Software Requirements Specification (SRS) Document

Revisions

Version	Primary Author(s)	Description of Version	Date Completed

Review & Approval

Requirements Document Approval History

Approving Party	Version Approved	Signature	Date

Requirements Document Review History

Reviewer	Version Reviewed	Signature	Date

Contents

1. Introduction

1.1 Introduction

The purpose of this document is to define and describe the requirements of the Lung Cancer Risk Prediction System. This Software Requirements Specification outlines the system's intended functionality, constraints, performance expectations, and operational behaviour. It serves as a formal reference for developers, stakeholders, and evaluators to ensure that the system is designed, implemented, and tested according to agreed-upon standards.

1.2 Scope of this Document

The customers and end users of this system include medical practitioners, health researchers, clinical staff, and individuals seeking risk assessment related to lung cancer. The developers of the system are the designated project team responsible for building the machine-learning-based prediction..

Constraints related to this section include the availability of reliable health-related datasets, the time-bound project development deadlines, and the requirement to adhere to medical data privacy and ethical handling standards.

1.3 Overview

The Lung Cancer Risk Prediction System is a software application that uses patient health attributes, such as age, gender, smoking history, air pollution exposure, family history, and respiratory symptoms, to estimate an individual's likelihood of lung cancer risk.

The system will feature:

- A structured database to store patient attributes.
- A machine learning model for risk prediction.
- A user interface for entering data and viewing results.
- Import/export support for CSV health records.
- Automated validation of user inputs to minimize diagnostic errors.

This system aims to assist healthcare professionals in preliminary screening and to support individuals in understanding potential early-warning indicators.

1.4 Business Context

Lung cancer remains one of the leading causes of mortality worldwide, with early detection significantly improving survival rates. The Lung Cancer Risk Prediction System provides an analytical tool for healthcare institutions, wellness centers , and research organizations to support early identification of at-risk individuals.

The system is especially valuable for public health programs, preventive care initiatives, and resource-limited regions, where early risk assessment can guide further medical evaluation. This tool does not replace professional diagnosis but enhances decision-making by offering an accessible, data-driven prediction mechanism.

2. General Description

2.1 Product Functions

The product should make the input of patient health attributes and the overall risk-prediction process simple, efficient, and user-friendly for both medical personnel and general users. The system streamlines the process of collecting data such as age, smoking history, air pollution exposure, and respiratory symptoms, and then generates an estimated lung cancer risk using a machine learning model. The goal is to reduce manual analysis time and provide fast, data-driven insights.

2.2 Similar System Information

There are multiple existing health-risk calculators and online diagnostic tools used for screening various diseases. Many of these rely on statistical models or predefined scoring techniques. The strength of our system lies in its machine-learning-based prediction, which can adapt to new data, increase accuracy over time, and provide more personalized risk assessments. Additionally, features such as CSV data import/export make it more flexible than many static online calculators.

2.3 User Characteristics

The users include healthcare professionals, clinic staff, and individuals seeking a preliminary lung cancer risk assessment. For medical staff, a basic understanding of data entry and patient screening procedures is required. General users only need basic computer literacy to enter data through the user interface. Training or user guidance may be provided to ensure smooth interaction with the system, especially for personnel analysing large datasets.

2.4 User Problem Statement

Traditional lung cancer screening methods are often slow, resource-intensive, and not accessible to every individual. Many users face difficulties in manually assessing risk factors due to lack of experience or limited access to medical evaluation tools. The existing process may lead to delayed risk identification, which increases the likelihood of late-stage diagnoses. Users require a faster, data-driven solution that can efficiently analyse multiple health attributes and generate reliable preliminary predictions.

2.5 User Objectives

Users want a system that:

- Stores and organizes patient health information efficiently.
- Quickly computes lung cancer risk based on multiple attributes.
- Reduces the time required for manual evaluation.
- Provides accessible and understandable risk results.
- Supports early detection and preventive healthcare decisions.

The major objective is to enhance speed, accuracy, and usability in the lung cancer risk assessment process.

2.6 General Constraints

Constraints include the need for an intuitive and easy-to-use interface, a system that runs on standard operating environments such as Windows, macOS, or Linux, and compatibility with common data formats such as CSV. The system must be built using easily maintainable technologies (Python Flask) and must comply with data privacy guidelines, especially concerning medical information. Additionally, accuracy is dependent on the quality of input data and the availability of sufficient training datasets for model development.

3. Functional Requirements

1. The patient health attributes shall be stored in the system database.

1. Patient information such as age, gender, smoking history, symptoms, and environmental exposure shall be stored securely in the system's database.
2. Data shall be stored locally on the machine or on a secure server and must contain complete fields as defined in the system schema.
3. Very high criticality
4. Limited internet availability or restricted access to cloud storage may present a technical challenge.
5. This risk is mitigated by ensuring the system can operate offline and synchronize data when connectivity becomes available.
6. This requirement forms the foundation of the entire system; all other features rely on accurate and accessible stored data.

