Software Requirements Specification For Simple Legal Information Retrieval System

Version 1.1

Group 15

Revision History

Date	Version	Description	Author
02/08/2025	1.0	Initial SRS	Nishan Kavinda
06/08/2025	1.1	Updated Appendix and tables	Nishan Kavinda

Table of Contents

Software Requirements Specification

1. Introduction

1.1 Purpose

This document defines the software requirements for the Simple Legal Information Retrieval System for Sri Lanka. It aims to serve as a reference for developers, testers, and stakeholders, defining what the system must do, how it will interact with users and external systems, and the constraints under which it must operate.

The primary goal of the system is to democratize access to legal information in both Sinhala and English using AI-driven tools such as LLMs, semantic search, and recommendation engines.

1.2 Scope

The system is a **bilingual web-based platform** that allows users to search, query, and receive legal information in natural language through a chatbot interface. The system also includes:

- A multilingual semantic search engine.
- A personalized legal document recommendation system.
- Automated legal document updates from GitHub.
- Integration with **Gemini API** for LLM responses.
- A **vector database (e.g., FAISS)** for semantic document retrieval.

  Key usage scenarios include:
- General citizens seeking simplified legal explanations.
- Legal professionals accessing detailed laws or gazettes.
- Students and researchers retrieving references or summaries.

1.3 Definitions, Acronyms, and Abbreviations

Table 1 - Definitions, Acronyms, and Abbreviations

Term	Description
RAG	Retrieval-Augmented Generation – A hybrid method where retrieved documents are used as context for generating responses with LLMs.
LLM	Large Language Model – An AI model trained on vast text corpora to understand and generate human-like text.
Gemini API	Google's API for accessing Gemini AI models.

Vector Database	A specialized database for storing and retrieving high-dimensional vectors (e.g., FAISS, Pinecone).
FAISS	Facebook AI Similarity Search – a vector database library for efficient similarity search.
LaBSE	Language-agnostic BERT Sentence Embedding – used for multilingual sentence embedding.
GitHub Actions	CI/CD automation tool used to fetch updated legal data.

1.4 References

- lk_legal_docs dataset – https://github.com/nuuuwan/lk_legal_docs
- FAISS documentation – https://python.langchain.com/docs/integrations/vectorstores/faiss
- Gemini API – https://ai.google.dev/
- LaBSE model – https://huggingface.co/sentence-transformers/LaBSE
- Legal-BERT model – https://huggingface.co/nlpaueb/legal-bert-base-uncased
- LankaLaw Platform – https://lankalaw.net/
- "Retrieval-Augmented Generation for Large Language Models: A Survey" – https://arxiv.org/abs/2312.10997

  (Accessed on 6 August 2025)

1.5 Overview

This document outlines both **functional** and **non-functional** requirements. It covers system use-cases, constraints, user characteristics, interface definitions, and quality expectations. It acts as a contract between the client and the development team.

2. Overall Description

2.1 Product Perspective

This is a **modular, web-based system** deployed on cloud infrastructure. It integrates external APIs (Gemini, GitHub), uses vector search for efficient retrieval, and supports scalability and extensibility.

2.2 Product Functions

- **Legal Chatbot:** Interprets user queries and delivers context-based legal answers.
- **Semantic Search:** Uses sentence embeddings to search legal content across Sinhala and English.
- **Recommendation Engine:** Recommends related laws based on past queries.
- **Auto-Updater:** Pulls and indexes legal content from GitHub repositories.

2.3 User Characteristics

Table 2 - User Characteristics

	User Type	Description
	General Public	May have minimal legal knowledge; it needs simple explanations.
	Legal Professionals	Require more technical accuracy and advanced search.
	Students/Researchers	Use system to find relevant acts, amendments, or gazettes.

- All users need an intuitive UI.
- Bilingual support is essential.
- Personalized features require authentication.

