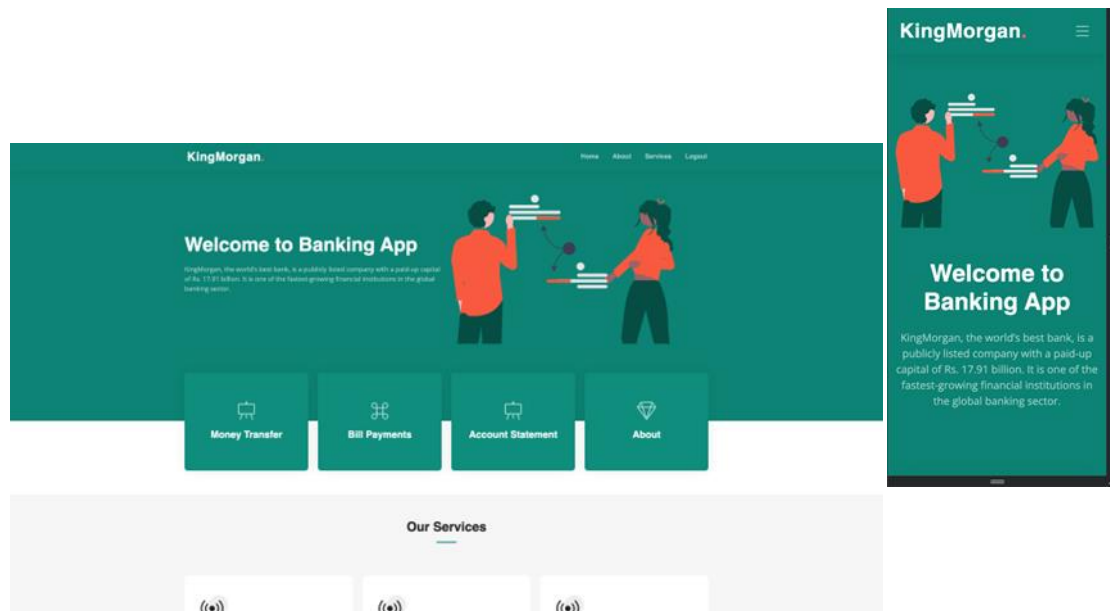
Keyboard shortcuts might involve:

- Alt + S to submit forms
- Alt + H to return home

3.2 Hardware Interfaces

Supported Device Types:

- **Desktops and Laptops:** The banking application is accessible on Windows, macOS, and Linux operating systems with modern web browsers such as Chrome, Firefox, Safari, and Edge.
- **Mobile Devices:** Fully responsive design supports iOS and Android devices, ensuring full functionality via mobile web browsers.
- **Tablets:** The application is tablet-friendly, designed to work seamlessly on iPads, Android tablets, and Windows tablets with appropriate scaling and touch interactions.



Data and Control Interactions:

- **Input Devices:** The application accepts input from standard devices, including keyboards, mice, touchpads, and touchscreens.
- **Output Devices:** Visual output is displayed through monitors, laptop screens, and mobile device screens, with compatibility for varying resolutions and screen sizes. Audio output for alerts and notifications is supported.
- **Card Readers:** For in-branch digital services, the application interfaces with card reader hardware to read debit/credit card data during transactions.
- **Printers:** Integration with standard printers for printing transaction receipts, statements, and other banking documents directly from the application.

Communication Protocols:

- **HTTP/HTTPS:** For secure web traffic, ensuring encrypted data transmission over the internet.

3.3 Software Interfaces

Database:

- **Management System:** SQLITE
- **Purpose:** Stores all user data, transaction records, account details, and other relevant banking information.

Operating System:

- **Supported Systems:** Windows 10 and above, macOS & Linux.
- **Purpose:** Provides the necessary environment for the application to run.

Backend Tools:

- **Framework:** Django
- **Purpose:** Serves as the backend framework handling business logic, database interactions, and server-side rendering.

Frontend Libraries:

- **Languages:** HTML5, CSS3, JavaScript
- **Purpose:** Facilitates dynamic content presentation and user interaction on the client side.

4. System Features

The major features of this product are listed here along with the use case description of each feature:

4.1 Sign-Up

The sign-up feature allows new users to create an account with the banking application. It's the first interaction a potential user has with the application, making it crucial for establishing trust and ensuring a smooth onboarding process.

Use Case ID	UC_01	
Use Case Name	Sign-Up	
Description	Allows new users to create an account with the banking web app to access its services.	
Primary Actor	User	
Secondary Actor	System	
Pre-Condition	User does not have an existing account.	
Post-Condition	A new user account is created, and the user can log in to the account.	
Basic Flow	User Action	System Action
	1. User selects the sign-up option and enters the required information for creating the account, such as name, email, and password.	2. The system validates the entered information, creates a new account, stores the data in the database, and confirms account creation to the user.
Alternate Flow	1b. User enters an email that is already associated with an existing account 2b. The system displays an error message indicating the issue and requests the user to either sign in or use a different email to create a new account.	

