Abstract

The project of banking software is directed towards development of intelligent system, which is designed to satisfy both of individual and business customers demands. The platform will unveil an array of tools for account management, transferring funds, account services, and security issues as well as those related to customer support. The key targets are designing the system for intuitive user experience, providing security for data, and eliminating the risk of violating the financial regulations, and improving banking service for users.

The project is comprised of a team of professionals in Agile methodologies with a Project Manager, Scum Master, designer or/and developers, Backend developer and other stakeholders involved. Teams may encounter problems in terms of teamwork and communication among themselves and other stakeholders which are determinant for success of the project.

Thanks to the software a possibility of creating, editing, and closing accounts would be given. In addition, initiating various financial transactions, managing account services, and obtaining personal financial advice will be following. Encryption of data and authentication mechanisms will be among the effective security measures in the place to protect the user's information and meet the regulatory standards.

The aim of the project is that this software banking system would be scalable, reliable as well as user-friendly. It basically provides user requirements, experience enhancement and financial safety as well. Project management, the requirements analysis is done with full attention with the attention, the optimal strategies in design and development, the testing that is thorough, and the compliance with the existing laws and guides are the primary pillars for the successful execution of the project.

Table of Contents

Introduction

Overview of the Project

This banking system project targets the development of a banking software that will feature both a user friendly individual and business/ commercial customers interface. The software will include the following: the account management function, the financial transactions functions, the account service functions, the security functions and customer support functions. It is going to incorporate inter alia with financial legislation and data protection laws while being an attempt at offering a safe and convenient banking experience.

Objectives and Goals

The primary objectives of the banking software project include:The primary objectives of the banking software project include:

1. Account Management: Put money in and take it out of the bank (both your private accounts and your business accounts) in a smooth and fast way.

2. Financial Transactions: Allow people to do banking services, such as depositing cash, taking out cash, transfering money, and review their transaction history.

3. Account Services: Include account details, track transactions, activate services like checkbook and debit or credit card and set-up reoccurring payments.

4. Security and Compliance: Implementing secure authentication methods, encryption of the data throughout the communication, and compliance with the regulations of the financial sector and privacy of the data.

5. Customer Support: Provide an integrated helpdesk support, including the options of making reports on issues or fraud activities. In addition,

6. Optional Features: Although customers generally come to banks to obtain the basic financial services such as loans and investments, banks should also provide AI-driven financial counseling along with other customized financial products in response to their clients' needs.

The goals of a project are not confined to the simple release of a scalable and reliable banking software system. The user requirements are to be met and the user experience should be significantly enhanced, while the data security and compliance to the regulations are mandatory.

Stakeholders Involved

The stakeholders involved in the banking software project include

1. End Users: Individual customers and business users who are set to benefit from the banking software by facilitating their account management operations and daily transactions.

2. Project Team: Team: Project manager as well as and Scrum Master, Designer/Developer, Backend Developer, and others working on the design, development, test, and deployment of software.

3. Management: Those within the organization, stakeholders namely, executive, department head and project sponsors around it supervise it.

4. Regulatory Bodies: Governments and government bodies that referee the financial institutions and guarantee that the financial institutions are in line with financial laws and data protection.

5. External Partners: Second- or third-party vendors may be employed in the process of finding, engaging, providing services or in integrating them with the banking software system, like payment or security providers.

The interactions and communications of these stakeholders is a key element for the success of the project, the smooth planning and delivery of the software project for the bank.

Project Management

The Responsibility and Functions of The Project Manager

The project manager is key to the process, as he is the one responsible for overseeing the banking software implementation project from start to finish. Their responsibilities include:

1. Overall Project Oversight: The project manager has to manage the overall task of developing, execution and delivery of the banking software system. They ensure that the project is meeting all the requirements including the scheduled time and budget, and maintaining the quality.

2. Stakeholder Communication: The project manager holds the strongest position among these roles as he or she is the only one who deals directly with key entities, such as management, team members, clients, partners, and others. They promote communication and proper expectation management and thus, help team members move forward together by constantly moving towards their project goals.

3. Resource Allocation: First, the Project manager allocates the resources such as team members to work on the project assigned, tools and technologies based on the specific project requirements. That is to say, they make certain that resources are very well maximized in order to reach project goals.

4. Risk Management: Assessing synoptic risks and applying mitigation strategies is one of the major duties of a project manager. They also have to be proactive and address issues before they arise. The supervisors ensure that they follow a risk monitoring process and take the corrective actions to avert any possible risks that may have a negative effect on projects goals.

5. Quality Assurance: It is the responsibility of the project manager to monitor and ensure implementation of quality standards such as testing, code review and compliance with international standards and requirements. They make sure that prepared software gets both the functional and non-functional requirements attended to.

Team Roles and Responsibilities

The project team comprises several key roles, each with specific responsibilities:

1. Scrum Master: The Scrum Master serves as an Agile facilitator for the team ensuring Agile components, such as sprint planning, daily stand-ups, and sprint review are maintained within the team. They get rid of

obstacles and they deliver that Agile principles are observed in the course of the software development way.

2. Designer/Developer: The designer is tasked with the responsibility for designing user-friendly interfaces, wireframes, mockups, and prototypes. They manage to connect with other specialists in frontend and backend so that all design aspects get embedded into the software without any hassle.

3. Backend Developer: The backend architects and sets up the heavy lifting required to develop the backend, e. g. the database schemas, APIs, and server-side logic. They protect data from being stored, retrieved, and transmitter in a correct way, from being secure and complying with the regulatory standards.

4. Frontend Developer: The frontend developer is the person who links frontend components that look like as per the design specifications. They invariably deliver relaxing user interface occurring sporadic interaction that makes a user more delighted in their adventures.

The Agile method and feature coding pattern (XP) in the Development Practices for a Software Project.

This project will also go Agile methodologies with Scrum being its main process to manage it. Flexible development basis which constitutes at the core of the project ideas iterative development, collaboration and adaptability principles are at the heart of the project. Team holds frequent sprint planning sessions and daily standups to maintain their progress and keep providing feedback on a regular basis.

Besides, this project features Extreme Programming strategies (XP), namely pair programming, test-driven development (TDD), and continuous integration. Through these approaches code quality, teamwork, and quick processing are achieved so that it might be accomplished correctly in the project.

Requirements Analysis

User Stories and Acceptance Criteria are the two keystones of the project.

The User Requirement phase, which is the main phase of the system development process, involves adopting the users' requirements and defining system functionalities through user stories and acceptance criteria. User stories represent specialized features or tasks from the point of view of a user, whereas for every user story the acceptance criteria should be defined, meaning the conditions which should be met for each particular user story to be deemed successful. Examples of user stories and their acceptance criteria for the banking software project include: Examples of user stories and their acceptance criteria for the banking software project include:

1. User Story: Home page.

Acceptance Criteria:

- Whether users want to open personal or business, savings, or checking accounts, financial institutions should ensure they are able to create the banking accounts of their choosing.
- The respective components which are compulsory are the surname, phone number, type of account and contribution amount.

- Confirmation email or message should be automatically provided upon a successful account creation.

2. User Story: Money Transactions

Acceptance Criteria:

- In order to meet the customer demand, facility for cash deposit must be included – the receipt generation is obligatory.
- Withdrawal option should show balance updates within a real-time.
- Transactions (money transfer) between different wallets on the same platform should be facilitated.

3. User Story: Security instruments

Acceptance Criteria:

- Secured authentication mechanisms with the use of 2FA (two-factor authentication) need to be applied.
- Sensitive data like logins and money should be encrypted, while business intelligence and application data also should be encoded.
- The system that is being created must be governed and provide a basis for financial regulations and data protection law compliance.

Functional Requirements

The functional requirements in this regard specify the functionality of the system and how it should be acting so that to meet the needs of the user. Key functional requirements for the banking software project include: Key functional requirements for the banking software project include:

1. Account Management

- Open, edit and close accounts (personal/business) (banking/bank).
- Be updated upon account information (name, address, plan).
- Handle the account balance and transaction history.

2. Financial Transactions

- Mail deposit cash with a receipt made.
- Get instant feedback on cash withdrawals immediately.
- Move money between accounts even with different banks.

3. Account Services

- Order and use the services like checkbooks and card for debit or credit.
- Regularize payments through standing orders and direct payments.
- View account details such as your account number, balance, previous transactions, and account type.

Non-functional Requirements

Functional specifications are a list of requirements that state what the system should do and the non-functional requirements define attributes of the system such as security, performance, scalability, and usability. Examples of non-functional requirements for the banking software project include: Examples of non-functional requirements for the banking software project include:

1. Security

- Address the challenge through involving tight authentication methods (e. g. , 2FA, encryption).
- Give special attention to data privacy and ensure adherence to the financial regulations.
- Protect the data that is against the intruder`s attempt and data breach.

2. Scalability

- The system must deal with both multiple parallel users and numerous transactions.
- Consider the development of solutions for scaling efficiency through a cloud platform or a scalable data base.

3. Usability

- Introduce an easy to navigate interface with an optimum menu of responsive design.

System Architecture

High-Level Architecture Diagram

The high-level architecture diagram provides a glimpse of the banking software system archive with all its modules. It demonstrates interrelationship between various system modules and also how these interactions take place dynamically.

