Software Requirement Specification

For

Automated Banking Website

Version 1.0 approved

Prepared by: Anamika Patel

HCL Training

Date: 15/01/2026

Submitted in partial fulfillment

of the requirements of

Software Engineering (Agile Methodology)

TABLE OF CONTENTS

List Of Figures

LIST OF TABLES

1. Abstract

An **Automated Banking Website** is a web-based application designed to provide customers with secure and convenient access to banking services such as account management, fund transfer, bill payments, transaction history, and loan services. In today's fast-paced digital world, customers expect instant access to banking facilities without visiting physical bank branches. This project focuses on the design and development of an **Automated Banking Website using Agile methodology**, which emphasizes **flexibility, iterative development, continuous improvement, and customer feedback**.

The proposed system provides a centralized digital platform where users can **register, log in securely, view account details, transfer funds, pay bills, check transaction history, and manage banking services online**. Traditional banking methods require customers to visit branches, fill out forms, and wait in queues, which is time-consuming and inconvenient. Manual handling of banking records can also lead to **data errors, delays, and inefficiency**. The proposed automated system overcomes these challenges by digitizing banking operations and providing real-time access to services.

The system includes essential modules such as **user authentication, account management, fund transfer, transaction processing, bill payment, and customer support**. It ensures secure access using **encrypted login credentials and role-based authentication**. Customers can perform banking operations anytime and anywhere using an internet-enabled device. The platform also provides instant transaction confirmation and maintains a complete transaction history for transparency.

The project follows the **Agile Software Development Model**, where development is carried out through multiple **sprints**. Each sprint delivers a functional module such as user registration, account dashboard, fund transfer, or transaction history. Agile practices such as **sprint planning, daily stand-ups, sprint reviews, and retrospectives** ensure continuous improvement, faster delivery, and high software quality.

The Automated Banking Website is designed using a **modular and scalable architecture**, allowing easy future enhancements such as mobile banking integration, AI-based financial assistants, and biometric authentication. The system ensures **data security, reliability, and compliance with banking standards**. A user-friendly interface ensures smooth navigation for customers of all age groups.

Overall, this project demonstrates how the integration of a **secure Automated Banking Website** with **Agile methodology** can significantly improve banking efficiency, enhance customer experience, reduce operational workload, and provide a reliable digital banking solution for modern financial institutions.

2. Introduction

2.1 Introduction

In today's digital era, banking services are rapidly shifting from traditional branch-based systems to online and mobile platforms. Customers demand quick, secure, and convenient access to their bank accounts without the need to visit physical branches. An Automated Banking Website provides a digital solution for managing banking operations efficiently.

This project aims to develop a modern Automated Banking Website that allows customers to perform banking activities such as checking account balance, transferring funds, and viewing transaction history online. The system is developed using Agile methodology to ensure flexibility, fast development, and continuous improvement.

2.2. Problem Identification

Traditional banking systems require customers to visit bank branches for basic services such as fund transfers, account updates, and bill payments. This results in long waiting times, paperwork, and inconvenience. Manual banking operations are time-consuming and prone to human errors, which can lead to incorrect transactions and record mismatches.

Banks also face challenges in managing large volumes of customer data and transactions manually. Lack of automation results in slow service delivery, poor customer experience, and high operational costs. Customers face difficulty in accessing banking services outside working hours, and there is limited transparency in transaction processing. These challenges highlight the need for a fully automated digital banking system.

2.3. Need of the Project

With the increasing adoption of digital payments and online financial services, there is a strong need for an automated banking platform that provides 24/7 access to banking facilities. Customers expect instant fund transfers, real-time account updates, and secure online transactions.

An Automated Banking Website helps banks reduce manual workload, minimize errors, and improve service efficiency. It enables customers to manage their finances easily from anywhere, saving time and effort. The proposed system fulfills the need for a secure, reliable, and user-friendly digital banking solution.

2.4 .Project Scheduling

The project follows Agile methodology and is divided into multiple sprints. Each sprint delivers a functional module of the banking system.

Sprint No.	Duration	Sprint Goal	Activities	Deliverables
Sprint 1	Week 1	Requirement Analysis	User stories, requirement gathering	Requirement document
Sprint 2	Week 2	System Design	UI design, database design	Design documents
Sprint 3	Week 3	Core Modules	Login, account dashboard	Working modules
Sprint 4	Week 4	Transaction System	Fund transfer, bill payment	Integrated features
Sprint 5	Week 5	Security & Testing	Authentication, testing	Secure system
Sprint 6	Week 6	Deployment	Deployment, documentation	Live system

2.5 Objectives

The main objective of this project is to develop a secure and efficient Automated Banking Website that enables customers to perform banking operations digitally.