4.2 Sign-In

The sign-in feature enables existing users to securely access their accounts by entering their credentials, typically an email address or username and a password. This process is critical for authenticating users and protecting their account information from unauthorized access.

Use Case ID	UC_02	
Use Case Name	Sign-In	
Description	Allows the user to access their account by entering credentials.	
Primary Actor	User	
Secondary Actor	System	
Pre-Condition	User must have an existing account with a username and password.	
Post-Condition	User gains access to their banking dashboard.	
Basic Flow	User Action	System Action
	1. User selects the sign-in option and enters their username and password	2. System validates the credentials and grants access to the user's account.
Alternate Flow	1b. User enters incorrect credentials. 2b. System displays an error message and prompts for re-entry of credentials.	

4.3 Pay Bills

The "Pay Bill" feature enables users to make payments to registered billers or add new ones, such as utility providers, credit card companies etc.

Use Case ID	UC_03	
Use Case Name	Pay Bill	
Description	Enables the user to pay various bills online through the banking app.	
Primary Actor	User	
Secondary Actor	System	
Pre-Conditions	<ol style="list-style-type: none"> 1. User must be signed in 2. User must have a valid payment method configured. 3. User must have sufficient amount in his account. 	
Post-Condition	The bill is paid, and the transaction is recorded.	
Basic Flow	User Action	System Action
	1. User selects the pay bill option and selects the biller.	2. System asks the user to enter details regarding account number, payment method and amount.
	3. The user enters the billing details	4. The system validates the entered details and asks the user to confirm the transaction.
	5. The user confirms the details and authorizes the payment.	The system processes the payment and updates the user's account balance and transaction history.
	The user is notified that the payment was successful and is presented with the transaction receipt.	

Alternate Flow	4b. User does not have enough money to pay bill. 6b. System displays an error message.	

4.4 Transfer Funds

The feature enables users to initiate one-time transfers, set up recurring transfers, and manage future-dated transfers. It supports intra-bank (between accounts within the same bank) and inter-bank (to accounts in different banks) transactions.

Use Case ID	UC_04	
Use Case Name	Transfer Funds	
Description	Permits the user to transfer funds to their own or another person's account.	
Primary Actor	User	
Secondary Actor	System	
Pre-Conditions	User must be signed in and have a valid account to transfer from.	
Post-Condition	The funds are transferred, and the action is logged in the transaction history.	
Basic Flow	User Action	System Action

	1. User selects the fund transfer option and inputs the beneficiary details.	2. System checks if the beneficiary is valid and asks the user for amount.
	3. The user enters the amount.	4. The system validates if the entered amount is present in the user's account and transfers the amount to the beneficiary.
	5. The user is displayed with the success message that money has been transferred.	6. The system updates both accounts accordingly.
Alternate Flow	2b. system displays error that beneficiary is invalid. 4b. System displays an error message that account balance is insufficient.	

4.5 Check Account Details

The feature enables users to view comprehensive details of their bank accounts, including but not limited to, account balances, transaction history, account numbers, and personal information associated with the account.

Use Case ID	UC_05
Use Case Name	Check Account Details
Description	Allows the user to view their account details, including remaining balance.
Primary Actor	User
Secondary Actor	System
Pre-Conditions	User must be signed in to access account details.

Post-Condition	User views the details of their account	
Basic Flow	User Action	System Action
	1. User selects the option to check account details.	2. System displays the requested account information.
Alternate Flow	1. System displays an error message that login credentials are invalid.	

4.6 Check card details

This feature allows users to view details of their linked debit and credit cards, including card numbers (typically displayed partially for security), expiration dates, available balance (for debit cards). Additionally, it provides option for freezing or unfreezing the card.

Use Case ID	UC_06	
Use Case Name	Check Card Details	
Description	Allows the user to view their card information, such as the card number, expiration date etc.	
Primary Actor	User	
Secondary Actor	System	
Pre-Conditions	User must be signed in to access their card details.	
Post-Condition	User is informed about their card details.	
Basic Flow	User Action	System Action
	1. User selects the option to check card details	2. System retrieves and displays the card information.

Alternate Flow	1. System alerts that user has entered invalid credentials. 1b. User selects a blocked card.
-----------------------	---

4.7 Mobile Top-Up

This feature supports instant mobile credit top-ups for various carriers and plans, enabling users to select the mobile operator, enter a mobile number, and choose the top-up amount before confirming the transaction.

Use Case ID	UC_07	
Use Case Name	Mobile Top-Up	
Description	Enables the user to add credit to a mobile phone balance through the banking app.	
Primary Actor	User	
Secondary Actor	System	
Pre-Conditions	1. User must be signed in. 2. User must have a valid payment method configured. 3. User must have sufficient amount in his account. 4. User enters a valid phone number.	
Post-Condition	The mobile number's balance is topped up.	
Basic Flow	User Action	System Action
	1. User selects the mobile top-up option.	2. System asks the user to enter phone number.
	3. User enters phone number.	4. System checks whether phone number is correct or not.
		5. If phone number is correct, systems asks for amount.
	6. User enters balance amount.	7. System checks if amount is

		present in user's account and updates the phone balance.
Alternate Flow	1. System alerts that user has entered invalid credentials. 4b. System displays error message if phone number is invalid. 6b. System displays an error message indicating that account balance is insufficient.	