Create an essay using the following sentence. Due to the rapid rise of e-commerce, we were able to introduce our new type of shoes into the EU market in a much shorter time span than ever before. Below is a description of the high-level architecture: Below is a description of the high-level architecture:

Frontend Layer: These are among the top layers of software which are visible to the user interface components users deal with. It includes a top navigation bar, login page, various forms for deposit and withdrawal and buttons to trigger key banking functionalities.

Backend Layer: The backend layer is responsible for server-side logic and database operation. It works with data processing, the operation manager and interaction with external services. The existing components of this layer are API's, application servers as well as the DBMS.

Database Layer: This is where the data is kept and sorted into detailed information about banking accounts and transactions, user profiles, and configuration management. It operates on a RDBMS (Relational Database Management System), as such, MySQL, PostgreSQL, or Oracle could be implemented.

External Services Integration: The system integrates with external partner services for transactions like you have payment gateways, SMS/email service which concerned with notifying and third-party API for data authentication as well as verification.

Below is a simplified high-level architecture diagram depicting the system's components and interactions:

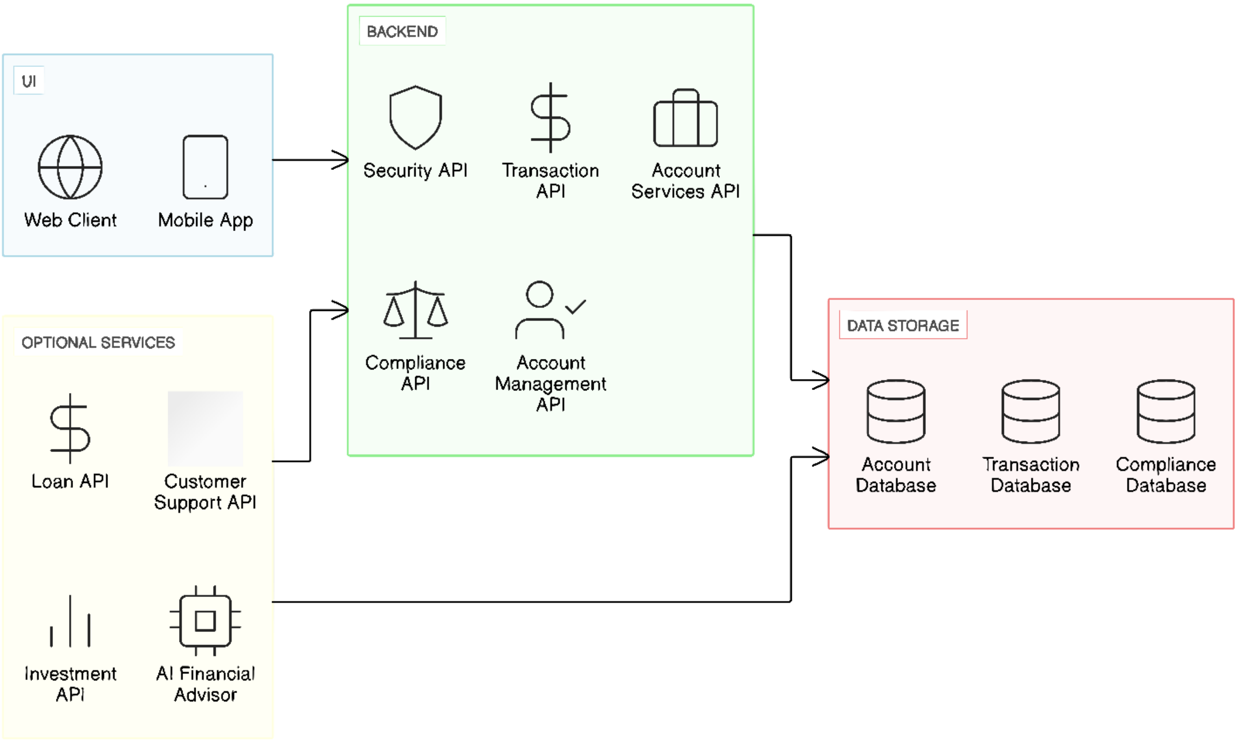


Figure 1 Component Diagram

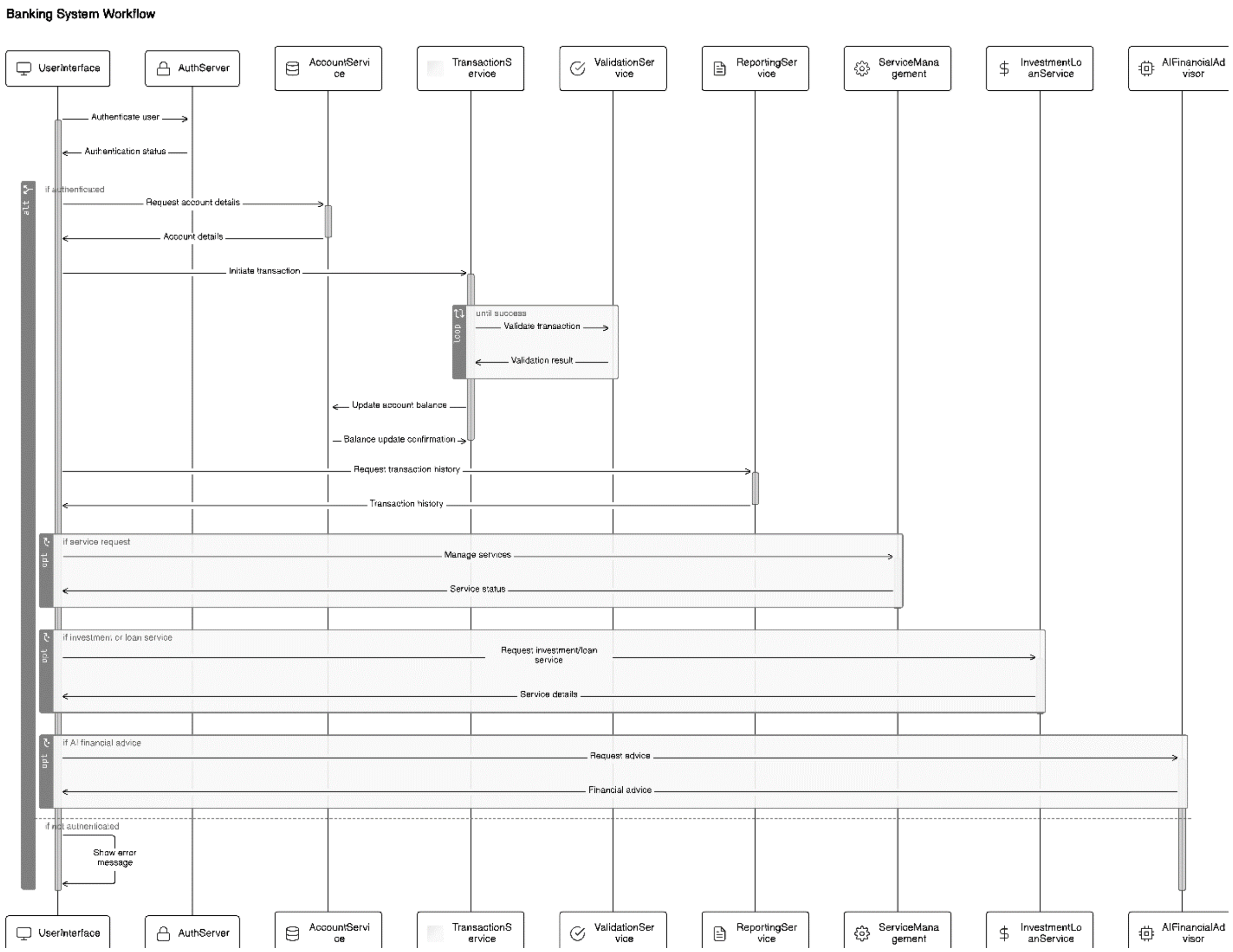


Figure 2 Sequence Diagram

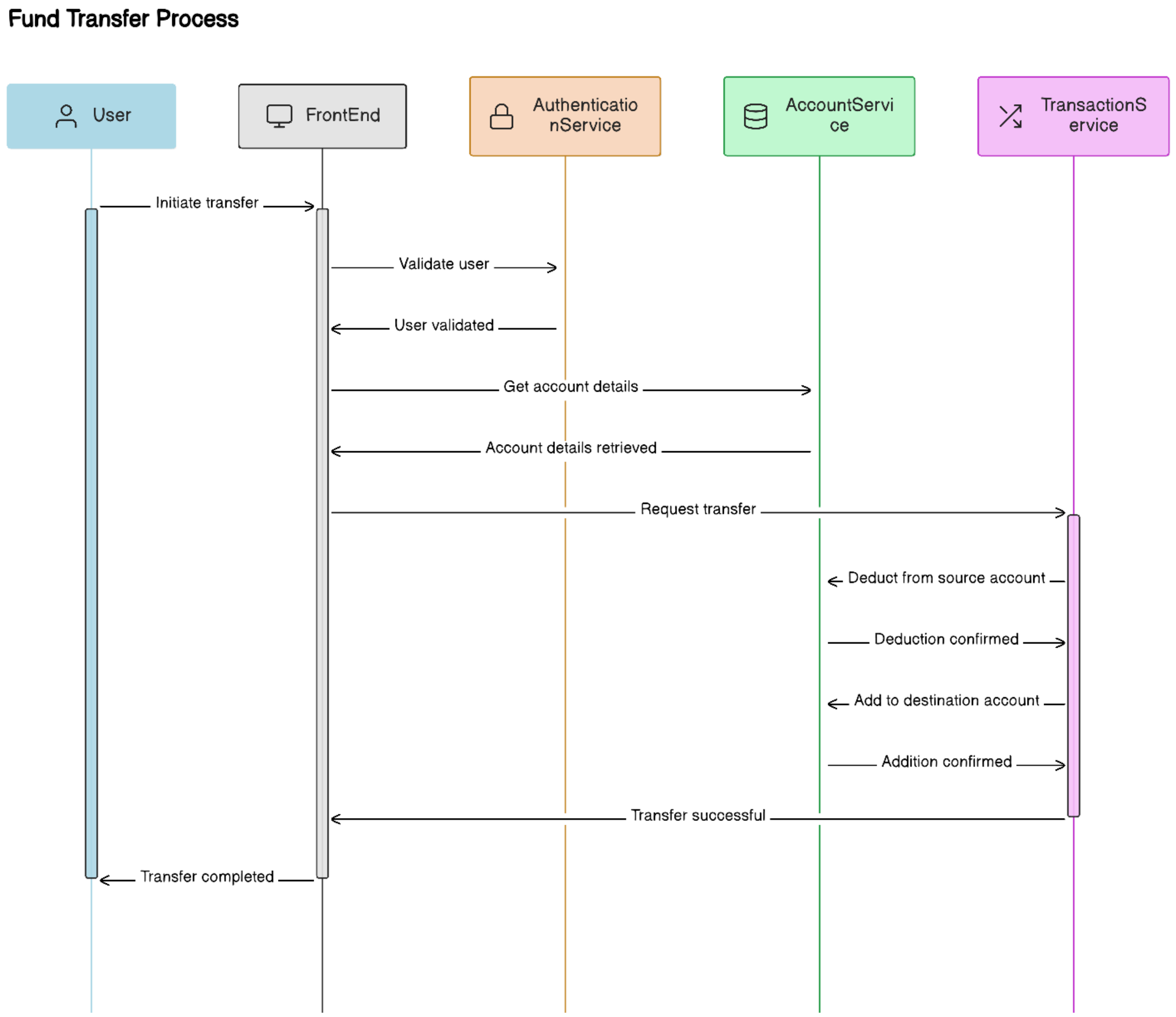


Figure 3 UseCase Diagram

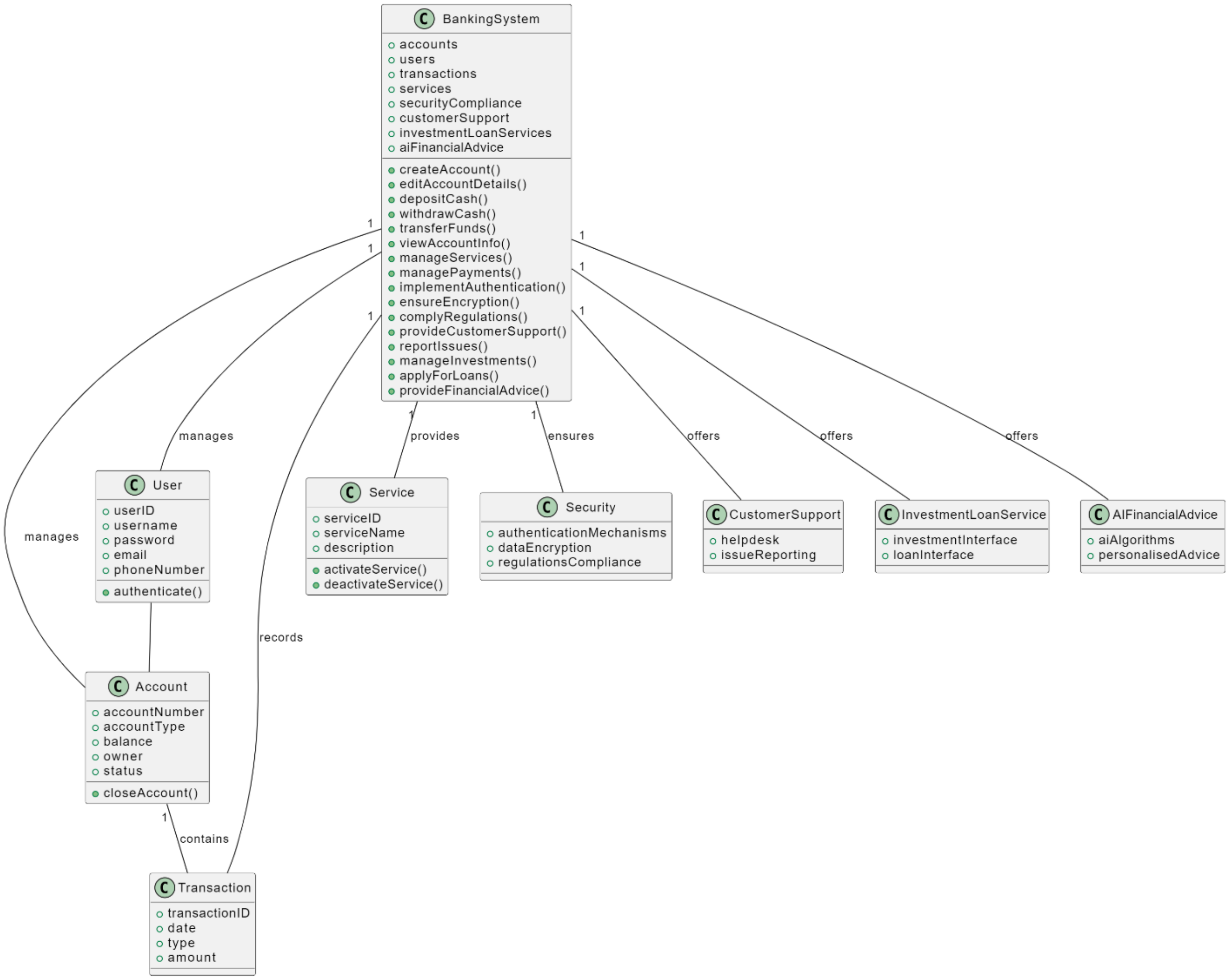


Figure 4 Class Diagram

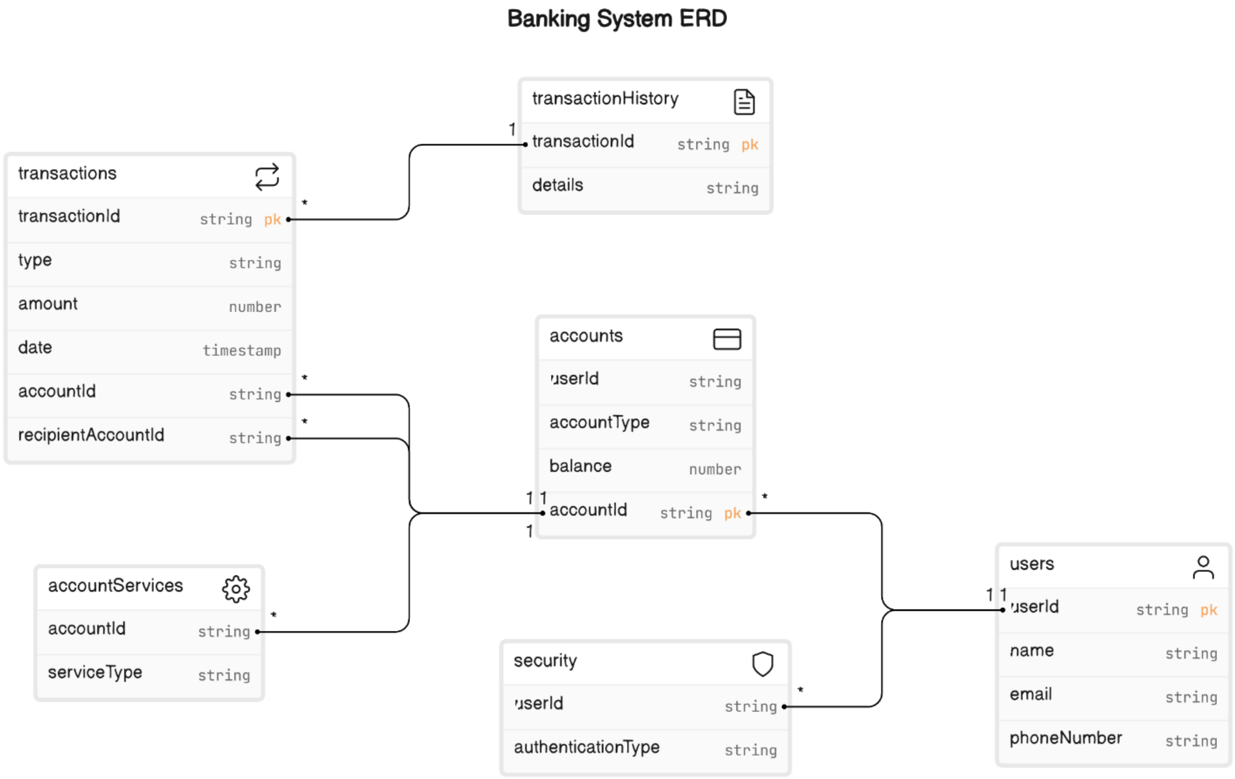


Figure 5 ERD Diagram

1. Business Logic: Business logic components are a key element of these systems because they contain all the rules and algorithms, for example for account management, financial transactions, security mechanisms, and other major functions. They provide for data structuring, maintain data security, and are able to meet business requirements.