The specific objectives are:

- To develop an online platform for banking services
- To provide secure login and authentication
- To enable online fund transfer and bill payment
- To provide real-time transaction history
- To improve customer convenience and satisfaction
- To reduce manual banking operations
- To ensure data security and reliability
- To implement Agile methodology for continuous improvement

3. Software Requirement Specification

The Software Requirement Specification defines the functional and non-functional requirements of the Automated Banking Website. It describes the purpose, scope, hardware and software requirements, tools used, and the software development process model followed for building the system.

3.1 Purpose

The purpose of this project is to design and develop a secure, reliable, and user-friendly Automated Banking Website that allows customers to access banking services digitally. The system aims to provide online banking facilities such as account management, fund transfer, bill payment, and transaction history through a single digital platform.

The platform reduces the dependency on physical bank branches and manual paperwork by automating banking operations. It helps banks improve service efficiency, reduce operational costs, and enhance customer satisfaction. The system ensures secure transactions, real-time updates, and easy access to financial services.

3.2 Scope

The scope of this project includes the development of a web-based Automated Banking Website that supports the following features:

- User registration and secure login
- Account dashboard with balance information
- Fund transfer between accounts
- Online bill payment services
- Transaction history and account statements
- Profile management and password update

The system allows customers to perform banking operations anytime and anywhere using an internet-enabled device. The platform is designed to be scalable, allowing future enhancements such as mobile banking integration, biometric authentication, AI-based financial assistance, and integration with payment gateways.

However, advanced financial services such as investment management, loan processing, and insurance services are not included in the current phase and can be implemented in future versions.

3.3 Hardware Requirements

The following hardware components are required for the development and execution of the Automated Banking Website:

Component	Specification	Description
Processor	Intel Core i3 or above	Required for smooth application execution

Component	Specification	Description
RAM	Minimum 4 GB (8 GB recommended)	Ensures efficient system performance
Hard Disk	Minimum 250 GB free space	Stores application and database
System Type	64-bit Architecture	Supports modern operating systems
Input Devices	Keyboard, Mouse	User interaction
Output Devices	Monitor, Printer (optional)	Display and reports
Internet Connection	Required	Online banking operations

3.4 Software Requirements

The following software components are required for the development and execution of the Automated Banking Website:

Software Component	Specification	Description
Operating System	Windows 10 / Linux / macOS	Platform for application
Programming Language	Java / Python	Backend development
Frontend Technologies	HTML, CSS, JavaScript	User interface
Database	MySQL	Data storage
Server	Apache Tomcat	Web application server
IDE	Visual Studio Code / Eclipse	Development environment
Version Control	Git & GitHub	Code management
Testing Tool	Selenium	Automated testing
Documentation Tool	MS Word / Google Docs	Project documentation

3.5 Tools

The following tools are used during development, testing, and documentation of the Automated Banking Website:

Tool	Purpose
Visual Studio Code / Eclipse	Application development
MySQL Workbench	Database design
Apache Tomcat	Application deployment
Git & GitHub	Version control
Selenium	Automated testing
Postman	API testing
Figma	UI/UX design
Draw.io	System diagrams
Google Chrome / Firefox	Browser testing

3.6 Software Process Model

The Agile Software Process Model is used for the development of the Automated Banking Website. Agile is an iterative and incremental approach that focuses on flexibility, continuous feedback, and fast delivery of working software.

The entire project is divided into multiple development cycles called **sprints**. Each sprint delivers a functional banking module such as user registration, account dashboard, fund transfer, or transaction history. Requirements are collected in the form of user stories and maintained in a product backlog.

Agile practices such as sprint planning, daily stand-up meetings, sprint reviews, and retrospectives are conducted regularly. This approach ensures high software quality, faster development, and continuous improvement of the system.

4 System Design

System design defines the overall architecture of the Automated Banking Website and explains how different components interact with each other. It includes database design, data flow, and user interaction models. The system is designed with a modular approach to ensure security, scalability, and reliability.

The major modules of the system include:

- User Management
- Account Management
- Fund Transfer
- Transaction History
- Bill Payment

The database stores customer details, account information, and transaction records. Secure authentication and validation ensure safe banking operations.

4.1 Data Dictionary

The Data Dictionary defines the core data elements used in the Automated Banking Website. It describes the structure of the main tables required for managing users, accounts, and transactions.