4.8 Transaction Details

This feature allows users to view detailed information about individual transactions, including date, time, amount, transaction type (e.g., deposit, withdrawal, payment, transfer), the status of the transaction, and, where applicable, the counterparties involved.

Use Case ID	UC_08	
Use Case Name	Transaction Details	
Description	Provides the user with a detailed view of their past transactions.	
Primary Actor	User	
Secondary Actor	System	
Pre-Conditions	User must be signed in to access their transaction history.	
Post-Condition	User reviews their past transactions.	
Basic Flow	User Action	System Action
	1. User chooses to view transaction details.	2. System presents a list of recent transactions to the user.

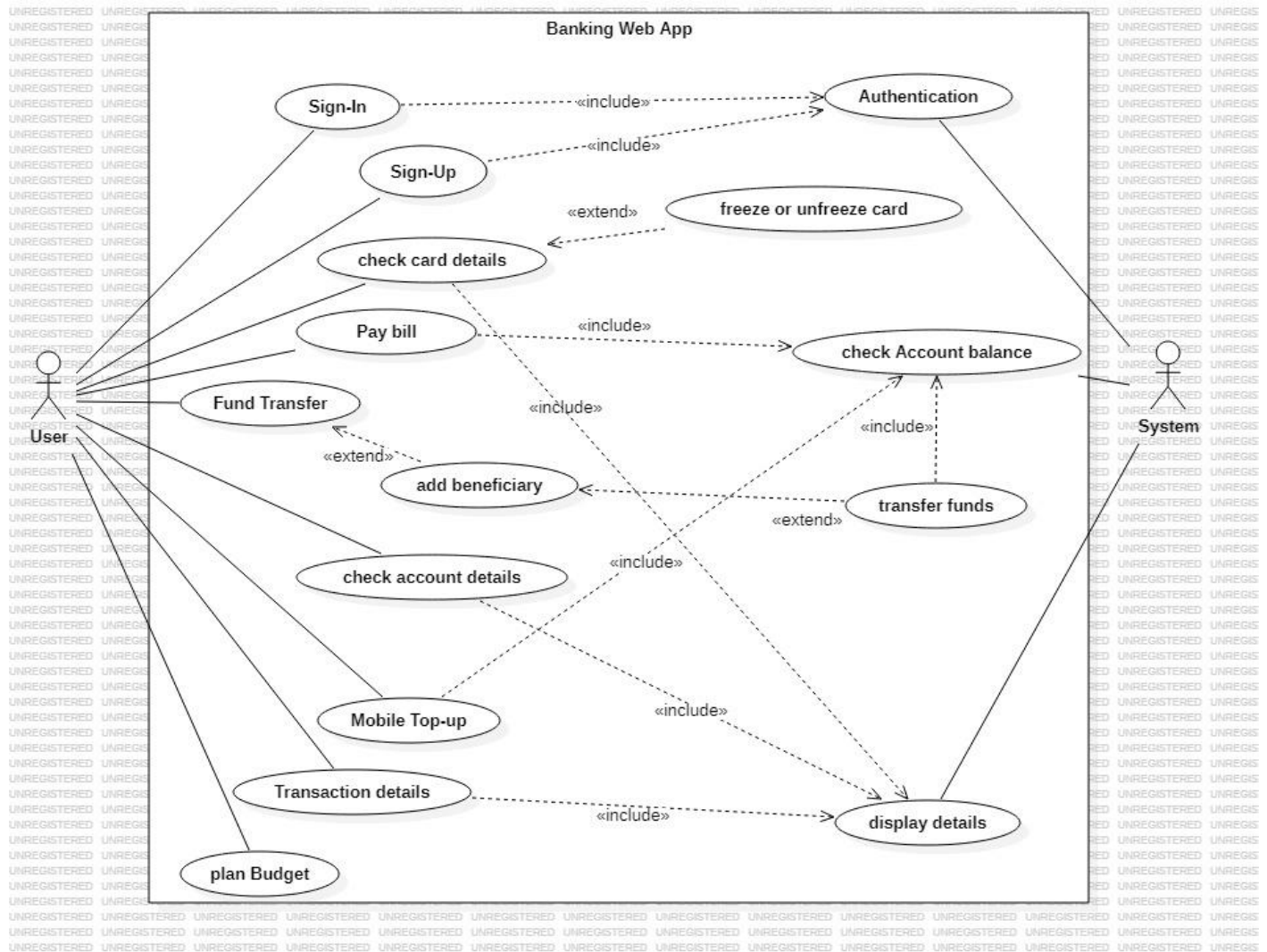
Alternate Flow	1. System alerts that user has entered invalid credentials.
-----------------------	---

4.9 Plan Budget

The "Plan Budget" feature allows users to set monthly or annual budgets for different categories (e.g., groceries, utilities, entertainment, savings) and track their progress towards these budgets in real-time. It integrates with the user's accounts to automatically categorize transactions and provide a comprehensive view of their financial health.