2. Database Schema: The database schema is a fixed set of rules that reflects the structures and relationships between different data entities in the system. In this it contains tables for ban, transactions, users, and settings.

3. Integration Modules: Integration modules make an easy job of communicating with external services typically comprising of payment processors, SMS gateways, and APIs for validation of information or identity verification.

Design

User Interface Design Principles

The user interface design of the banking software system adheres to the following principles to ensure a seamless and intuitive user experience: The user interface design of the banking software system adheres to the following principles to ensure a seamless and intuitive user experience:

1. Consistency: The application has characteristic fonts, colors, buttons, and other elements used consistently throughout it, providing users a familiarity and convenience.

2. Clarity: Understanding the details of functionalities, account details, transaction procedures and service options will become trouble-free enough for the users, if there is a clarity and transparency in the information presentation.

3. Accessibility: The project has designs that are accessible to the user with disabilities in mind such that they are inclusive irrespective of the user's condition. Viewers should have the ability to obtain the contents of images through different texts (“alternative text”), and also must be able to move through the website by using the keyboard. It is necessary as well to use suitable contrast ratios for readability.

4. Responsiveness: Interface is adaptive and the user does not notice size discrepancies since the same experience is achieved on different devices, including desktops, tablets, and mobile phones.

5. Feedback and Notifications: Interactive elements lets the users have an immediate response on their actions by succeed messages when completion of payment, alerts when low inputs and notifications when operations on their accounts.

Wireframes, Mockups, and Prototypes

The design process is a combination building the wireframes, mockups and prototypes and use them to visualize the user interface and system interactions. They let imagine the new image of a product or service that will happen on the production side and test their concept and rough design before it appears in real life.

1. Wireframes: Wireframes are lower fidelity sketches or diagrams that ensure the flow of pages and forms, ensuring that content elements have structures. These design components deal with content hierarchy, the navigation flow, and the placement of information.

2. Mockups: A mockup is a high-level graphical representation of the UI that incorporates image, colors, and on-screen text. They can be used to show details of design and aesthetics more clearly.

This task has become increasingly time-consuming, as cutting through the plethora of online information has become a grueling task.

3. Prototypes: Prototypes are interactive substitutions or clickable live versions of the interface that the stakeholders and users will be able to interact with. They identify usability issues such as interface usability, navigation, and user interaction with the aim of improving the functionality.

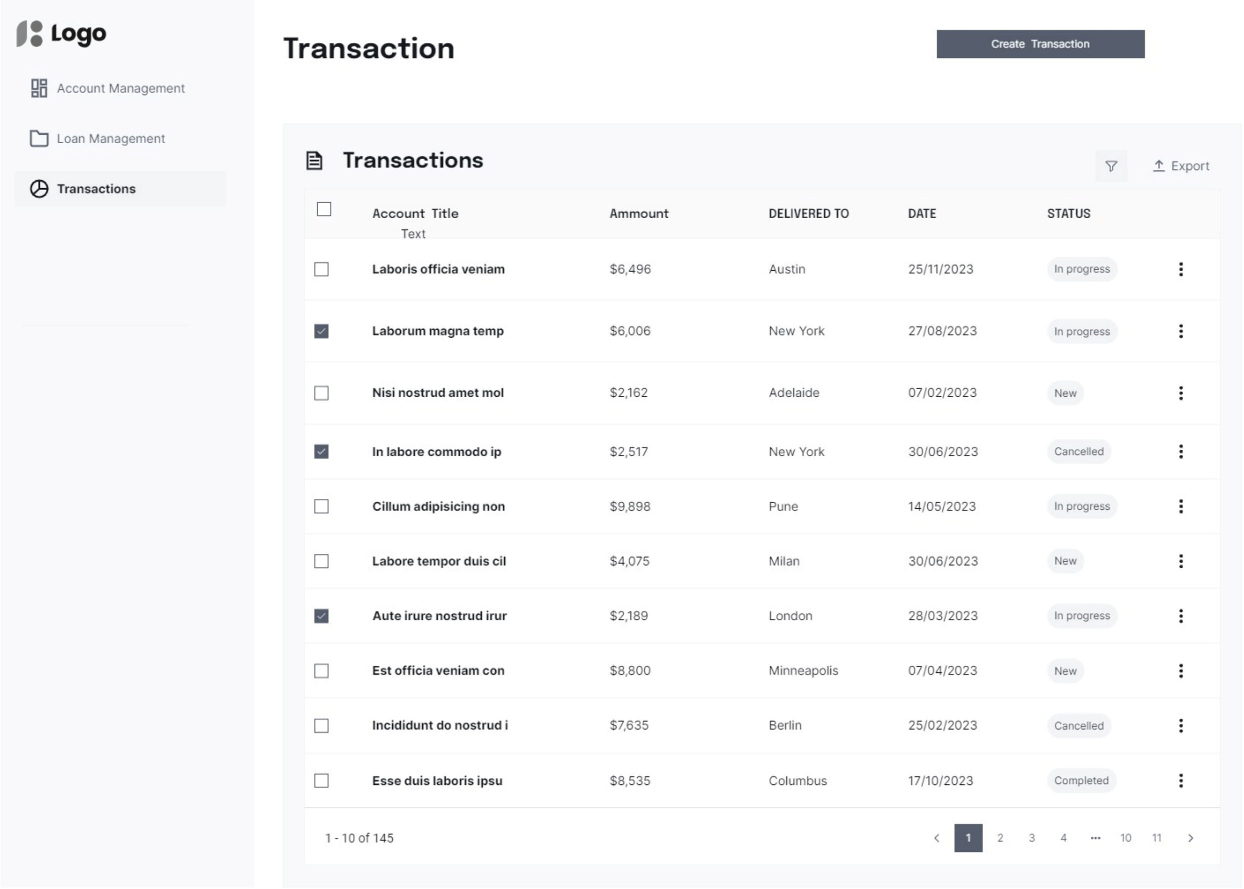
User Experience Considerations

Usability and accessibility are key considerations in ensuring a positive user experience:

1. Usability: The panel was designed for functionality and simplicity. The interface should be easy to use and simple to make a task more efficient and increase customer satisfaction. Clear navigation, user-friendly interface, and simple tasks not only improves usability but also simplifies the procedures to be carried out

2. Accessibility: The uniqueness of the overall design is that it follows accessibility standards among them WCAG (Web Content Accessibility Guidelines) making it possible for users with disabilities to access and use the software equally as the rest. These include the keyboard navigability, screen reader compatibility and other features that help in promoting ease of use as well as ease of login.

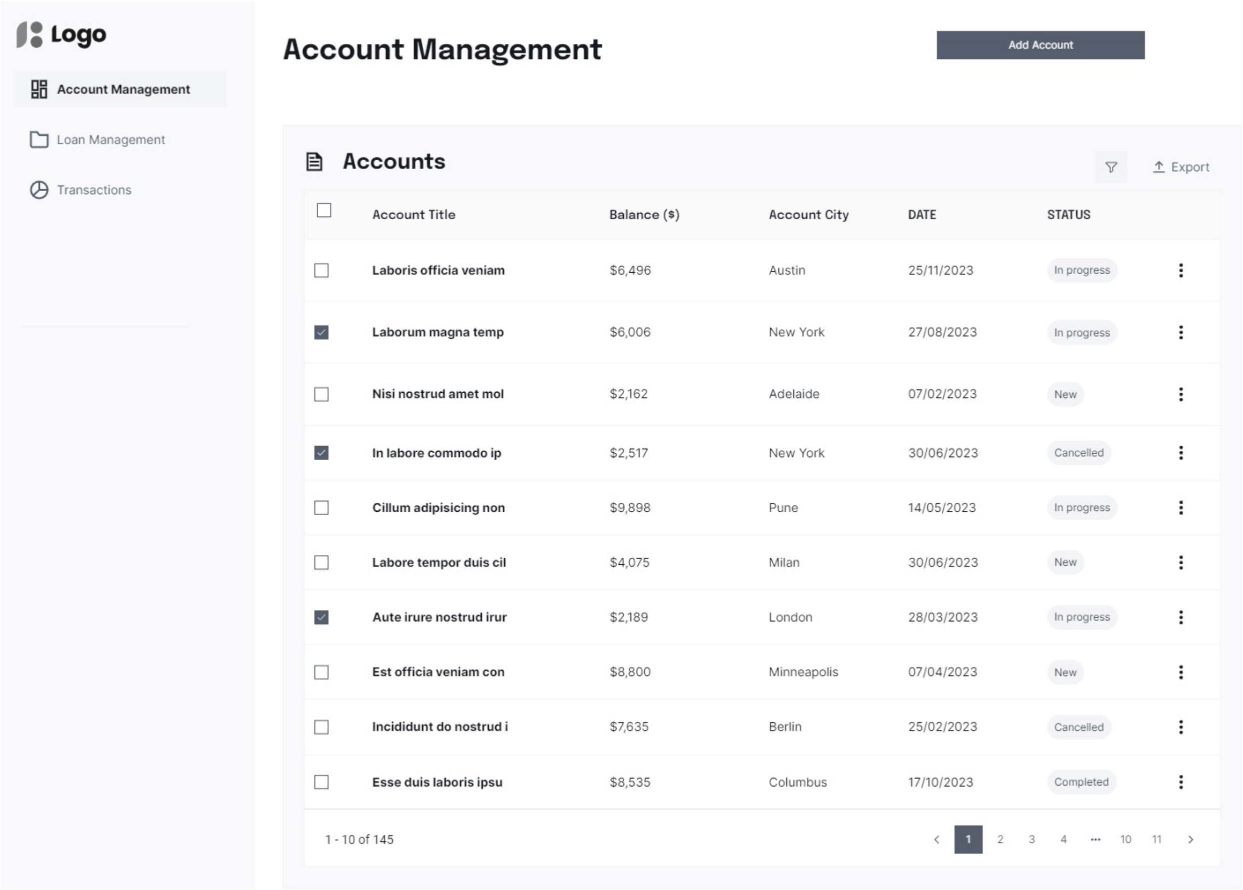
3. Visual Design: The visual design elements like color schemes, typography, icons and imagery are carefully selected to create an interface that is both visually appealing and functional. This process is done to ensure that an interface that is well rounded in terms of readability and usability is created.