---

2. The stored health data shall be accessible through queries and reports.

1. Users should be able to generate reports summarizing patient inputs, predicted risk levels, and system performance metrics. They should also be able to run search queries on the stored dataset.
2. Very high criticality
3. No major technical issues are anticipated in implementing this requirement, as modern database systems readily support queries and reporting.
4. Given the capabilities of the chosen database technology (MongoDB), this requirement is fully achievable.
5. This requirement depends on Requirement 1, as querying relies on properly stored data.

---

3. The stored patient data should be modifiable through forms.

1. Patient attributes and other information should be addable, editable, and updatable through user-friendly input forms.
2. Very high criticality
3. No significant technical risks are expected during implementation.
4. The primary issue could be users entering incorrect or invalid data; this will be addressed through input validation and user training/support.
5. This requirement also depends on Requirement 1 since modifications require correctly stored and structured data.

---

4. The system shall generate lung cancer risk predictions using a machine learning model.

1. The system must process patient attributes and compute a risk score (e.g., low, moderate, high) using a trained ML model.
2. Very high criticality
3. Potential challenges include insufficient dataset quality or computational limitations, which may affect prediction accuracy.
4. These issues can be mitigated by optimizing the model, preprocessing data, and ensuring periodic retraining with updated datasets.
5. This requirement depends on the successful setup of data storage (Requirement 1) and input modification forms (Requirement 3).

---

5. The system shall display prediction results clearly to the user.

1. The system must present risk results in an easy-to-understand format, including textual scores, color-coded risk indicators, or graphical outputs.
2. High criticality
3. No major technical issues are expected for implementing output views.
4. User misunderstanding could occur if results are overly complex; this can be resolved using simple UI design and explanatory notes.
5. This requirement depends on Requirement 4, as predictions must be generated before being displayed.

---

6. The system shall validate user inputs before generating predictions.

1. Input fields such as age, smoking years, and symptoms must be checked for correctness, completeness, and valid ranges.
2. High criticality
3. Incorrect validation rules or missing constraints could pose risks.
4. These risks will be mitigated through robust validation logic and testing of edge cases.
5. This requirement supports Requirement 4 by ensuring that only valid data is used for predictions.

---

7. The system shall support data import and export through CSV files.

1. Users must be able to upload patient datasets and export prediction results in standard spreadsheet formats.
2. Medium criticality
3. File format issues or version incompatibilities may pose challenges.

4. These can be resolved by using widely supported formats and libraries.
5. This requirement depends on Requirement 1 and Requirement 2 for proper data handling.

4. Interface Requirements

4.1 User Interfaces

• 4.1.1 GUI

The user interface for this program is a graphical web-based interface designed using standard web technologies (HTML, CSS, JavaScript) and served through a framework such as Flask . The GUI includes input forms for entering patient attributes, buttons for submitting prediction requests, and panels for displaying risk results. The system may also include data visualization components such as charts or colored indicators to represent risk levels. Form-based input validation ensures that users provide complete and accurate information.

• 4.1.2 CLI

There is no dedicated command-line interface for general users. System administrators or developers may optionally run backend scripts through a CLI for testing, model training, or maintenance, but this is not intended for end-user interaction.

• 4.1.3 API

The current product does not include an external API. All operations- data entry, prediction, and reporting are handled internally within the application. An API may be considered in future versions for integration with hospital systems or research databases.

• 4.1.4 Diagnostics or ROM

The application includes basic troubleshooting mechanisms such as error messages for invalid inputs, server logs for system monitoring, and a help/documentation section guiding users on how to enter data and interpret prediction results. Additional debugging information is available to developers through system logs maintained by the backend framework.

4.2 Hardware Interfaces

The system runs on standard computer hardware or server infrastructure. It uses the device's storage system for saving datasets and uses RAM and CPU resources to load and execute the machine learning model. Hardware interactions such as storage access and network communication are managed by the operating system and the hosting environment.

4.3 Communications Interfaces

If the system is deployed on a network or cloud platform, communication occurs over standard web protocols (HTTP/HTTPS). The operating system and web server handle networking activities, including sending prediction requests, retrieving stored data, and managing multiple user sessions. In offline/local deployments, no external communication is required except for manual file imports and exports.

4.4 Software Interfaces

The system may import and export data using spreadsheet formats such as CSV. This functionality is supported by the backend framework through standard libraries (e.g., pandas for Python). The software also interacts with the machine learning model stored in serialized format (such as .pkl or .joblib). The web framework acts as the primary interface between the user interface, database, and prediction engine.

