UE22CS341A: Software Engineering
Case Study

Unit 1 Deliverable

Software Requirements Specification (SRS) for Life Insurance corperation.

Synopsis :

The LIC Insurance Management System aims to revolutionize the way insurance policies are managed by automating and streamlining processes within the Life Insurance Corporation of India (LIC). This project focuses on developing a robust, secure, and user-friendly system that facilitates the management of customer data, policy details, claim processing, payments, and agent interactions. The system will replace legacy systems and manual operations, enabling LIC to enhance efficiency, reduce errors, and improve overall customer satisfaction.

This system will provide essential functionalities, including customer registration, policy management and claims processing. Customers will be able to view and update their profiles, link policies and file claims. The system will also allow agents to manage customer relationships. With integrated real-time reporting features, LIC administrators can generate detailed reports on policies, claims making data-driven decisions easier and faster.

Security and scalability are central to the design of this system. The LIC Insurance Management System will incorporate encryption and role-based access controls to ensure data privacy and protection against unauthorized access. Furthermore, the system will be designed to handle a large volume of concurrent users, ensuring that operations continue smoothly even during peak times. The system will also comply with the Insurance Regulatory and Development Authority of India (IRDAI) guidelines, ensuring all processes are aligned with industry standards and regulations.

In addition to functional capabilities, the system will provide a responsive, user-friendly interface accessible via web browsers and mobile devices. With a focus on maintainability and modularity, the system can be easily updated as new requirements emerge. Ultimately, the LIC Insurance Management System will be a comprehensive solution that enhances LIC's operational efficiency, delivers a superior customer experience, and ensures compliance with legal and regulatory requirements.

1. Introduction

1.1 Purpose

The purpose of this document is to provide a detailed description of the LIC Insurance Policy Management System. This document will outline the functional and non-functional requirements, system design constraints, and other critical aspects of the system. It serves as a guideline for developers, testers, and stakeholders involved in the project.

1.2 Scope

The LIC Insurance Policy Management System is designed to automate and streamline the management of insurance policies. The system will cover various functionalities, including policyholder management, insured person details, policy management, agent management, nominee management, policy document storage, and payment processing.

1.3 Definitions, Acronyms, and Abbreviations

LIC: Life Insurance Corporation

SRS: Software Requirements Specification

Policyholder: The person who owns the insurance policy.

Insured Person: The individual whose life or health is covered by the insurance policy.

Agent: An individual who sells and manages insurance policies.

Nominee: The person designated to receive policy benefits upon the policyholder's death.

Payment: Financial transactions related to policy premiums and claims.

1.4 References

- IEEE Standard for Software Requirements Specifications
- LIC Official Guidelines and Documentation

1.5 Overview

This SRS document includes a comprehensive description of the system's functionalities, interfaces, performance requirements, and design constraints. It provides a clear understanding of the system requirements for all stakeholders.

2. Overall Description

2.1 Product Perspective

The LIC Insurance Management System is a web-based application that will integrate with existing financial systems and customer databases. It will replace legacy systems and manual processes, offering a unified interface for managing policies, claims, payments, and agent interactions.

2.2 Product Functions

- **Customer Management**: Registration, profile management, and policy linking.
- **Policy Management**: Creation, update, and tracking of various insurance policies.
- **Claim Management**: Filing, processing, and tracking claims.
- **Agent Management**: Agent registration, policy assignment, and commission calculation.

2.3 User Classes and Characteristics

- **Customers**: Individuals who purchase insurance policies. They can view and manage their policies, make payments, and file claims.
- **Agents**: LIC representatives who sell policies. They manage customer relationships and track sales.
- **Administrators**: LIC employees who manage the system and ensure data accuracy and compliance.
- **Claims Processors**: Staff responsible for processing and approving insurance claims.

2.4 Operating Environment

- **Client Side**: Web browsers (Chrome, Firefox, Edge)
- **Server Side**: Linux-based web server, MySQL/MariaDB database
- **Security**: role-based access control

2.5 Design and Implementation Constraints

- **Regulatory Compliance**: The system must comply with IRDAI regulations.
- **Scalability**: The system should handle concurrent users during peak times.
- **Data Security**: Sensitive data must be encrypted and access controlled.

2.6 Assumptions and Dependencies

- **Internet Access**: Users must have a stable internet connection.

3. External Interface Requirements

3.1 User Interfaces

Admin Dashboard: For managing users, policies, and payments.