2.4 Constraints

- Supports Sinhala and English (UI + document content)
- Accessible via modern desktop and mobile browsers.
- Chatbot answers must be **legally accurate** and traceable to source content.
- Must use open-source or free tools unless otherwise approved.

2.5 Assumptions and Dependencies

- GitHub API is available and properly configured.

- Gemini API remains free/affordable and accessible.
- Data (laws, gazettes) from lk_legal_docs is accurate and public domain
- Adequate internet connectivity for API communications.

2.6 Requirements Subsets

Project will be developed in the following phases:

1. Chatbot module (RAG-based)
2. Semantic search interface
3. User authentication + profile + history
4. Personalized recommendation engine
5. Legal document auto-update integration

3. Specific Requirements

3.1 Functionality

3.1.1 Chatbot for Legal Queries

- Accepts natural language questions in Sinhala/English.
- Uses Gemini API to generate context-aware answers.
- Uses RAG technique to ground responses in relevant legal texts.

3.1.2 Recommendation System

- Tracks user queries and visited documents.
- Provides recommendations via cosine similarity of document embeddings.

3.1.3 Multilingual Legal Search Engine

- Accepts Sinhala or English search terms.
- Searches using both lexical (keyword) and semantic (vector) approaches.
- Returns ranked list of documents.

3.1.4 Automatic Updates

- Uses GitHub Actions to pull updates weekly.
- Automatically parses new content, embeds it, and stores in vector DB.
- Users are notified of significant updates (e.g., new gazette).

3.2 Usability

3.2.1 User Interface

- The system will have a simple, intuitive user interface.
- The main interface will consist of a search bar, chatbot interface, and recommended documents section.

3.3 Reliability

3.3.1 System Availability

- The system will be available 99% of the time, excluding planned maintenance.
- Scheduled downtimes will be notified to users in advance.

3.3.2 Error Handling

- If a search or query fails, the system will display a user-friendly error message.
- The chatbot will suggest rephrased questions if it cannot find an answer.

3.3.3 Accuracy

- The system aims to return legally accurate and relevant responses based on verified documents.

- The chatbot responses will be based on document sections fetched through semantic similarity, ensuring high contextual relevance.
- Accuracy of semantic search results will be tested and improved through continuous evaluation and feedback.

3.3.4 Consistency

- Repeated queries must yield same/similar results unless database changes.
- Embeddings managed in a single unified vector space.

3.4 Performance and Security

3.4.1 Response Time

- Search results should be returned within 3 seconds.
- Chatbot responses should be generated within 5 seconds.

3.4.2 Security

- User data will be stored securely with encryption.
- The system will require user authentication for personalized features.

3.5 Supportability

3.5.1 System Maintenance

- Regular system updates will be performed to improve functionality and security.
- Backup of legal documents will be maintained periodically.

3.6 Design Constraints

3.6.1 Tech stack

- Must use Python stack (preferred: FastAPI + React).
- Embeddings: Use LaBSE or LegalBERT.
- Vector DB: FAISS, Pinecone or equivalent.

3.6.2 Compatibility

- The system must be compatible with modern browsers and mobile devices.

3.7 Online User Documentation and Help System Requirements

- In-app help section (e.g., chatbot FAQs, usage tips).
- User onboarding for first-time users.
- GitHub-based technical documentation.

3.8 Interfaces

3.8.1 User Interfaces

Table 3 - User Interfaces

	Page	Elements
	Home	Search bar, chatbot box, language toggle
	Document Viewer	Document title, metadata, highlight query
	Dashboard	Past queries, recommendations
	Admin (optional)	Update log, error monitor

3.8.2 Hardware Interfaces

Browser-based. No specific hardware dependencies beyond:

- RAM: 4GB+ (for developer testing)
- Disk: 2GB for vector + text storage (cloud-hosted).