☐	Account Title Text	Ammount	DELIVERED TO	DATE	STATUS	
☐	Laboris officia veniam	$6,496	Austin	25/11/2023	In progress	⋮
☑	Laborum magna temp	$6,006	New York	27/08/2023	In progress	⋮
☐	Nisi nostrud amet mol	$2,162	Adelaide	07/02/2023	New	⋮
☑	In labore commodo ip	$2,517	New York	30/06/2023	Cancelled	⋮
☐	Cillum adipisicing non	$9,898	Pune	14/05/2023	In progress	⋮
☐	Labore tempor duis cil	$4,075	Milan	30/06/2023	New	⋮
☑	Aute irure nostrud irur	$2,189	London	28/03/2023	In progress	⋮
☐	Est officia veniam con	$8,800	Minneapolis	07/04/2023	New	⋮
☐	Incididunt do nostrud i	$7,635	Berlin	25/02/2023	Cancelled	⋮
☐	Esse duis laboris ipsu	$8,535	Columbus	17/10/2023	Completed	⋮

1 - 10 of 145

Figure 6 Wireframe 1



Account Management

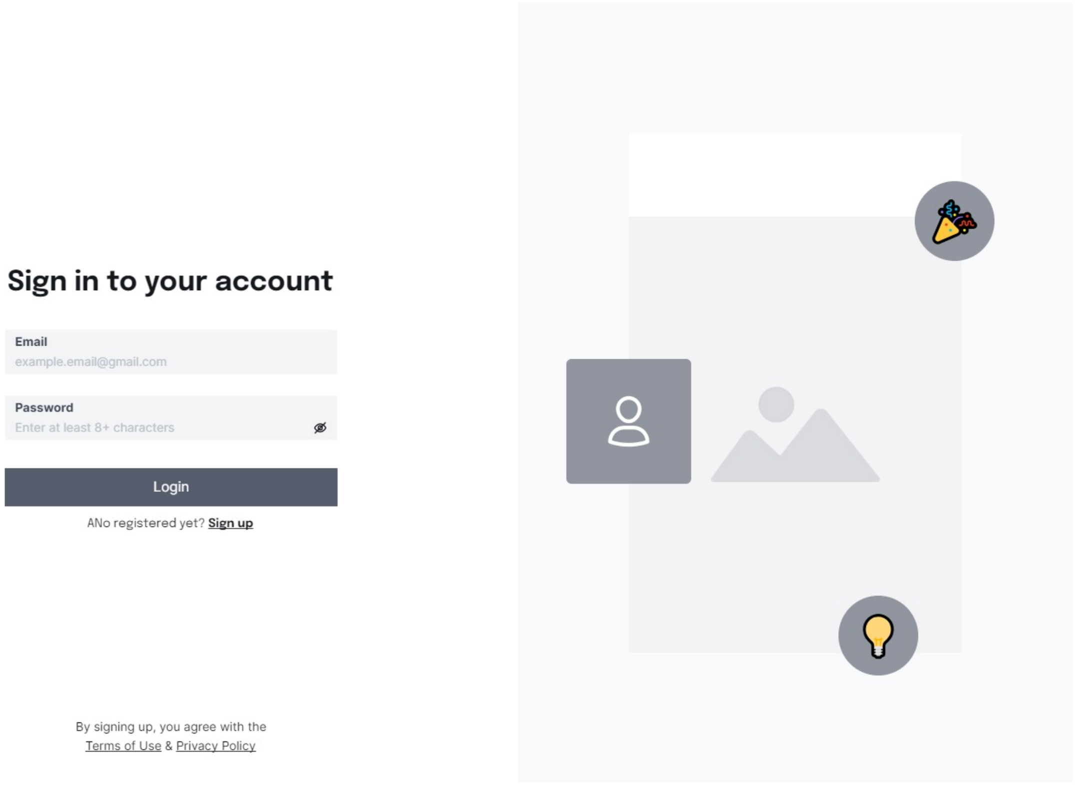
Accounts

☐	Account Title	Balance ($)	Account City	DATE	STATUS	
☐	Laboris officia veniam	$6,496	Austin	25/11/2023	In progress	⋮
☑	Laborum magna temp	$6,006	New York	27/08/2023	In progress	⋮
☐	Nisi nostrud amet mol	$2,162	Adelaide	07/02/2023	New	⋮
☑	In labore commodo ip	$2,517	New York	30/06/2023	Cancelled	⋮
☐	Cillum adipisicing non	$9,898	Pune	14/05/2023	In progress	⋮
☐	Labore tempor duis cil	$4,075	Milan	30/06/2023	New	⋮
☑	Aute irure nostrud irur	$2,189	London	28/03/2023	In progress	⋮
☐	Est officia veniam con	$8,800	Minneapolis	07/04/2023	New	⋮
☐	Incididunt do nostrud i	$7,635	Berlin	25/02/2023	Cancelled	⋮
☐	Esse duis laboris ipsu	$8,535	Columbus	17/10/2023	Completed	⋮