Agent Portal: For viewing and managing policies .

Policyholder Portal: For viewing policy details, making payments, and updating personal information.

Nominee Access: Limited access for nominees to view specific policy details.

3.2 Hardware Interfaces

The system will require standard servers and networking hardware compatible with LIC's current infrastructure.

3.3 Software Interfaces

The system will integrate with existing LIC customer management and financial systems.

3.4 Communication Interfaces

The system will use HTTPS for secure web-based communication.

4 Non-Functional Requirements

4.1 Performance Requirements

- The system shall handle up to 10,000 concurrent users.
- The system shall process payments and updates within 3 seconds.

4.2 Security Requirements

- The system shall use SSL encryption for all data transmission.
- User authentication and role-based access control must be implemented.
- Data should be encrypted at rest and in transit.

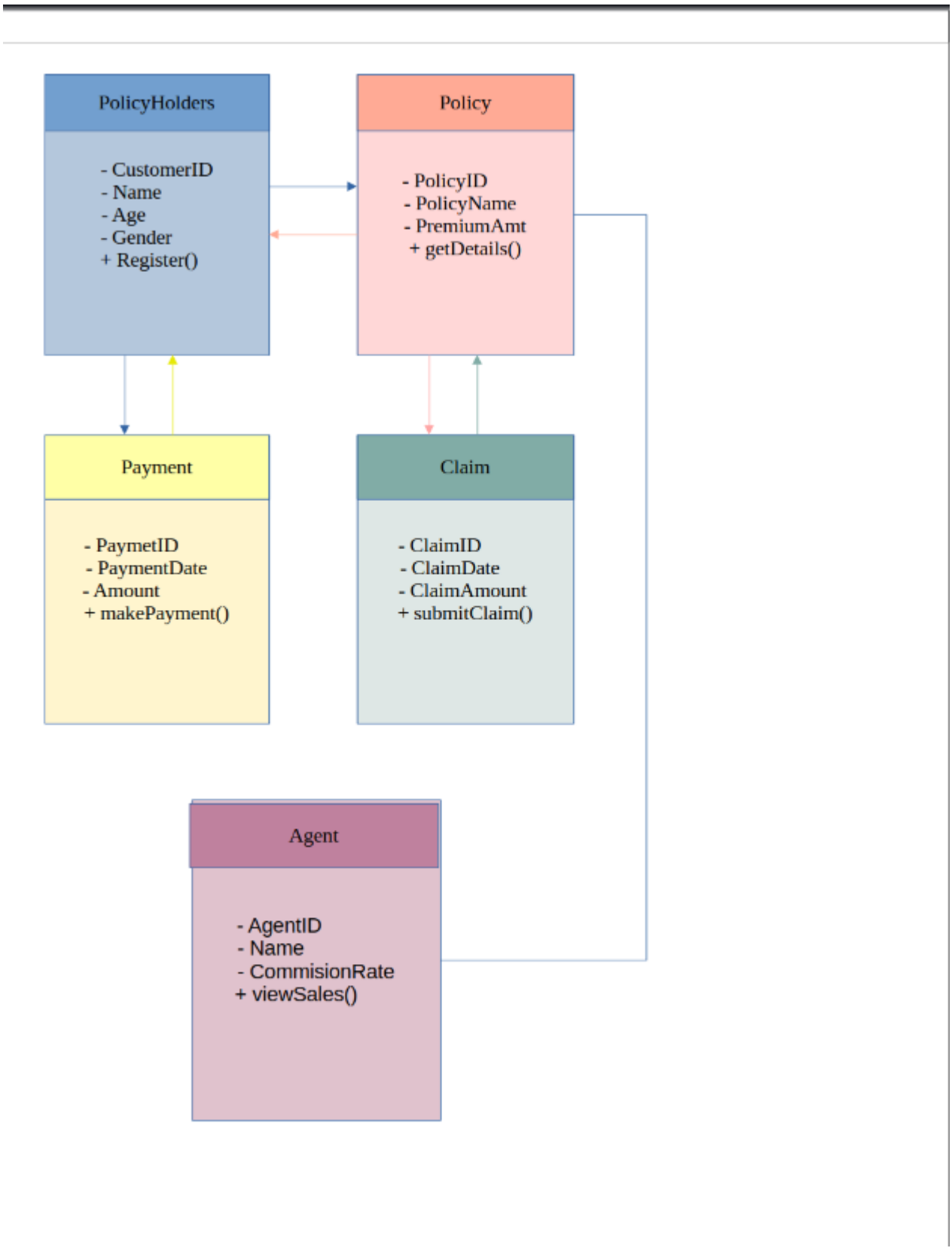
4.3 Usability Requirements

- The system interface shall be intuitive and easy to use for all user classes.
- The system shall provide help documentation and customer support contact details.