3.8.3 Software Interfaces

Table 4 - Software Interfaces

	API	Purpose
	Gemini	LLM responses
	GitHub	Fetch legal data
	VectorDB	Semantic search (FAISS)

3.8.4 Communications Interfaces

- HTTPS (REST APIs)
- GitHub API (polling or webhooks)
- Async HTTP for embedding & Gemini requests.

3.8.5 System Architecture

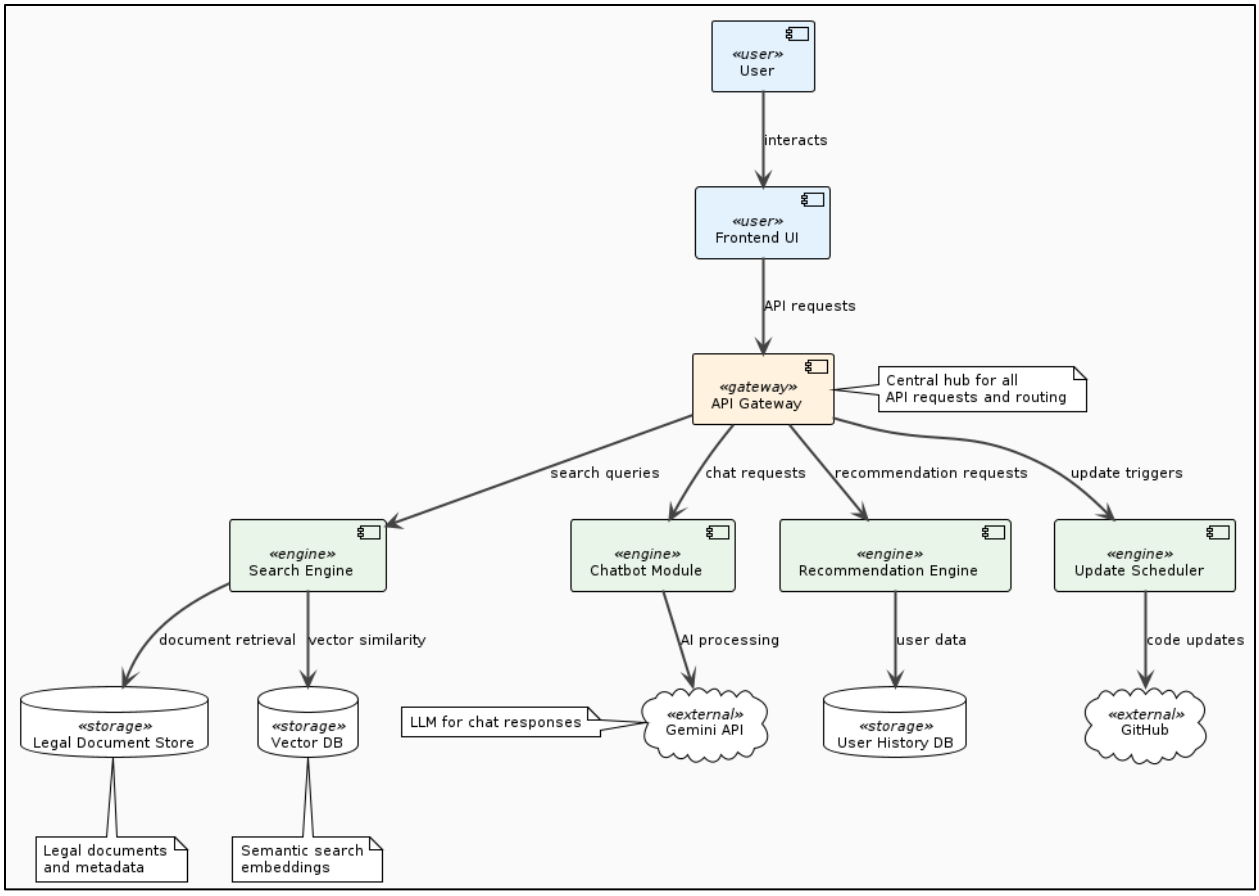

Figure 1- System Architecture

3.9 Database Requirements

Table 5 - Database Requirements

	Table	Purpose
	Users	Profile, credentials
	Queries	Text, timestamps, matched docs
	LegalDocuments	ID, content, title, metadata
	Embeddings	Document vectors
	Feedback	User ratings of responses

3.10 Licensing, Legal, Copyright, and Other Notices

- Only public-domain legal texts are used.