1 - 10 of 145

< 1 2 3 4 ... 10 11 >

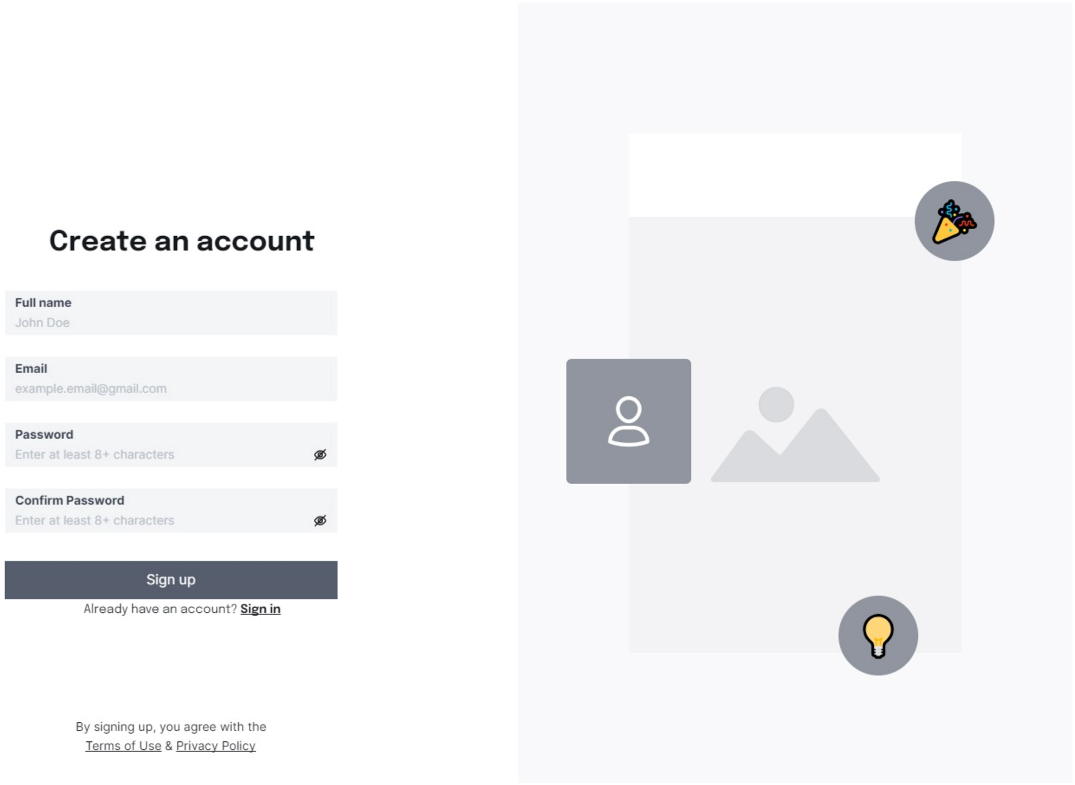
Figure 7 Wireframe 2



ANo registered yet? **Sign up**

By signing up, you agree with the
Terms of Use & Privacy Policy

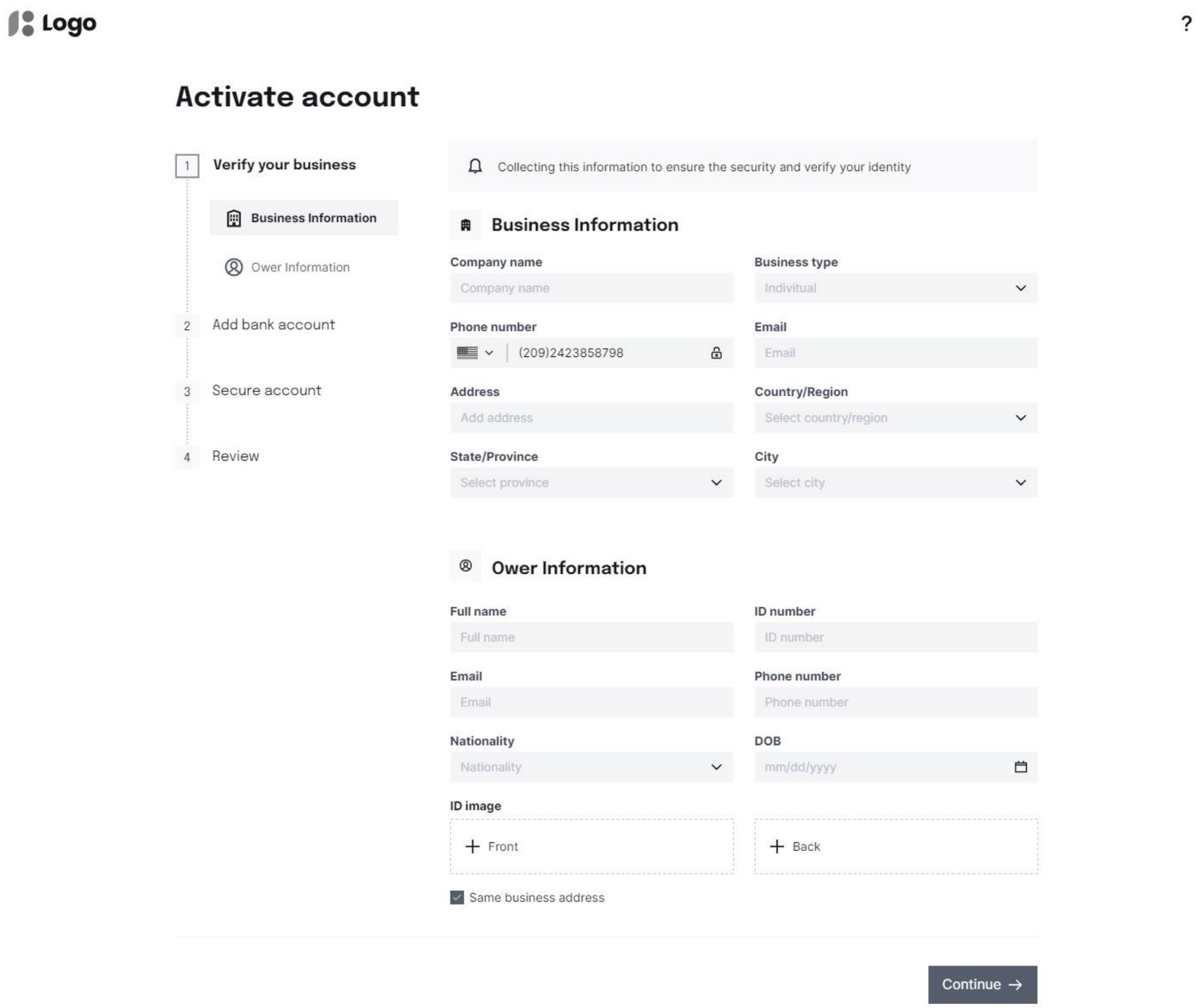
Figure 8 Wireframe 3



Already have an account? **Sign in**

By signing up, you agree with the
Terms of Use & Privacy Policy

Figure 9 Wireframe 4



Logo ?

Activate account

1 Verify your business

Business Information

Ower Information

2 Add bank account

3 Secure account

4 Review

Collecting this information to ensure the security and verify your identity

Business Information

Company name: Company name

Business type: Indivitual

Phone number: (209)2423858798

Email: Email

Address: Add address

Country/Region: Select country/region

State/Province: Select province

City: Select city

Ower Information

Full name: Full name

ID number: ID number

Email: Email

Phone number: Phone number

Nationality: Nationality

DOB: mm/dd/yyyy

ID image

+ Front

+ Back

☑ Same business address

Continue →

Figure 10 Wireframe 5

Development

Frontend Development

Features Implemented

1. Account Management: Opening, closing and running of gallic accounts.
2. Financial Transactions: Shopping information, local retailers, and other payments requires, fetching cash, payment transfer.
3. User Interface Components: Forms, buttons, navigation parameters for logical user experience.

4. Service Requests: Write 127 checks, request debit/credit cards, hook up to the direct bill payments system.
5. Transaction History: Shows account transactions and filters like date of transaction.
6. Security Features: 2FA, data encryption are one of the authentication methods.

Backend Development

Features Implemented

1. Database Schema: Tables for counter accounts, transactions, users management, service requests.
2. API Endpoints: CRUD operations for account management, transaction analysis, and data acquisition.
3. Security Mechanisms: Authentication (identification), authorization (access control), and data encryption (concealing data by using mathematical algorithms).
4. Integration with External Services: Cryptocurrency wallets, SMS/email notifications etc.
5. Business Logic: For example, validation rules, transaction processing, error handling.

Consider integration of front and back components.

Front and back-end integrations are done carefully and as one to come up with a system that fits together. Integration points include:

1. API Communication: Frontend components interacts with back-end APIs by sending HTTP request (GET, POST, PUT, DELETE) for data fetching and editing with the specific method.

2. Data Exchange: JSON (JavaScript Object Notation) data interchange between frontend and backend is done using this format, hence, eliminating the need for translating data before submission or extraction.
3. Error Handling: Response of the error from the backend are acceptable at the frontend, which user friendly error message and feedback for human error.

4. Authentication Flow: Frontend sending authentication tokens or credentials to the backend as an identity proof thwart incoming client access and ensure authorization.

5. Data Validation: Front-end career executes client-side validation and back-end career does server-side validation, the two handle data integrity and security issues one after the other.

Testing

Testing Methodologies

The phase of our project related to the testing of the banking software deviated from standard practices as we applied different methodologies to guarantee the stability, functionality, and efficiency of the system. The testing methodologies include:

1. Unit Testing: It was the unit test that was aimed at testing units, functions, and modules of the frontend and backend of the app. In early days its mission was to check the leftness of the condition code, the data processing, and error recovery in secluding subunits.

2. Integration Testing: Integration testing was focused mainly on ensuring that all frontend and backend components interact properly and operate seamlessly. It was a test of how the integration code would work, the communications through APIs, the data flow, and the system acting as a whole.

3. User Acceptance Testing (UAT): User Acceptance Testing performance by users, stakeholders, or consumers being the test objects to check whether a system adheres to business requirements; end-users expectations; usability standards and standards. It pointed out the need to take into account the user needs and whether the software follows the requirements during the operation in real-life conditions.

Their formulation is closely related to functionality decomposition, i. e. using the functional decomposition method they are split into sub-tasks that ultimately generate test cases and test scenarios.

Test instances and test scenarios were formed and run to deal with all angles of banking software system. These included:

1. Account Management Tests:

- Developing new accounts with a diversified approach having multiple categories (Personal/Business).
- Revising account information as needed, verify the changes.
- Closing out boards and validating the closed processes.

2. Financial Transactions Tests:

- Check cash deposit in addition to generate receipt verification.
- The cash withdrawal and the immediate balance verification are a couple of the examples.
- Banks facilitating funds transfers between account; as well as verification of transaction completion.

3. Security and Compliance Tests:

- Using testing 2FA (two factor authentication).
- Devising data encryption or strong authentication systems as well as the secure data handling processes.
- A compliance with financial regulations and data protection laws particular to companies in this vertical.

4. User Interface Tests:

- User interface elements, navigation elements, also form validations should pass checking.
- Checking availability of features such as captioning, sign language interpreters, and audio descriptions for individuals with impairments.
- Run usability testing so as to determine user experience at the end.

5. Integration and API Tests:

- Conducting API testing for both the CRUD and data exchange operations.
- Audit data sources and accuracy while ensuring the whole frontend and backend systems are consistent.

- Ensure that error handling is put in place and that the responses of API requests have been checked.