Use Case ID	UC_09	
Use Case Name	Plan Budget	
Description	Allows the user to set up and manage a personal budget plan.	
Primary Actor	User	
Secondary Actor	System	
Pre-Conditions	User must be signed in to access budget planning tools.	
Post-Condition	The user has a budget plan set up which can be tracked against their spending.	
Basic Flow	User Action	System Action
	1. User accesses the budget planning feature and inputs budget details	2. System creates a budget plan and provides tools for tracking spending
Alternate Flow	1. System alerts that user has entered invalid credentials. 1b. User enters unrealistic budget constraints.	

A centralized view of all the actors, their use cases, and their relationship with each other is shown in a complete use case diagram:



5. Other Nonfunctional Requirements

5.1 Performance Requirements

- Fast response times for user interactions to ensure a smooth user experience.
- Efficient handling of peak loads, especially during high transaction periods.
- Scalability to accommodate growing numbers of users and transactions without degradation in performance.

5.2 Security Requirements

- Secure data transmission (e.g., using SSL/TLS[2] encryption).
- Data encryption at rest to protect sensitive information stored in databases.
- Authentication (multi-factor authentication) and authorization mechanisms.

5.3 Software Quality Attributes

- Code and architecture designed for ease of maintenance and updates.
- Comprehensive documentation for developers and maintainers.
- Intuitive and user-friendly interface design.
- Integration capabilities with external systems, such as payment gateways, credit score services, and other banking institutions.

Appendix A: Glossary

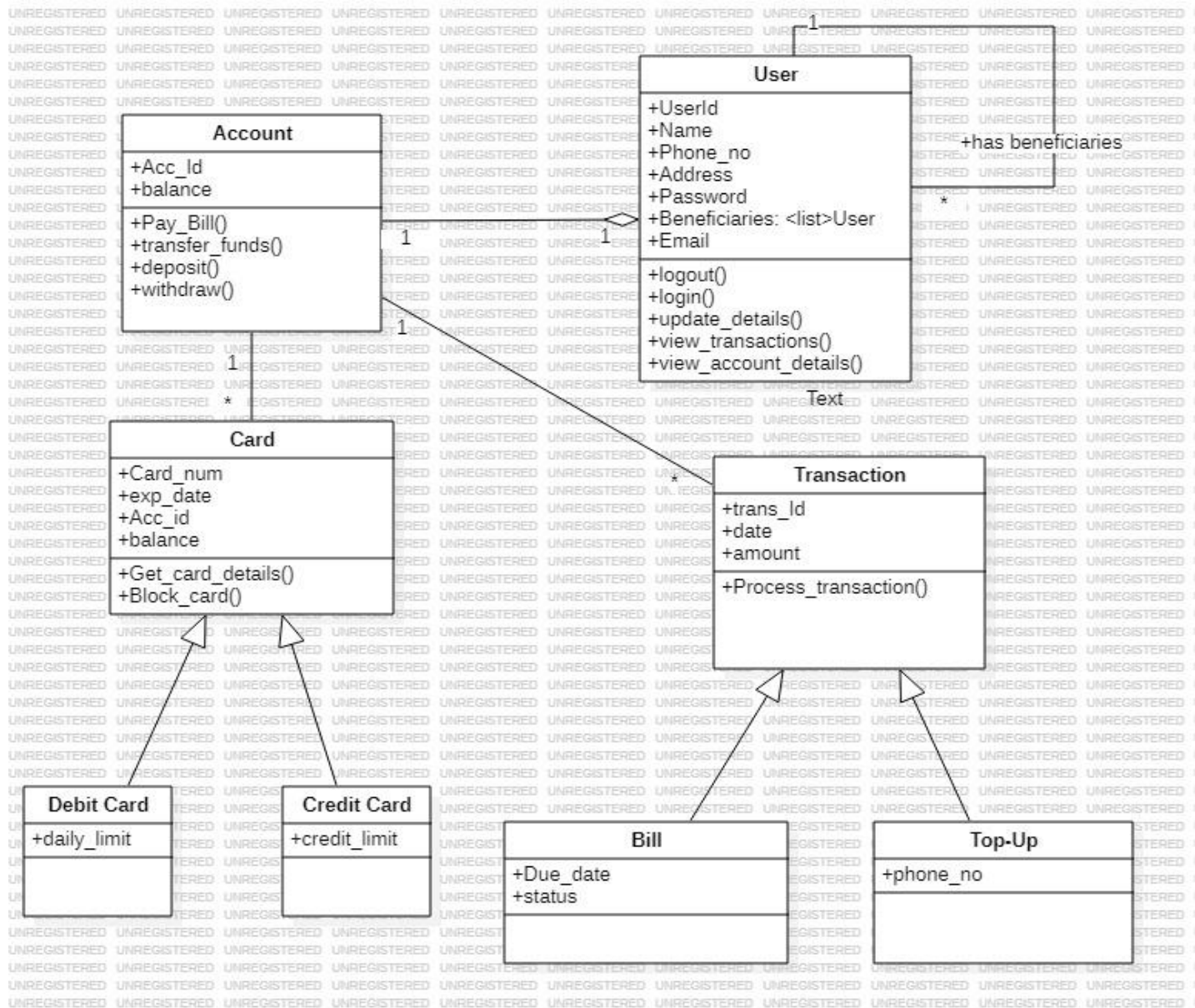
[1] API (Application Programming Interface): A set of protocols and tools for building software applications which in the context of your banking web application, allows for interaction with third-party services like payment gateways.

