# Software Requirements Specification (SRS) Template

**Project: Insurance workflow automation software (IWAS)**  
**Version:** 1.0  
**Authors:**

* **Lalithadithya N**
* **Khushi Kogganur**
* **Kashish K S**
* **Keerti Kallugadde**

**Date:** 31-08-2025  
**Status:** Draft

## Revision history

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Version | Date | Author | Change summary | Approval |
| 1.0 | 31-08-2025 | Lalithadithya N | Initialize SRS |  |

## Approvals

|  |  |  |  |
| --- | --- | --- | --- |
| Role | Name | Signature / Email | Date |
| Course Coordinator |  |  |  |

## Table of Contents

1. Introduction  
2. Overall description  
3. External interfaces  
4. System features (detailed)  
5. Non-functional requirements (detailed)  
6. Quality attributes & Acceptance tests  
7. UML Use-Case Diagram  
8. Requirements Traceability Matrix (RTM)

## 1. Introduction

1.1 Purpose

This document is a Software Requirements Specification (SRS) for an Insurance workflow automation system. It defines functional and non-functional requirements, interfaces, and verification criteria intended for instructors and students to use as a reference.

1.2 Scope  
Automates end‑to‑end insurance business processes across new business intake, underwriting, policy administration, billing, claims, endorsements, renewals, and compliance. Core capabilities include:

* + - Workflow orchestration & task management
    - Rules engine for straight‑through processing (STP)
    - Document & e‑signature handling
    - Integrations with rating, payment gateways, CRM, DMS
    - Notifications (email/SMS/push) & SLA tracking
    - Reporting, dashboards, and audit trails

1.3 Audience  
Developers, QA Engineers, System Integrators, Maintenance Technicians, and Assessment Evaluators.

1.4 Definitions

**List of Acronyms**

* + - **PAS: Policy Administration System**
    - **U/W**: Underwriting
    - **STP**: Straight‑Through Processing
    - **FNOL**: First Notice of Loss
    - **KYC**: Know Your Customer
    - **DMS**: Document Management System
    - **SLA**: Service Level Agreement
    - **PII/PHI**: Personally Identifiable / Protected Health Information

## 2. Overall description

2.1 Product perspective  
The IWAS is a modular web application with API‑first architecture. It sits between customer‑facing channels and core systems (PAS, billing, claims) to orchestrate and automate workflows

2.2 Major product functions (detailed)

* + - Workflow orchestration & task management
    - Rules engine for straight‑through processing (STP)
    - Document & e‑signature handling
    - Integrations with rating, payment gateways, CRM, DMS
    - Notifications (email/SMS/push) & SLA tracking
    - Reporting, dashboards, and audit trails

2.3 User roles and characteristics (expanded)  
 - **Prospect/Customer**: Submits applications, documents, e‑signs, receives notifications.

**- Agent/Broker**: Creates quotes, submits applications, manages endorsements/renewals.

**- Underwriter**: Reviews referrals, pricing, risk decisions, binds policies.

**- Claims Adjuster**: Handles FNOL, investigation, reserves, settlements.

**- Operations (Policy Admin)**: Issues policies, endorsements, cancellations, reinstatements.

**- Billing Specialist**: Manages invoices, dunning, refunds, write‑offs.

**- Compliance/QA**: Conducts audits, ensures regulatory adherence.

**- Supervisor/Manager**: Monitors SLAs, workload balancing, approves exceptions.

**- System Admin**: Manages users, roles, configuration, integrations.

2.4 Operating environment  
- Kubernetes‑backed containers; Linux nodes; managed PostgreSQL/SQL Server; object storage for documents; managed queue (e.g., SQS/Kafka).

- **Web**: Chromium‑based/Firefox browsers

2.5 Constraints  
 - Security & privacy by design; encryption at rest/in transit; role‑based access control (RBAC).

- Regulatory archiving & retention policies (jurisdiction‑specific).

- Zero‑downtime deploys for minor releases.

## 3. External interface requirements

3.1 User interfaces  
Primary UI: touch-enabled screen with clear menu structure. Accessibility: screen-reader friendly prompts for visually impaired users and high-contrast mode.

3.2 Hardware interfaces  
- EMV card reader (ISO/IEC 7816)  
- PIN keypad (secure pin entry)  
- Cash dispenser module (vendor-specific interface)  
- Deposit acceptor  
- Receipt printer  
- Status LEDs and operator panel

3.3 Software interfaces  
- Bank Core API (authorized JSON over TLS) for account validation, debit/credit, and settlement.  
- Monitoring & Logging API for health checks and alerts.  
- Optional third-party services (cash recycler monitoring).

3.4 Communications  
- TLS 1.2+ enforced, mutual TLS for bank core where required.  
- Retry/backoff strategy and offline queuing with secure signing for offline transactions.

<< Make sure overall there are at least 15 FRs for overall project, 5 NFRs, 2 security objectives and 5 Security requirements>>

## 4. System features (detailed)

Each requirement below includes acceptance criteria and a reference test case. IDs follow ATM-F-###.