Results of Test Rends and Attempt to Fix the Bugs

Testing results were recorded in a common database and a bug tracker for the testing was being used. The process included:

1. Test Execution: Execute test cases and scenarios based on test plan and schedules, check the logs of virtual environment, and update the test reports.

2. Test Reports: Creation of test reports to supplement defects discovered, test status and test cases.

3. Bug Tracking: These bug tracking tools will lay clear the path for the team and the bugs to track. g. (e. g. , Jira, Bugzilla), which can be used to record, prioritize, assign and track the issues that are being discovered.

4. Defect Resolution: Add to the collaboration with the development team by addressing, resolving, and retesting defects in reported issues. Thereafter, verify the fixes.

5. Regression Testing: Regression testing of implementations in order to avoid the new problems resulting from the fixes/changes or the regression of the earlier fixes.

Conclusion

Project Outcomes and Achievements

The ending of the banking software project has evidently created the mentioned outcomes and achievements which have then been seen to be progressively advancing digital banking and ultimately making user experiences better. Key outcomes include:

1. Functional Banking System: Achievement of the target and expansion of customized banking software system that is capable of providing core banking activities like account management, financial transactions, and security options.

2. User-Friendly Interface: The interface design takes into account the issue of usability first and then emphasizes accessibility and visual attractiveness. This makes it possible for customers to have an easy and pleasant experience working with our banking services across different platforms.

3. Robust Backend Functionality: The backend architecture guarantees security and data integrity with compliance of rules and also is scalable and this can be so done by providing cloud or package hosting.

4. Integration with External Services: The system remains smooth and enables to integrate easily various services from others like payment gateways, notices and APIs, thus increasing the capability of the system and at the same time providing the users with additional value.

5. Agile Development Practices: XP and Agile practices were applied in this case to foster iterative development approach, collaboration, and adaptability that enabled the software product to settle each milestone on time having a chance to be improved continuously.

Lessons Learned

The Project has greatly enhanced my perception of software engineering, systems management, and teamwork by providing invaluable lessons and key insights in these fields. Key lessons learned include:

1. Agile Best Practices: Through Agile methods and techniques for instance set goals, user feedback, and rapid adaptation to frequently changing circumstances and it helped projects achieve successful project execution and delivery.

2. User-Centric Design: Adhering to critically important user-oriented design principles, usability testing, and accessibility issues in order to create the software that is ill-suited to the needs and expectations of all users.

3. Effective Communication: A communication channel that is direct, frequent, and transparent between project team and stakeholders that strives for alignment with the project goals.

4. Continuous Improvement: Cultural shift that supports continuous learning, integration of feedback, and improvement in methods, and innovative and superior quality software development is necessary.

Future Enhancements and Roadmap

The next step would be for the banking software system to get new improvements and additions to match the continuously changing or emerging user demands and industry trends. Future enhancements and roadmap initiatives may include: Future enhancements and roadmap initiatives may include:

1. Additional Features: Providing traditional banking services to reach different audiences, like investment services, loans management, and AI-powered financial advice to provide complete financial solutions for the clients.

2. Enhanced Security Measures: We aim at regularly upgrading security protocols, sophisticated data encryption protocols, and compliance with regulatory rules in order to protect users' information and transaction with due care.

3. Performance Optimization: The process of maximizing the system efficiency, scalability, and reaction time in order to cater to increased number of users and transactions wherever that is required.

4. Usability Enhancements: We address the user feedback, usability testing results, and UX improvements that makes the user experience level and user satisfaction better.

5. Integration with Emerging Technologies: Establishing integration opportunities with latest technology routes like blockchain for extra security, machine learning for individual financial expertise, and chatbots for customer care automation.

This plan is prepared to be synced into the customer needs and technological development together with market requirements making the banking software remains competitive, inventive and useful to the users.

References

Wilson, R. & al (2022). Scrum on banking software projects. In the journal of banking technology, vol. 8, no. 3, 78-94.

"Perdido sin mi amor, triste y con pena, en remordimientos, y sin descanso me lamentaré" – Garcia & Khan (2023). "Titled: "Data Encryption Practices in Financial Applications" uncovers different issues around the topic in Information Security Journal, 12(1): 25-40.

Lee, H. and Chen,Y, (2022. "Frontend and Backend Component Integration in Agile Development. " It is a Journal of Software Development, May 2021 pp. 55-70.