- AI responses do not constitute legal advice.
- All third-party tools under permissible licenses (e.g., Apache 2.0).

3.11 Applicable Standards

- Web Content Accessibility Guidelines (WCAG) 2.1 for accessibility
- GDPR/Sri Lankan Privacy Guidelines for user data.

4. Supporting Information

This section provides supplemental materials that enhance the usability and clarity of the Software Requirements Specification (SRS). These materials include a table of contents, index, appendices such as user-interface prototypes, and references to standards, tools, technologies, and theoretical concepts used in the system design.

4.1 Appendices

The following appendices contain additional supporting artifacts:

4.1.1 Appendix A: UI Prototypes

Wireframes and early mockups of the planned user interface created using Figma (Accessed on 2025-08-06).

Note: All appendices are to be considered **integral parts** of the system requirements unless stated otherwise.

4.2 References

Below is a curated list of references in IEEE style, supporting various aspects of the project:

4.2.1 Legal Corpus and Domain Knowledge

[1] Nuwan Waidyanatha, lk_legal_docs: Sri Lankan Legal Corpus, GitHub Repository, [online]. Available: https://github.com/nuuuwan/lk_legal_docs

[2] “Parliament of Sri Lanka,” [Online]. Available: https://www.parliament.lk/

[3] “Department of Government Printing – Sri Lanka,” [Online]. Available: http://www.documents.gov.lk/

4.2.2 Tools and Technologies

[4] “Gemini API,” Google Developers, [online]. Available: https://developers.google.com/gemini/

[5] “LangChain Documentation,” [Online]. Available: https://docs.langchain.com/

[6] “Chroma Vector DB,” [Online]. Available: https://www.trychroma.com/

4.2.3 Research Articles

[7] Y. Liu et al. (2023). "Retrieval-Augmented Generation for Large Language Models: A Survey." arXiv preprint arXiv:2312.10997. https://arxiv.org/abs/2312.10997