4.4 Reliability Requirements

- The system shall have a minimum uptime of 99.9%.
- The system shall have failover capabilities in case of server failures.

5. <u>Analysis Models [UML model]</u> :



6. System Features

6.1 System Feature 1: Policyholder Management

Description: Manage policyholder information, including creation, update, and deletion of policyholder profiles.

Priority: High

Acceptance Criteria: The system must allow administrators and agents to create, update, and manage policyholder details, ensuring data accuracy and uniqueness.

6.2 System Feature 2: Insured Person Management

Description: Manage details of insured persons, including personal information and health data.

Priority: Medium

Acceptance Criteria: The system must allow the creation and updating of insured person profiles linked to the correct policyholder.

6.3 System Feature 3: Policy Management

Description: Create, update, and manage insurance policies, including details such as policy type, coverage, and premium amounts.

Priority: High

Acceptance Criteria: The system must support policy creation and updates, calculate premiums accurately, and manage policy status effectively.

6.4 System Feature 4: Agent Management

Description: Manage agent profiles and monitor their performance and sales.

Priority: Medium

Acceptance Criteria: The system must allow the addition, updating, and deletion of agent profiles and provide performance metrics.

6.5 System Feature 5: Nominee Management

Description: Manage nominee information associated with each policy, including personal and contact details.

Priority: Medium

Acceptance Criteria: The system must enable the addition and updating of nominee information, ensuring it is linked to the respective policyholder.

6.6 System Feature 7: Payment Processing

Description: Handle premium payments, process claims.

Priority: High

Acceptance Criteria: The system must validate payments, update payment statuses

7. Other Requirements

7.1 Database Requirements

The system shall use a relational database (e.g., MySQL) to store all policy, user, and transaction data.

7.2 Compliance Requirements

The system shall comply with local and international data protection laws and regulations.

Appendices

Appendix A: Glossary

Policyholder: Owner of the insurance policy.

Insured Person: Individual covered by the policy.

Agent: Individual who manages insurance policies.

Nominee: Person designated to receive policy benefits.

Payment: Transactions for premiums and claims.

Appendix B: Field Layouts

Database Tables: Layouts of database tables including fields, data types, and relationships.

UI Forms: Layouts of web forms for policy management, payments.

Appendix C: Requirement Traceability Matrix

Description: A matrix mapping each requirement to its design and implementation components, ensuring complete coverage and tracking throughout development.

Requirement ID	Requirement Description	Design Specification ID	Implementation Module/Code	Test Case ID	Verification Method	Comments
FR-001	The system must enable agents to create, update, and manage policies.	DS-001	IM-001 (PolicyManagement.js)	TC-001	Functional Test	
FR-002	Administrators must be able to manage agent profiles and performance.	DS-002	IM-002 (AgentManager.php)	TC-002	Functional Test	
FR-003	The system must securely store and manage policy documents.	DS-003	IM-003 (DocumentStorage.java)	TC-003	Security Test	

Requirement ID	Requirement Description	Design Specification ID	Implementation Module/Code	Test Case ID	Verification Method	Comments
FR-004	The system must process payments for premiums and claims securely.	DS-004	IM-004 (PaymentProcessor.cs)	TC-004	Security Test	
FR-005	The system must support multi-factor authentication for all users.	DS-005	IM-005 (MFAService.py)	TC-005	Security Test	
NFR-001	The system must be deployable on both Linux and Windows environments.	DS-06	IM-06 (DeploymentScripts.sh)	TC-06	Portability Test	
NFR-002	The system should have a modular codebase that supports easy updates and maintenance.	DS-07	IM-07 (ModuleManager.java)	TC-07	Maintainability Test	
BR-001	All policy changes must be reviewed and approved by an administrator.	DS-013	IM-013 (ApprovalWorkflow.php)	TC-013	Functional Test	
BR-002	The system must comply with insurance regulations and standards.	DS-014	IM-014 (ComplianceChecker.js)	TC-014	Compliance Test	Legal review required

- PES1UG22CS178 [Devendra S Alase]

- PES1UG22CS151 [Venkat Satya Nikhil]

- Section C