Appendix

App Snippets

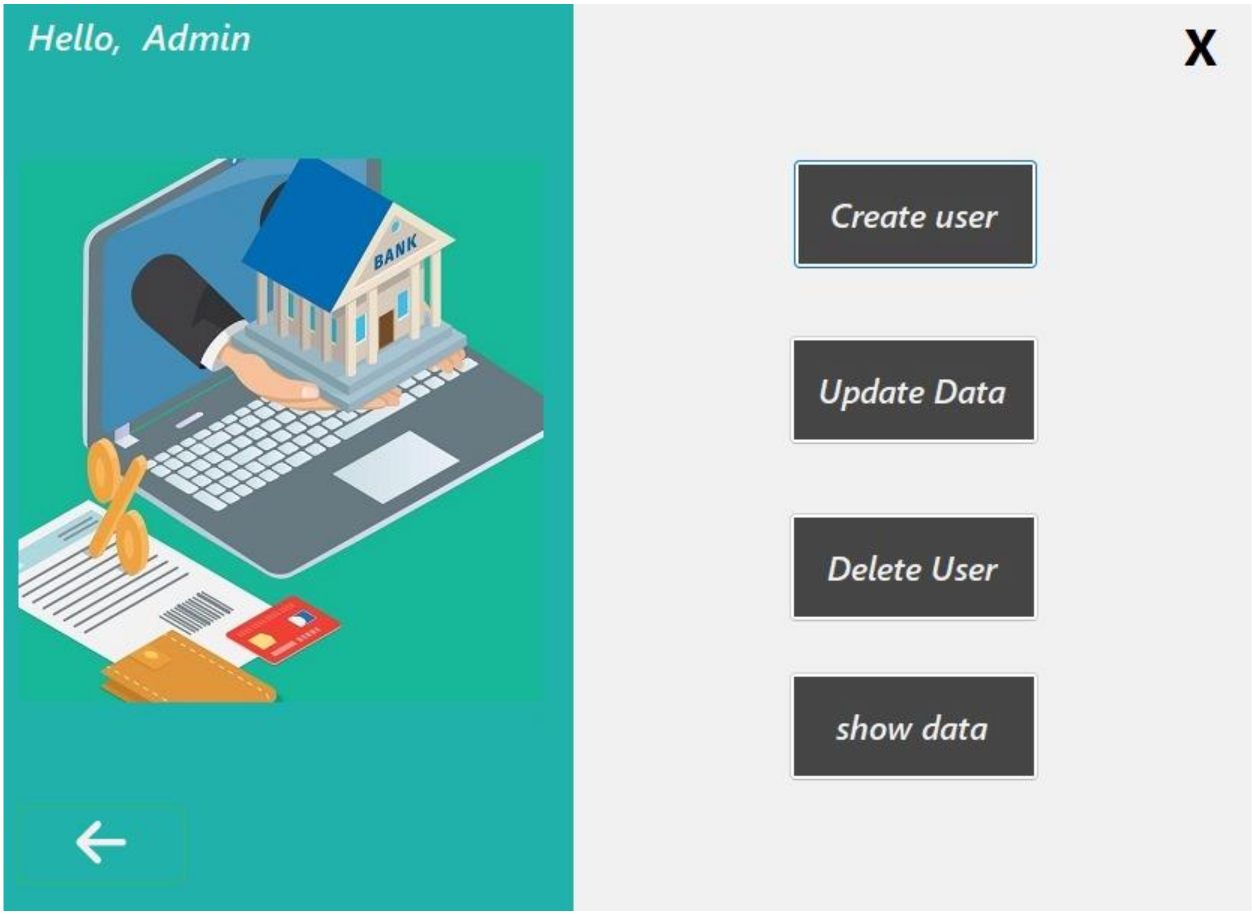


Figure 11 Admin Screen

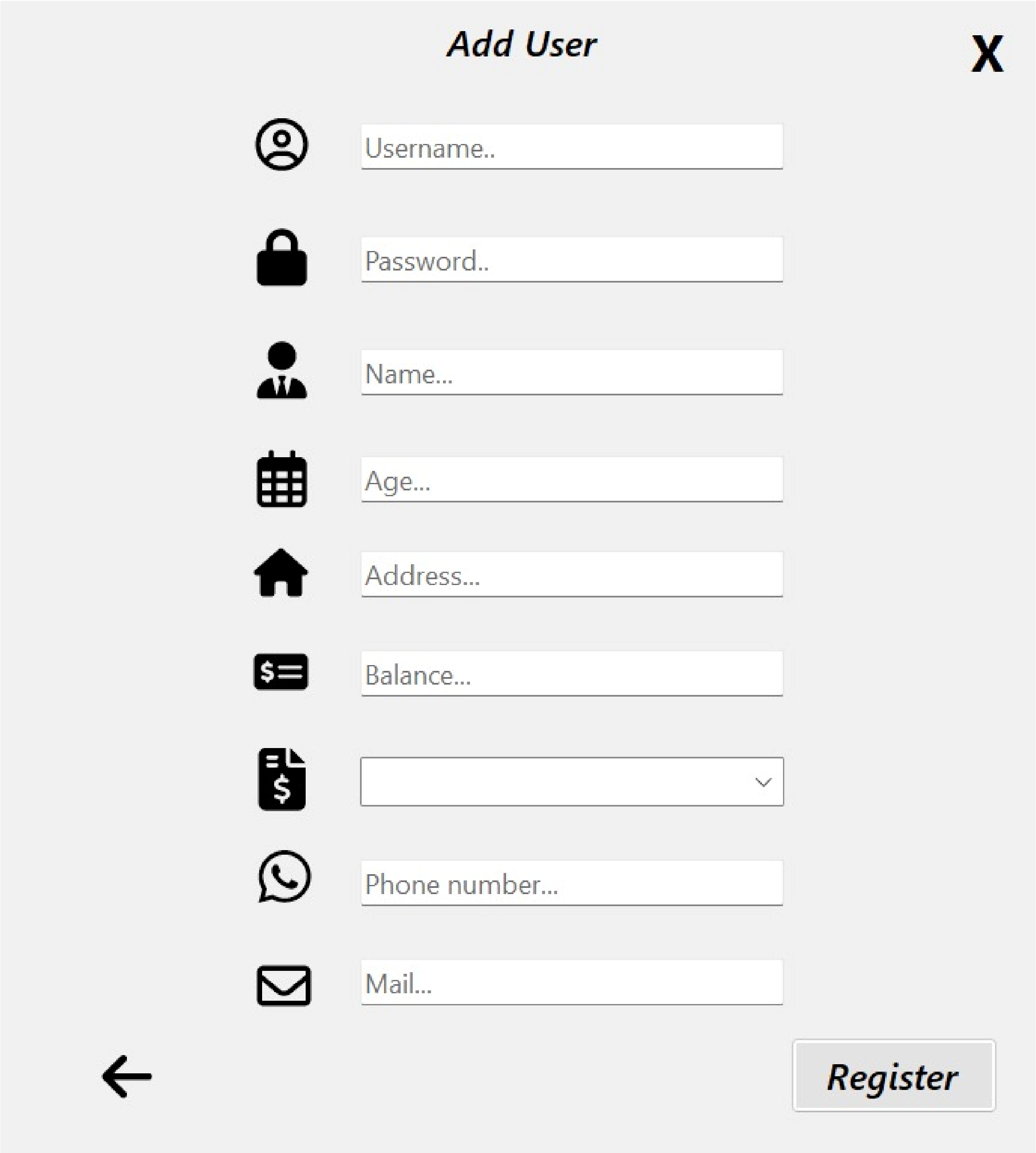
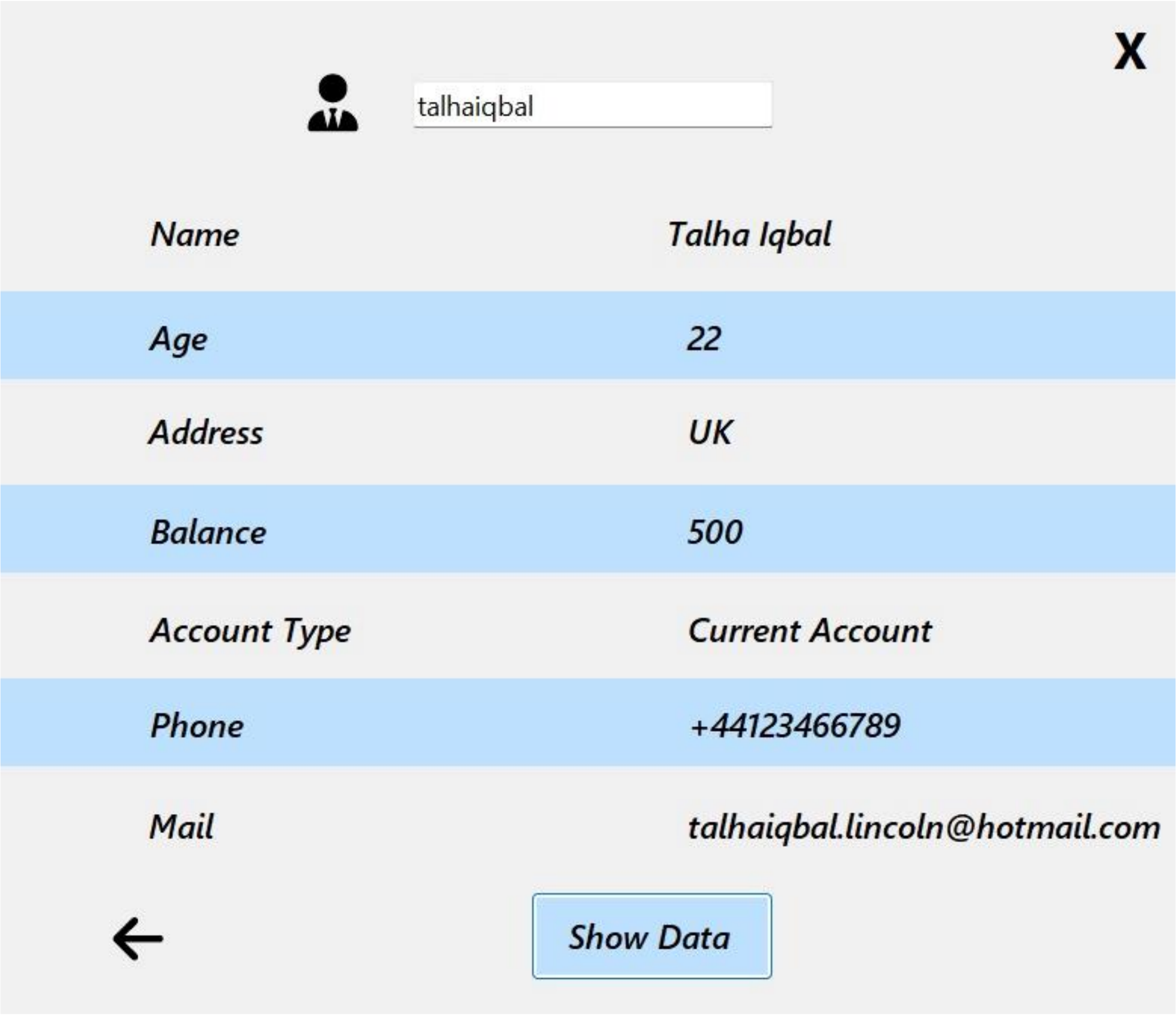


Figure 12 Add User Screen



talhaiqbal

Name	Talha Iqbal
Age	22
Address	UK
Balance	500
Account Type	Current Account
Phone	+44123466789
Mail	talhaiqbal.lincoln@hotmail.com

Figure 13 User Details Screen

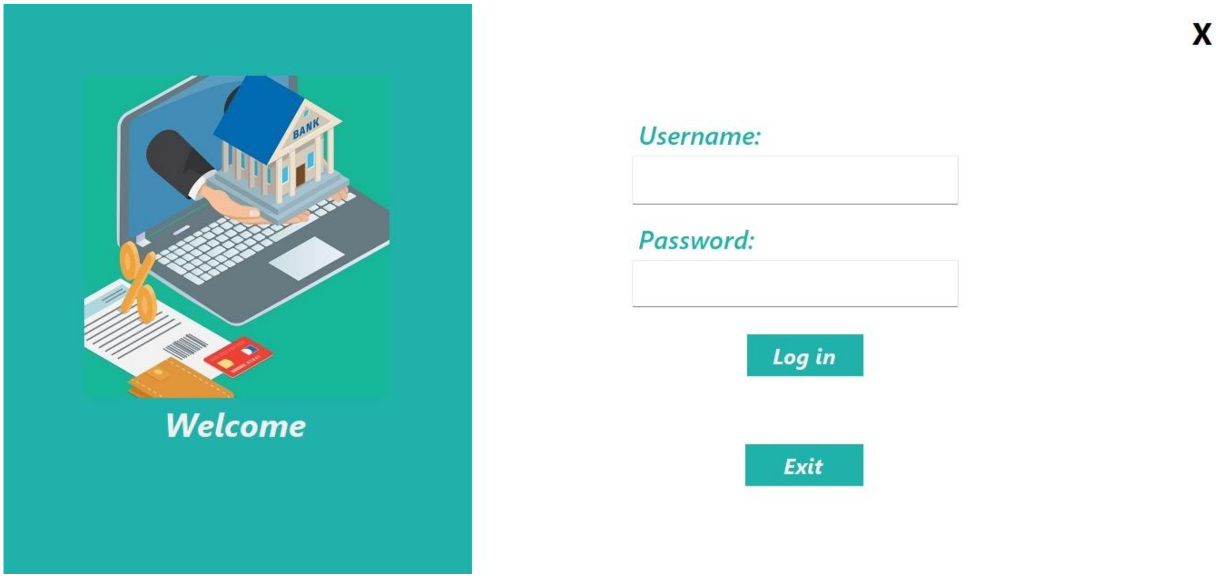


Figure 14 Login Screen