Appendix A: UI Prototypes

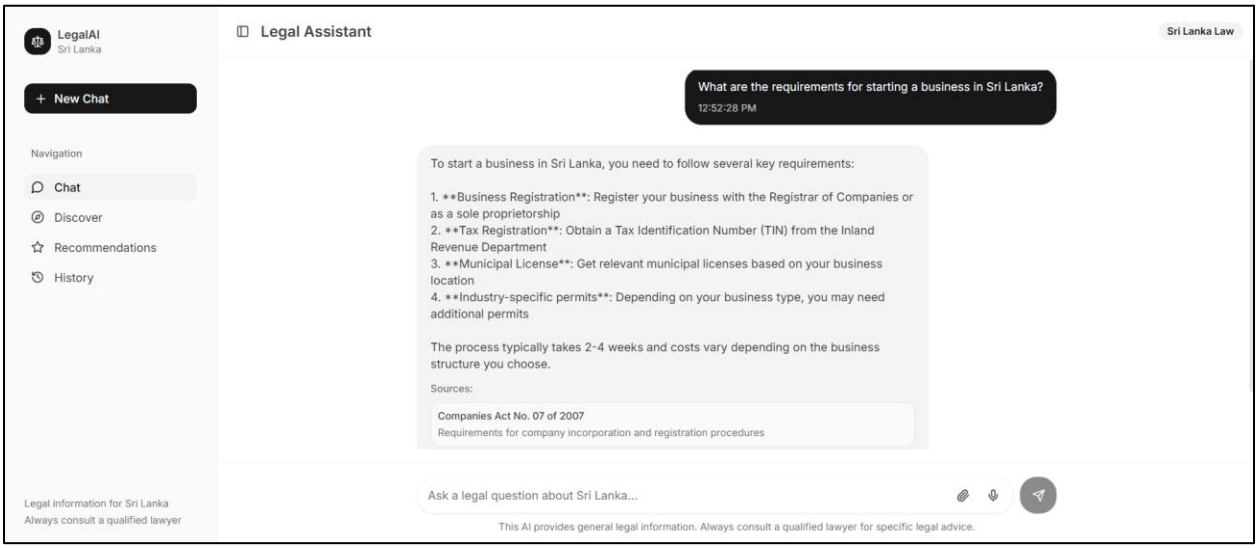

Figure 2 - Home page

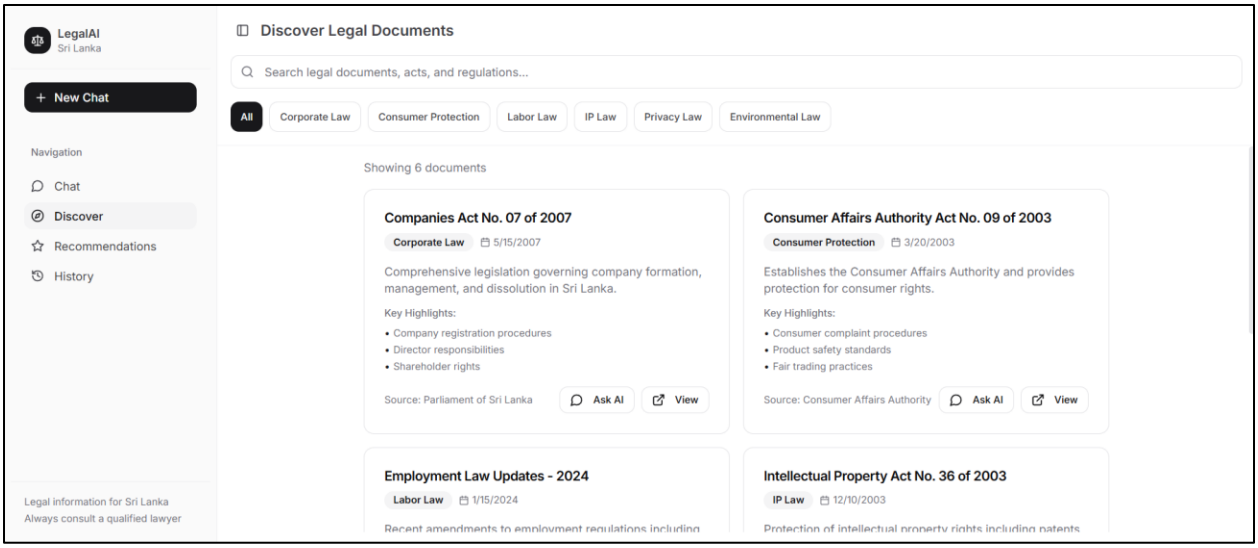

Figure 3 - Discover page