[2] SSL/TLS (Secure Sockets Layer/ Transport Layer Security): Cryptographic protocols designed to provide communications security over a computer network, important for securing online transactions.

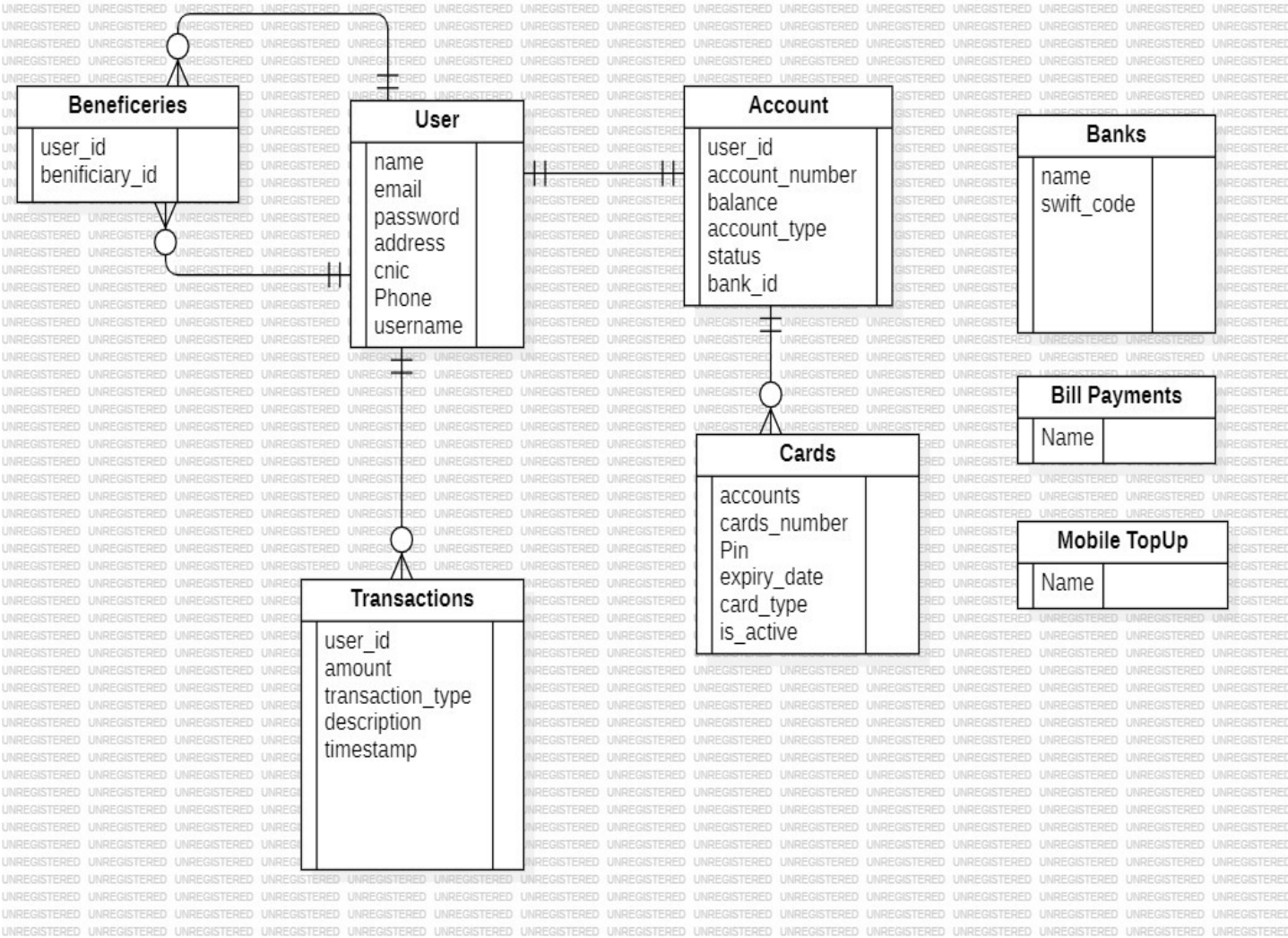
Appendix B: Analysis Models

Class diagram:

