

Experiment 1:- Comparative Study of SDLC Models

Objective: Compare and contrast different SDLC models (Waterfall, Prototyping, Evolutionary, RAD, Spiral).

Procedure: Research each model, create a comparative chart highlighting their features, advantages, and disadvantages, and discuss scenarios where each model is most suitable.

Expected Outcome: Understanding of the strengths and weaknesses of various SDLC models and their appropriate application

1. Introduction

A Loan Management System (LMS) is a software-based application designed to automate and manage loan-related operations in banks, financial institutions, and organizations. The system provides a structured and efficient way to handle loan applications, approvals, disbursements, and repayments.

The system maintains borrower records, loan details, repayment schedules, and outstanding balances in a centralized database. It also helps staff track active loans, overdue payments, and generate financial reports.

Additionally, the Loan Management System improves transparency by allowing administrators to easily monitor loan status and payment history. Secure access control ensures that sensitive financial data is accessible only to authorized users.

Overall, the Loan Management System increases efficiency, reduces paperwork, minimizes errors, and ensures accurate handling of loan data.

2. Objectives of Loan Management System

The main objectives of the system are:-

- To automate loan application and processing
- To maintain customer and loan records
- To calculate EMI and outstanding balance automatically
- To generate loan statements and reports
- To reduce manual work and human errors

3. Major Modules of the System

3.1 Customer Module

- Apply for loan
- View loan details
- Check repayment schedule
- View payment history

3.2 Loan/Accounts Module

- Add and manage loan records
- Approve or reject loans
- Update payment details
- Generate loan statements
- Track overdue loans

3.3 Admin Module

- Manage users
- Define loan types and interest rates
- Generate reports
- Ensure data security

4. Requirements Analysis

Hardware Requirements

- Dual Core Processor or higher
- Minimum 4 GB RAM
- 500 MB free disk space
- Keyboard, Mouse, Monitor

Software Requirements

- Operating System: Windows / Linux
- Application Type: Web-based
- Programming Language: HTML / CSS / JS / PHP
- Database: MySQL / SQLite
- Development Tools: Eclipse / NetBeans / Visual Studio Code

5. Software Development Model

The Spiral Model is selected for developing the Loan Management System because it is a risk-driven and iterative model. The system involves financial data, security requirements, and complex calculations such as EMI and interest. Spiral model helps in identifying risks early and improving the system step by step.

5.1 Justification for Spiral Model

1. High Risk Project

Loan Management System handles sensitive financial information. Spiral model focuses on identifying and reducing risks in every iteration.

2. Changing Requirements

Interest rates, loan policies, repayment schedules, and penalty rules may change frequently. The Spiral Model easily supports these changes because development is done in iterations, allowing modifications without disturbing the entire system.

3. Continuous Testing

In Spiral Model, testing is performed in every phase of development. This continuous testing process helps in identifying errors early, reducing system failures, and improving overall software reliability.

4. User Feedback

After each development cycle, users can review the working version of the system and provide feedback. This feedback is used to improve functionality, usability, and performance of the system.

5. Better Quality Product

Since the system is developed through repeated cycles of planning, development, and testing, the final product becomes more reliable, secure, and better aligned with user needs.

5.2 Advantages of Spiral Model

- Early risk detection
- High reliability
- Flexible to changes
- Continuous testing
- Suitable for large systems

5.3 Limitations of Spiral Model

- Costly
- Complex to manage
- Requires expert team

5.4 Future Enhancements

- Online loan application
- SMS / Email alerts
- Mobile app
- Banking system integration

Conclusion

After analysing various SDLC models, it is concluded that the **Spiral Model** is the most suitable model for the Loan Management System because it effectively manages risks, supports changes, and ensures high security and quality.

Experiment 2:- Creating a Software Requirements Specification (SRS) Document

Objective:- Develop a comprehensive SRS document for a hypothetical software project.

Procedure:- Conduct requirement analysis, distinguish between functional and non-functional requirements and document them according to IEEE standards.

Expected Outcome:- A well-documented SRS that includes all the requirements and specifications for a software project.

Result:-

Software Requirements Specification (SRS)

For: Loan Management System

Version: 1.0

Date: 31 January, 2026

Prepared By: Kartik Khatri

1. Introduction

1.1 Purpose

The purpose of this document is to describe the functional and non-functional requirements of the Loan Management System (LMS). This system is intended to manage loan processing, repayment tracking, and report generation in an organization.

1.2 Scope

The Loan Management System is a software application designed to automate loan management activities in banks and financial institutions. The system helps in maintaining customer records, managing loan details, tracking repayments, generating loan statements, and producing reports.

