HCL Project report 4

Automated Banking Website using Agile Methodology

1. Abstract (300–400 Words)

In today's digital era, banking services have rapidly shifted from traditional branch-based operations to automated online platforms. Customers now expect 24×7 access to banking services such as account management, fund transfers, bill payments, and transaction tracking. An Automated Banking Website provides a secure, efficient, and user-friendly solution that eliminates manual processes and reduces dependency on physical bank visits.

This project focuses on the development of an Automated Banking Website using Agile methodology, which emphasizes incremental development, continuous user feedback, and adaptability to changing requirements. Agile methodology divides the development process into short iterations called sprints, ensuring faster delivery, improved quality, and better alignment with user needs. Unlike traditional waterfall models, Agile allows banking systems to evolve with regulatory changes, security updates, and customer expectations.

The system includes essential banking modules such as user registration and authentication, account management, balance inquiry, fund transfer, transaction history, loan information, and administrative controls. Security is a critical aspect of this project, and the system incorporates authentication mechanisms, role-based access, and secure transaction handling.

Administrators can manage customer accounts, monitor transactions, generate reports, and handle system configurations. Customers can securely log in, view account details, transfer funds, pay bills, and track their financial activities. Agile practices such as sprint planning, daily stand-ups, sprint reviews, and retrospectives ensure continuous improvement, transparency, and collaboration throughout development.

In conclusion, this project demonstrates how Agile methodology can be effectively applied to build a secure, scalable, and reliable Automated Banking Website. The system enhances customer convenience, reduces operational costs, and improves banking efficiency. Its modular and flexible architecture

allows future enhancements such as mobile banking, AI-based fraud detection, and integration with third-party financial services.

---

2. Introduction

2.1 Introduction

Banking systems play a vital role in economic growth and financial stability. With the advancement of internet technologies, customers demand fast, secure, and convenient access to banking services. An Automated Banking Website enables customers to perform banking operations online without visiting a bank branch.

Traditional banking systems involve manual paperwork, long waiting times, and limited service availability. These limitations often result in inefficiencies, errors, and customer dissatisfaction. An automated online banking platform integrates multiple banking services into a single system, providing seamless access to users.

This project adopts Agile methodology to ensure flexibility, iterative development, and continuous feedback. Agile allows developers to respond quickly to changes in banking regulations, security requirements, and customer needs, making it ideal for financial systems.

2.2 Problem Identification

The following challenges exist in traditional or semi-automated banking systems:

- Limited banking hours and dependency on physical branches
- Manual processes leading to errors and delays
- Lack of real-time transaction updates
- Difficulty in handling large customer data securely
- Inflexible systems that cannot adapt to changing requirements
- Security risks in legacy systems

These problems highlight the need for a secure and fully automated online banking solution.

2.3 Need of the Project

The need for an Automated Banking Website arises due to:

- Increasing demand for online banking services
- Need for secure and real-time financial transactions
- Reduction of manual workload and operational costs
- Improved customer convenience and satisfaction
- Centralized management of customer and transaction data
- Agile-based development for continuous enhancement

2.4 Project Scheduling

The project is developed using Agile methodology and divided into sprints:

- Sprint 1: Requirement gathering and system design
- Sprint 2: User registration, login, and authentication
- Sprint 3: Account management and balance inquiry
- Sprint 4: Fund transfer and transaction history
- Sprint 5: Admin module, testing, and deployment

Each sprint includes planning, development, testing, review, and retrospective phases.

2.5 Objectives

The objectives of the project are:

- To develop a secure Automated Banking Website
- To apply Agile methodology for iterative development
- To provide real-time banking services to users
- To reduce manual banking operations
- To ensure data security and system reliability
- To deliver a scalable and maintainable solution

---

3. Software Requirements Specification (SRS)

Software Requirements Specification for

Automated Banking Website using Agile Methodology

Version: 1.0 Approved
Prepared by: Arushi
Organization: Academic Project
Date Created: January 2026

SRS Table of Contents

Revision History

Version	Date	Description	Author
1.0	Jan 2026	Initial SRS creation	Arushi

Overview

This SRS document defines the functional and non-functional requirements of the Automated Banking Website. It serves as a reference for developers, testers, and stakeholders.

Intended Audience

• Developers
• Testers
• Project Managers
• System Analysts
• End Users

II. Overall Description

Product Perspective

The Automated Banking Website is a web-based application developed using Agile methodology. It interacts with users through web browsers and stores data in a centralized database.

Product Functions

- User registration and secure login
- Account details and balance inquiry
- Fund transfer between accounts
- Transaction history and statements
- Admin management of users and transactions
- Secure authentication and authorization

User Classes

- Admin: Manages users, accounts, and transactions
- Customer: Uses banking services
- System: Handles validations and security

Operating Environment

- OS: Windows / Linux
- Browser: Chrome, Firefox, Edge
- Server: Cloud or Local Server
- Database: MySQL / PostgreSQL

---

4. System Design

4.1 Data Dictionary

Data Item	Description
User_ID	Unique customer identifier
Account_No	Bank account number
Balance	Account balance

Data Item	Description
Transaction_ID	Unique transaction record
Amount	Transaction amount

4.2 ER Diagram

Entities include:
- User
- Account
- Transaction
- Admin

Relationships:
- User owns Account
- Account generates Transactions

4.3 Data Flow Diagram (DFD)

- User inputs login credentials
- System validates credentials
- User performs banking operations
- Data stored and retrieved from database

4.4 Diagrams Used

- Use Case Diagram
- Class Diagram
- Sequence Diagram
- Activity Diagram

---

5. Implementation

5.1 Program Code

The system is implemented using:
- Frontend: HTML, CSS, JavaScript

• Backend: Java / Python / Node.js
• Database: MySQL

Modules are developed incrementally in Agile sprints.

5.2 Output Screens

• Login Page
• User Dashboard
• Account Summary Page
• Fund Transfer Page
• Transaction History Page
• Admin Dashboard

---

6. Testing

6.1 Test Data

Test Case	Input	Expected Output
Login	Valid credentials	Successful login
Transfer	Valid amount	Transaction success

6.2 Test Result

All modules were tested successfully. Errors were fixed during sprint reviews, ensuring system reliability.

---

7. User Manual

7.1 How to Use the System

1. Open the website
2. Register or login
3. Access dashboard
4. Perform banking operations

5. Logout securely

7.2 Screen Layouts and Description

Each screen is designed to be user-friendly, responsive, and secure.

---

8. Project Applications and Limitations

Applications

• Online banking services
• Financial institutions
• Educational banking simulations

Limitations

• Internet dependency
• Limited to web platform
• Advanced security features not included

---

9. Conclusion and Future Enhancement

The Automated Banking Website successfully demonstrates the use of Agile methodology in developing a secure and efficient banking system. It improves customer convenience and operational efficiency.

Future Enhancements:
• Mobile banking application
• Biometric authentication
• AI-based fraud detection
• Integration with UPI and digital wallets

---

10. Bibliography & References

• Pressman, Software Engineering
• Agile Manifesto
• IEEE SRS Standards

- Banking System Documentation
- Online Research Articles