User Table

Field Name	Data Type	Description
user_id	INT (PK)	Unique user ID
name	VARCHAR(100)	Customer name
email	VARCHAR(100)	Email address
password	VARCHAR(100)	Encrypted password
phone	VARCHAR(15)	Contact number

Account Table

Field Name	Data Type	Description
account_id	INT (PK)	Bank account ID
user_id	INT (FK)	Reference to User
account_type	VARCHAR(20)	Savings / Current
balance	DECIMAL	Account balance

Transaction Table

Field Name	Data Type	Description
transaction_id	INT (PK)	Transaction ID
account_id	INT (FK)	Account reference
amount	DECIMAL	Transaction amount
transaction_type	VARCHAR(20)	Debit / Credit
transaction_date	DATE	Date of transaction

4.2 ER Diagram

The ER Diagram shows the relationship between User, Account, and Transaction entities. A user can have multiple accounts, and each account can have multiple transactions.

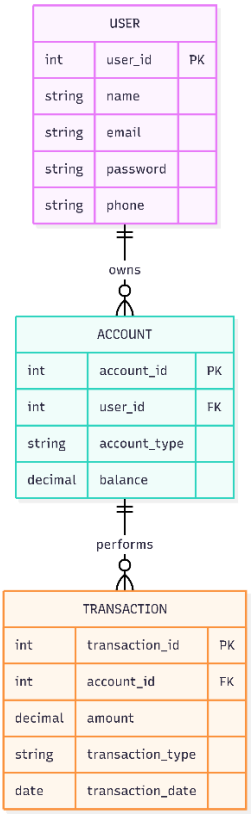

Fig 4.2 ER DIAGRAM

4.3 Data Flow Diagram (DFD)

The Data Flow Diagram shows how data flows between the user, banking processes, and the database. It illustrates how account details and transactions are processed and stored.

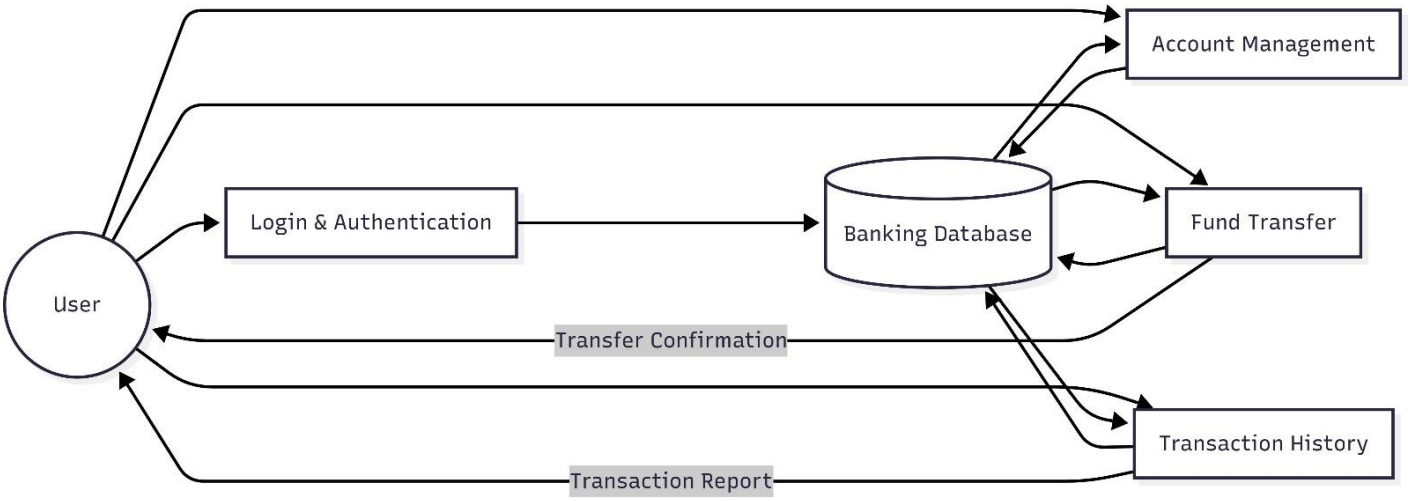

FIG 4.3 DATA FLOW DIAGRAM

4.4 Use Case Diagram

The Use Case Diagram represents the interaction between users and the Automated Banking Website. It shows how customers perform banking operations such as fund transfer, bill payment, and viewing transaction history.