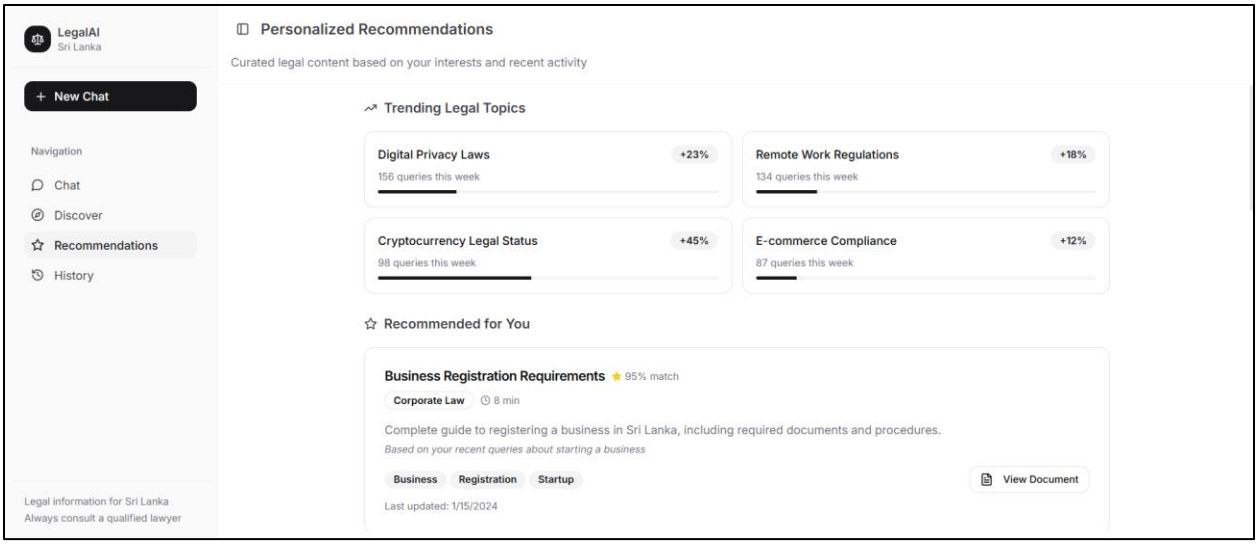

Figure 4 - Recommendation page

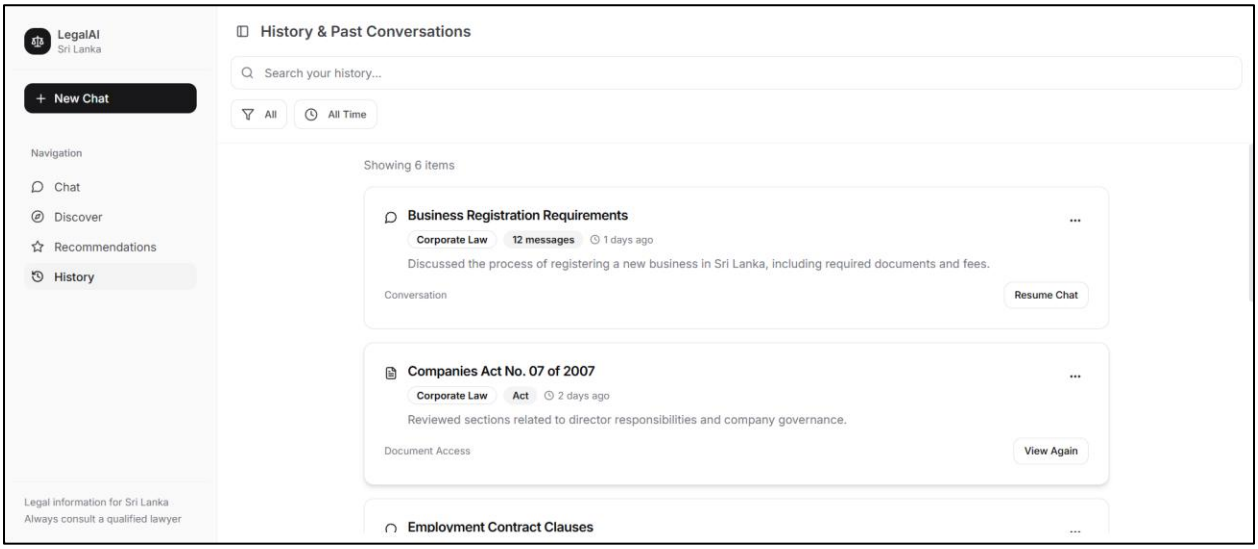

Figure 5 - History page