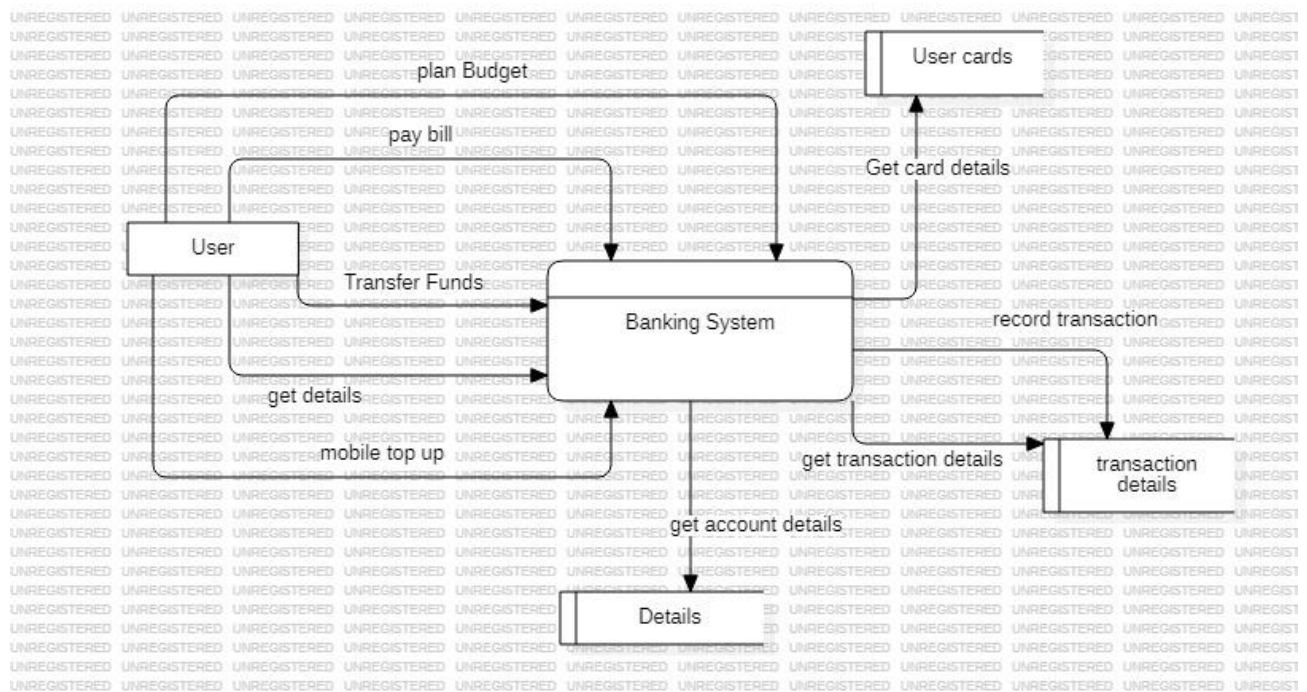
Attached below is a detailed class diagram that show different classes and their relationship with each other as well as the important data members that will be used to perform all the functionalities.



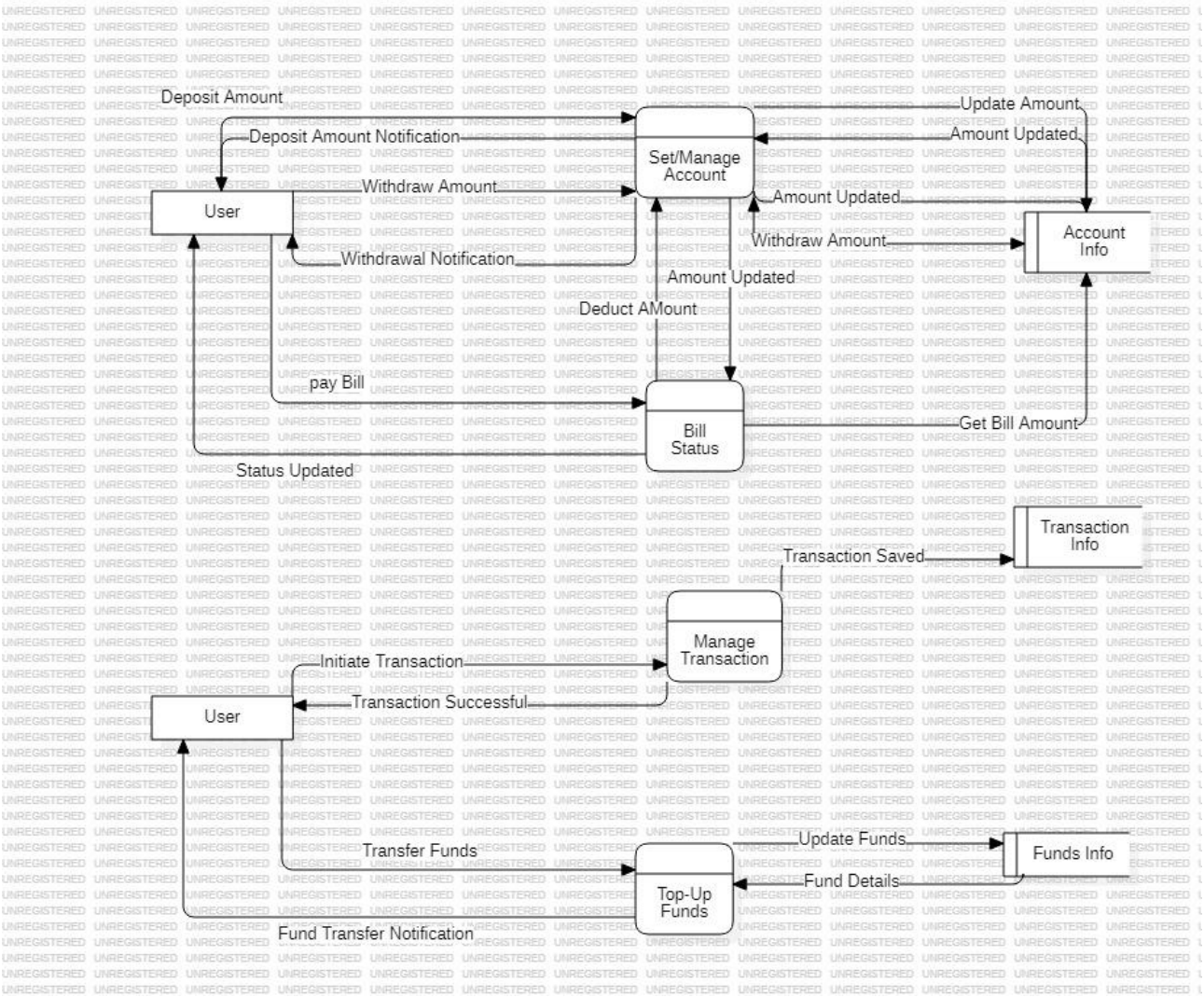
Entity Relationship Diagram (ER):