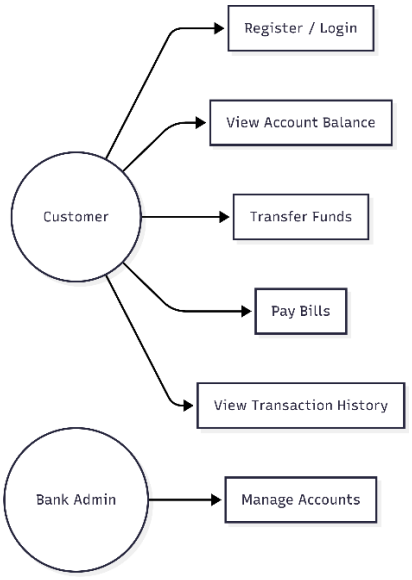

FIG 4.4 USE CASE DIAGRAM

5 Implementation

The implementation phase focuses on converting the system design into a fully functional Automated Banking Website. The application is developed using modern web technologies for the frontend and a secure backend for handling banking operations and database management. The system follows a modular architecture to ensure easy maintenance, scalability, and high security.

The Automated Banking Website is implemented with separate modules for user authentication, account management, fund transfer, transaction processing, and bill payment. Each module is developed independently and later integrated into the main system. This modular approach improves system reliability and allows easy future enhancement.

The frontend of the system is designed using HTML, CSS, and JavaScript to provide a responsive and user-friendly interface. The backend is developed using Java/Python to handle business logic such as login verification, balance checking, fund transfer processing, and transaction validation. The database is implemented using MySQL to store customer information, account details, and transaction records securely.

Security is a major focus during implementation. User authentication is handled using encrypted passwords and session management. Each transaction is validated before execution to prevent fraud and incorrect fund transfers. Role-based access control ensures that only authorized users can access sensitive banking operations.

Exception handling and input validation mechanisms are implemented to prevent system crashes and incorrect data entry. All critical operations such as fund transfer and bill payment generate instant confirmation messages and transaction receipts. The system maintains a complete transaction history for customer reference.

During development, each module is tested individually and then integrated with other modules to ensure smooth functioning. The final system is deployed on a web server, making it accessible to users from any internet-enabled device.

Overall, the implementation ensures that the Automated Banking Website is secure, reliable, efficient, and capable of delivering seamless digital banking services to customers.

6 Testing

Testing is performed to verify that the Automated Banking Website works correctly and meets all functional and security requirements. The main objective of testing is to identify defects, validate system behavior, and ensure that all banking operations such as login, fund transfer, bill payment, and transaction history work accurately and securely.

Different modules of the system are tested using both valid and invalid inputs. Functional testing ensures that each module performs its intended task correctly. Security testing verifies that user authentication and transaction validation mechanisms protect customer data and prevent unauthorized access. Testing helps in improving system reliability, performance, and accuracy before deployment.

6.1 Test Data

Test data is used to validate the correctness of banking operations. Sample customer records and transaction details are entered into the system to verify account handling, fund transfer, and transaction processing.

TEST DATA TABLE

Test Case ID	User Name	Account Type	Transaction Type	Amount	Expected Result
TC01	Rahul	Savings	Fund Transfer	5000	Transfer successful
TC02	Anjali	Current	Bill Payment	1200	Payment successful
TC03	Karan	Savings	Fund Transfer	3000	Transfer successful
TC04	Priya	Savings	Bill Payment	800	Payment successful
TC05	Amit	Current	Fund Transfer	10000	Transfer successful

6.2 Test Result

The test results show that all test cases were executed successfully and the system performed as expected. Login authentication validated users correctly, and only authorized users were able to access account information. Fund transfer and bill payment transactions were processed accurately, and account balances were updated in real time.

The system generated instant transaction confirmations and maintained a complete transaction history for customer reference. Input validation prevented incorrect transaction amounts and unauthorized access. Based on the test execution, the Automated Banking Website is stable, secure, and ready for deployment.

TEST RESULT TABLE

Test Case ID	**Module**	**Test Scenario**	**Expected Result**	**Actual Result**	**Status**
TC01	Login	Login with valid credentials	Login successful	Login successful	Pass
TC02	Fund Transfer	Transfer funds	Transfer successful	Transfer successful	Pass
TC03	Bill Payment	Pay electricity bill	Payment successful	Payment successful	Pass
TC04	Account	View account balance	Balance displayed	Balance displayed	Pass
TC05	History	View transaction history	History displayed	History displayed	Pass

7 User Manual

The User Manual provides step-by-step instructions to help users operate the Automated Banking Website easily and efficiently. The system is designed with a simple, secure, and user-friendly interface so that customers can perform banking operations without visiting physical bank branches.