The system will serve the following users:

- Administrator – manages users, loan policies, and system settings
- Loan Officer/Clerk – processes loans and records payments
- Customer – views loan details and payment status

The main goal of the system is to reduce manual work, avoid calculation errors, improve transparency, and ensure secure handling of financial data.

1.3 Definitions, Acronyms and Abbreviations

- LMS – Loan Management System
- SSL – Secure Socket Layer
- UI – User Interface
- DFD – Data Flow Diagram
- ERD – Entity Relationship Diagram

1.4 References

- IEEE Std 830-1998 (IEEE Recommended Practice for Software Requirements Specifications)
- UML 2.5 Standard Documentation
- PHP & MySQL Development Guides

1.5 Overview

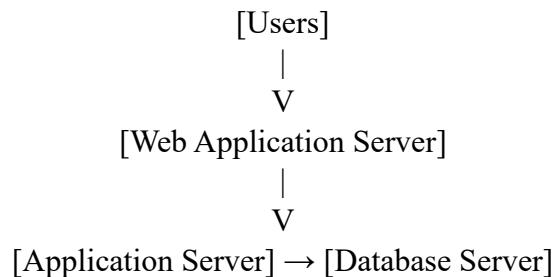
This document provides an overall description of the Loan Management System, including system features, functional and non-functional requirements, system models, and necessary details for development.

2. Overall Description

2.1 Product Perspective

The LMS will be an independent web-based application accessible through modern web browsers. It will be connected to a centralized database server for storing and managing data.

High-Level Architecture Diagram (Text Representation)



2.2 Product Functions

Major Functions:

- Customer registration and record management
- Loan application and approval
- EMI calculation
- Payment recording
- Loan statement generation
- Report generation
- User authentication and authorization

2.3 User Classes and Characteristics

Administrator: Full system access, manages users, loan policies, and reports.

Loan Officer/Clerk: Handles loan processing and payments.

Customer: Limited access, views loan status and payment history.

2.4 Operating Environment

- **Platform:** Web-based (HTML, CSS, Javascript, PHP, MySQL)
- **Server:** Apache / Nginx
- **Database:** MySQL 8+
- **Browsers:** Chrome, Firefox, Edge

2.5 Design and Implementation Constraints

- Secure handling of financial data is mandatory
- Role-based access control required
- Internet or Intranet connectivity required
- Must follow organizational policies

2.6 Assumptions and Dependencies

- Every user has unique login credentials
- Organization provides required hardware and network facilities

3. Specific Requirements

3.1 Functional Requirements

FR1: User Authentication & Authorization

- Users must log in using username and password
- Role-based access control

FR2: Customer Management

- Add, update, delete customer details
- Assign unique customer ID

FR3: Loan Management

- Add new loan
- Approve or reject loan

FR4: EMI Calculation

- Automatic EMI calculation

FR5: Payment Tracking

- Store payment records
- Display due amount

FR6: Report Generation

- Daily and monthly loan reports
- Customer-wise reports

3.2 Non-Functional Requirements**NFR1: Performance**

- Response time \leq 2 seconds

NFR2: Security

- SSL encryption
- Password hashing

NFR3: Usability

- Simple and user-friendly interface

NFR4: Reliability

- System uptime \geq 99%

NFR5: Maintainability

- Modular design

4. System Models**4.1 Use Case Diagram (Text Representation)**

Actors:

- Admin
- Loan Officer
- Customer

Use Case:

Admin: Manage users, manage loan policies, generate reports

Loan Officer: Process loans, record payments

Customer: View loan status, download statement

4.2 Data Flow Diagram (DFD)

Level 0:

Users → LMS → Database

Level 1 (Loan Processing Example):

- Input: Customer ID
- Process: Validate data and store information
- Output: Loan approval and updated status

4.3 ER Diagram (Text Representation)

Entities:

- Customer(Customer_ID, Name, Address)
- Loan(Loan_ID, Amount, Interest_Rate)
- Payment(Payment_ID, Amount, Date)
- User(User_ID, Role, Username, Password)

Relationships:

- Customer ↔ Loan (One to Many)
- Loan ↔ Payment (One to Many)

5. Data Dictionary

Entity	Attribute	Description	Datatype
Customer	Customer_ID	Unique customer identifier	Integer
Loan	Loan_ID	Unique Loan identifier	Integer
Payment	Payment_ID	Unique payment identifier	Integer

6. Appendices

- **Gantt Chart** – Development schedule
- **Hardware Requirements** – Server, PCs, networking
- **Test Cases** – Sample functional & security tests

Expected Outcome:

This SRS document follows IEEE standards and clearly defines all functional and non-functional requirements for the Loan Management System. It provides sufficient details for system design and implementation.

Kartik Khatri

MCA – G2

Roll No. – 44