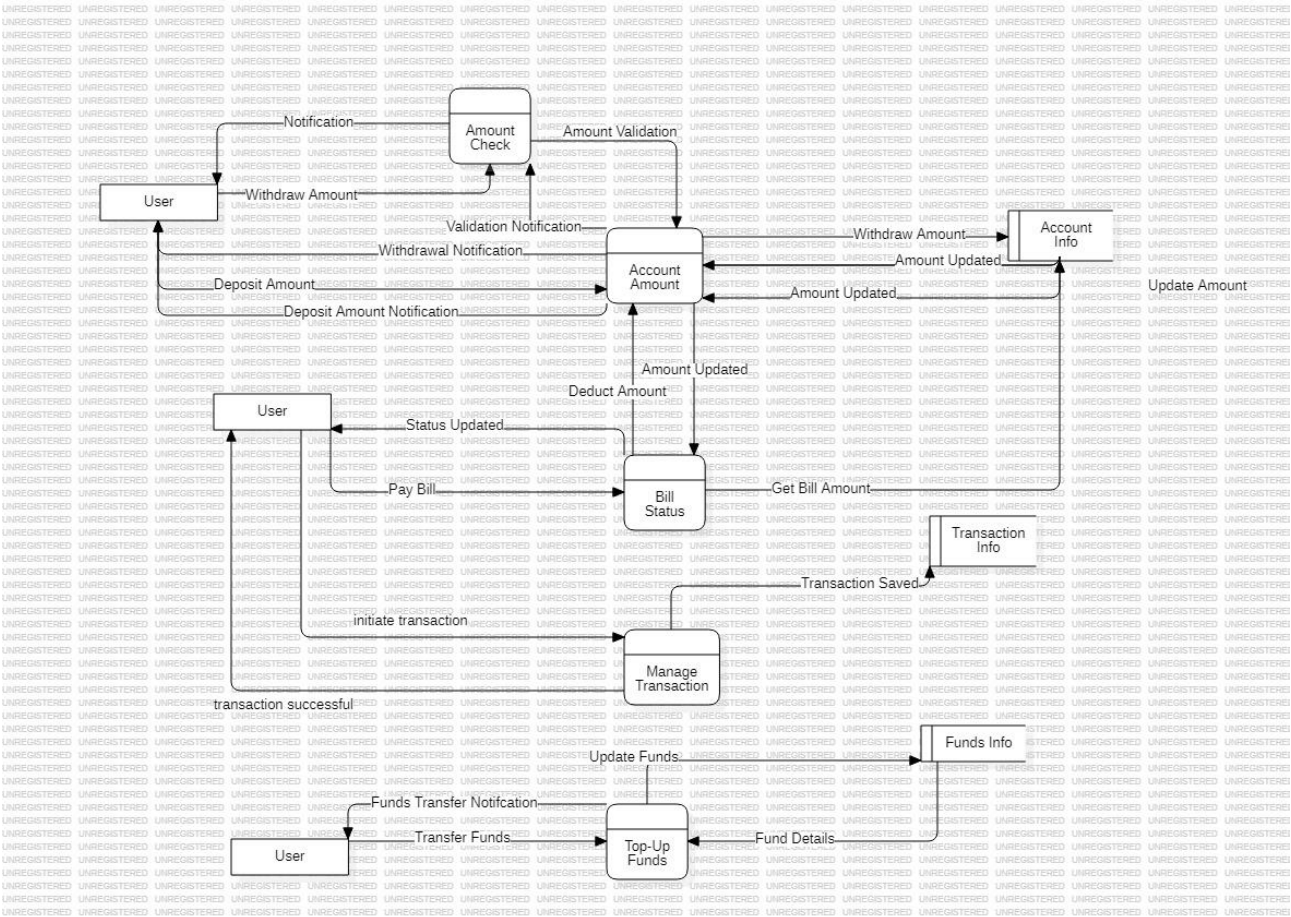
Data Flow Diagram (level 0)