### 4.1 Authentication

Description: Authenticate the cardholder using EMV card and PIN; support chip & contactless where applicable.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Req ID | Requirement | Type | Priority | Source/Stakeholder | Acceptance criteria / Test case ref | Comments / Dependencies |
| ATM-F-001 | The system shall validate card authenticity (EMV) and verify PIN via bank core. | Functional | High | Security / Banking | AC-ATM-F-001: Valid EMV card and correct PIN leads to authenticated session. Test: TC-Auth-01 | Requires EMV reader and Core API |
| ATM-F-002 | The system shall mask PIN entry and not log PIN characters. | Functional | High | PCI-DSS | AC-ATM-F-002: PIN masked on UI and absent from logs. Test: TC-Auth-02 | Audit review required |
| ATM-F-003 | The system shall lock the card after 3 consecutive failed PIN attempts and record an audit event. | Functional | High | Security | AC-ATM-F-003: After 3 failures, further attempts are blocked and message displayed. Test: TC-Auth-03 | Card must be flagged in Core |

### 4.2 Cash Withdrawal

Description: Dispense cash in validated denominations and ensure ledger consistency with bank core.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Req ID | Requirement (shall...) | Type | Priority | Source/Stakeholder | Acceptance criteria / Test case ref | Comments / Dependencies |
| ATM-F-010 | The system shall verify sufficient available balance and daily limit before dispensing cash. | Functional | High | Business | AC-ATM-F-010: Dispense occurs only if balance >= amount and within limits. Test: TC-WD-01 | Core API check required |
| ATM-F-011 | The system shall dispense requested amount using available denominations and update cash cassette counts. | Functional | High | Operations | AC-ATM-F-011: Dispensed amount equals requested; cassette counts decremented. Test: TC-WD-02 | Hardware interaction with dispenser |
|  |  |  |  |  |  |  |

### 4.3 Cash/Check Deposit

Requirements: The system shall accept deposits (envelope or bulk acceptor), validate amounts where possible, and queue offline transactions if network unavailable. IDs: ATM-F-020..ATM-F-025.

### 4.4 Funds Transfer

Requirements: Support intra-bank transfers between own accounts with confirmation and receipt. Reject transfers exceeding configured limits. IDs ATM-F-030..ATM-F-032.

### 4.5 Maintenance & Monitoring

Requirements: Provide a technician menu for diagnostics, remote monitoring API integration, secure admin authentication, and encrypted log shipping. IDs ATM-F-040..ATM-F-045.

## 5. Non-functional requirements (detailed)

NFRs below are measurable and tied to test plans. IDs ATM-NF-###.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Req ID | Requirement | Category | Priority | Acceptance criteria / Measurement |
| ATM-NF-001 | Overall transaction response (from card insertion to end) shall be ≤ 5 seconds for 90% of transactions under normal load. | Performance | High | 90th percentile ≤ 5s in production-like test. Test: TC-Perf-01 |
| ATM-NF-002 | System shall provide 99.9% availability monthly; scheduled maintenance windows excluded. | Reliability | High | Uptime reports show ≥99.9% per month. Test: Ops reports. |
| ATM-NF-003 | All sensitive cardholder data must comply with PCI-DSS; PAN must not be stored in plaintext. | Security/Compliance | High | PCI-DSS audit checklist pass. Test: TC-Sec-01 |
| ATM-NF-004 | System shall provide logs with timestamped events retained for 7 years for audit purposes. | Audit/Data Retention | High | Automated archival and retrieval verified. Test: TC-OPS-01 |
| ATM-NF-005 | System shall support accessibility features (screen reader, high contrast) conforming to WCAG 2.1 AA where applicable. | Usability/Accessibility | Medium | Accessibility audit pass. Test: TC-UX-01 |

## 5.1. Security

## 5.1.1 Security Objectives

This section will list 2 to 4 security objectives for the project.

## 5.1.2 Security Requirements

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Req ID | Requirement (shall...) | Type | Priority | Acceptance criteria / Test case ref |
| PRJ-SR-001 | TLS 1.2+ mandatory for all network connections. | Security | High |  |

## 6. Quality attributes & Acceptance tests

- Exit criteria for acceptance: All high-priority functional requirements implemented and verified, no critical NFR failures, and RTM shows all test cases passed.

- Acceptance test suites: Authentication, Withdrawal, Deposit, Transfer, Performance, Security, and Accessibility tests.

## 7. System models and diagrams

## 7.1 UML Use-Case diagram

At least 2 UML use case diagrams to be created.

## 8. Requirements Traceability Matrix (RTM)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Req ID | Requirement short | Section ref / Design Spec | Module | Test case(s) | Status (N/P/A) | Comments |
| ATM-F-001 | Validate PIN | 4.1 / DS-Auth-01 | AuthModule | TC-Auth-01 | N |  |
| ATM-F-010 | Dispense cash | 4.2 / DS-Dispense-01 | DispenseModule | TC-WD-01, TC-WD-02 | N |  |
| ATM-NF-001 | Response time target | 5 / DS-Perf-01 | WebUI / CoreAPI | TC-Perf-01 | N |  |