The Automated Banking Website allows customers to access their bank accounts online, transfer funds, pay bills, and view transaction history at any time. Secure login authentication ensures data privacy and prevents unauthorized access.

7.1 How to Use Project Guidelines

Follow the steps below to use the Automated Banking Website:

- Open the Automated Banking Website in a web browser.
- Register a new account or log in using valid username and password.
- After login, the dashboard will be displayed with available banking services.
- Select **View Account** to check account balance and details.
- Select **Fund Transfer** to transfer money to another account.
- Select **Bill Payment** to pay electricity, mobile, or other utility bills.
- Select **Transaction History** to view past transactions.
- Update profile details if required.
- Log out securely after completing your work.

7.2 Screen Layouts and Description

The Automated Banking Website consists of the following main screens:

- Login/Register Screen: Allows users to create an account and log in securely.
- Dashboard Screen: Displays account summary and banking options.
- Account Details Screen: Shows account balance and account information.
- Fund Transfer Screen: Allows users to enter beneficiary details and transfer funds.
- Bill Payment Screen: Allows users to pay utility bills online.
- Transaction History Screen: Displays all past transactions with date and amount.
- Profile Screen: Allows users to update personal information.

8 Project Applications and Limitations

8.1 Applications

The Automated Banking Website can be widely used in the banking and financial services sector to provide customers with convenient and secure digital banking facilities. The system enables customers to access their bank accounts anytime and perform transactions without visiting physical branches.

The platform is useful for banks, financial institutions, and cooperative banks that want to offer online banking services. It helps reduce manual workload, paperwork, and operational costs. The system improves customer satisfaction by providing quick fund transfers, instant bill payments, and real-time transaction updates.

The Automated Banking Website is also useful for rural and remote areas where access to physical bank branches is limited. Customers can use the platform to manage their finances using internet-enabled devices. The system supports digital payments and promotes a cashless economy.

8.2 Limitations

The current version of the Automated Banking Website requires a stable internet connection to access banking services. Customers without internet access cannot use the platform.

The system does not include advanced financial services such as loan processing, investment management, or insurance services in the current phase. Biometric authentication and mobile banking applications are not supported yet.

The platform depends on external payment gateways and banking servers, so downtime or technical issues may affect service availability. Multi-language support and voice-based banking features are not available in the present version but can be added in future upgrades.

9 Conclusion and Future Enhancement

9.1 Conclusion

The Automated Banking Website project successfully provides a secure and efficient digital platform for modern banking operations. The system simplifies traditional banking processes by allowing customers to access their accounts, transfer funds, pay bills, and view transaction history online. It eliminates the need for customers to visit bank branches for routine banking services, saving time and effort.

The platform ensures data security through secure login authentication and transaction validation mechanisms. Testing confirms that all modules function correctly and deliver accurate and reliable results. The system improves customer convenience, reduces manual workload for banks, and enhances overall service quality.

Overall, this project demonstrates how an automated banking system can transform traditional banking operations into a fast, reliable, and user-friendly digital banking solution.

9.2 Future Enhancement

The following features can be implemented in future versions of the Automated Banking Website:

- Development of mobile banking applications for Android and iOS
- Biometric authentication using fingerprint and face recognition
- AI-based virtual banking assistant for customer support
- Integration with UPI and digital wallet services
- Multi-language support for wider accessibility
- Voice-based banking services
- Real-time fraud detection system
- Cloud-based deployment for scalability and reliability
- Integration with investment and loan management services

10 Bibliography

1. Pressman, R. S., *Software Engineering: A Practitioner's Approach*, McGraw-Hill Education
2. Sommerville, I., *Software Engineering*, Pearson Education
3. Kendall, K. E., & Kendall, J. E., *Systems Analysis and Design*, Pearson Education
4. Laudon, K. C., & Laudon, J. P., *Management Information Systems*, Pearson Education
5. Agile Alliance – Agile Methodology Guide
6. Scrum.org – Scrum Framework Documentation
7. Oracle Java Documentation – Official Website
8. MySQL Reference Manual – MySQL Official Documentation
9. W3Schools – Web Development Tutorials (HTML, CSS, JavaScript)
10. GeeksforGeeks – Programming and Database Tutorials
11. TutorialsPoint – Software Development and Database Tutorials