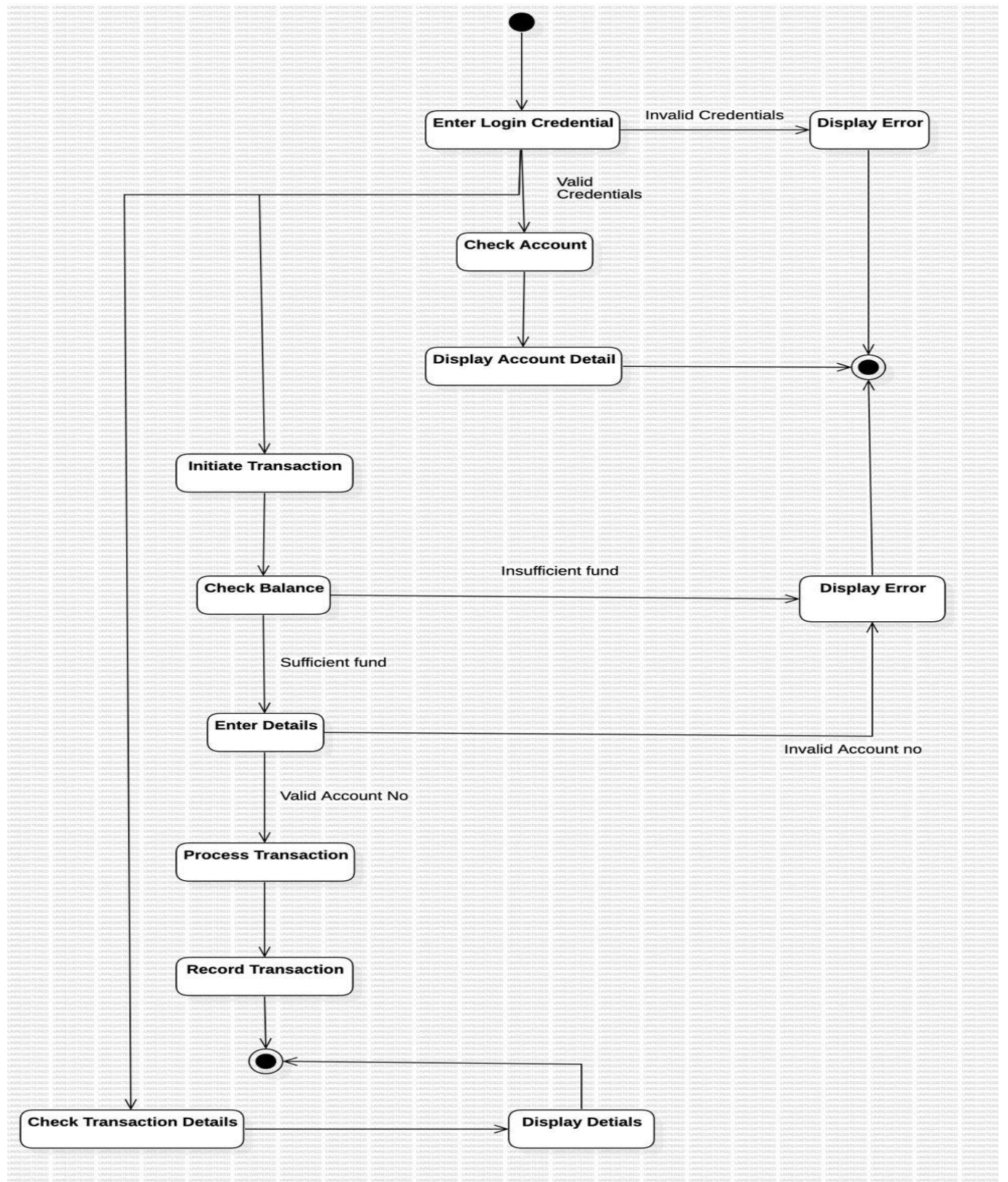
Data Flow Diagram (level 1)



Data Flow Diagram (level 2)



State Diagram:



Appendix C: To Be Determined List

- **Security Protocols:** Determine the specific security protocols to be implemented for data encryption and secure user authentication.
- **Third-Party Payment Gateways:** Selection of third-party payment gateways for processing transactions needs finalization.