5. Performance Requirements

The Lung Cancer Risk Prediction System is designed to operate as a lightweight web-based or desktop application, requiring no specialized hardware. The performance requirements are minimal, as the system primarily handles form-based input, basic data storage, and execution of a pre-trained machine learning model, which is computationally inexpensive once deployed.
The system should run efficiently on any modern machine capable of supporting a standard web browser or Python-based environment. Only a negligible amount of storage space is required for storing patient data and model files.

Typical system requirements for running the application (web) are as follows:

- **1 GHz processor or higher** (dual-core recommended)
- **2GB RAM or higher**
- **500MB available hard drive space** for model files, logs, and stored data
- **Modern operating system** such as:
  - Windows 7 or later
  - macOS 10.12 or later
  - Linux distributions (Ubuntu, Fedora, etc.)
- Any modern web browser, including:
  - Google Chrome
  - Firefox
  - Microsoft Edge
  - Safari

If the system is deployed locally using Python:

- **Python 3.8 or higher**
- Required libraries such as Flask, pandas, scikit-learn, and joblib

No additional hardware is needed beyond what is already required for running standard productivity software. Because the application is lightweight, it can run smoothly on low- to mid-tier machines, ensuring accessibility for clinics, research labs, and individual users.

6. Other non-functional attributes

6.1 Security

The system shall be designed with a level of security appropriate for the sensitivity of medical and personal information stored in the database. Further consultation with the client may be required to determine how sensitive the collected patient data is and what level of protection is needed. Although no critical financial information such as credit card details is stored, the system may handle personal health information, which must be protected. Security options may include encrypting stored data, enforcing password-protected access, and restricting system usage to authorized personnel only.

6.2 Binary Compatibility

The system will be compatible with any modern computer capable of running a standard web browser or a Python-based application. It is designed with multi-device compatibility in mind and can operate on Windows, macOS, or Linux systems without requiring specialized hardware. The system only requires the machine to meet minimal software and browser requirements.

6.3 Reliability

Reliability is a critical attribute of this system. Regular backups of stored patient data and model files will ensure that the system can be restored with minimal data loss in the event of failure or unexpected issues. The software will undergo thorough testing by the development team to validate prediction accuracy, user input handling, and interface functionality to ensure dependable operation.

6.4 Maintainability

The system shall be maintained by designated administrators such as healthcare staff, IT personnel, or another responsible member of the organization. Maintenance tasks include updating the machine learning model, managing data backups, and ensuring that the system dependencies remain up to date. Clear documentation will support future maintenance activities.

6.5 Portability

The system shall be designed to run on multiple computers and can be deployed on different environments such as local machines, institutional servers, or cloud platforms. As long as the required runtime environment (Python/Browser) is present, the system can be transferred and executed with minimal configuration.

6.6 Extensibility

The system shall be designed and documented in a manner that allows developers or technical staff to extend it for future needs. This includes adding more medical attributes, updating the machine learning model, integrating APIs, or enhancing the user interface. Clear internal documentation and modular code structure will help ensure ease of extension.

6.7 Reusability

The system should be reusable for repeated screenings or large-scale data collection efforts. The database structure, risk prediction engine, and user interface can be repurposed for continuous patient assessments, new datasets, or expanded healthcare programs with minimal modification.

6.8 Application Affinity/Compatibility

The system requires a modern web environment or Python runtime rather than Microsoft Office tools. It is compatible with common web browsers and integrates with data-processing libraries such as pandas and scikit-learn. It may also interact with spreadsheet software (Excel/CSV) for importing and exporting data.

6.9 Resource Utilization

The resources used in the creation of this system include the project supervisor or domain expert, the development team, computing resources such as laptops or laboratory systems, and internet connectivity for downloading required libraries and datasets. The system itself uses minimal CPU and memory for operation, as prediction tasks are lightweight.

6.10 Serviceability

The system should be maintainable by any person with a basic understanding of web applications or Python system administration. Troubleshooting and updates should be straightforward due to clear documentation, simple structure, and easily accessible code modules.

7. Operational Scenarios

Scenario A: Initial Patient Data Entry

The user shall enter the patient's health attributes into the system for initial data collection and system usage. Fields such as age, gender, smoking history, symptoms, and exposure levels will be completed through an input form designed to validate and manipulate the data appropriately. This scenario supports the primary function of building a dataset for analysis and prediction.

Scenario B: Risk Prediction Process

The user shall input or select patient information and request a lung cancer risk evaluation. The system will process the provided attributes using the machine learning model and display the predicted risk (e.g., low, moderate, high). The user will also be able to review or save the prediction results. This scenario covers the main operational purpose of the system—delivering quick and informative risk assessments.

Scenario C: Data Review and Maintenance

The user may need to modify or delete patient information after it has been entered into the database. In such cases, the system shall allow authorized users to update, correct, or remove existing data records through the provided interface. This scenario ensures that the database remains accurate, current, and clean for both reporting and future predictions.

Scenario D: Bulk Data Import

The user shall import multiple patient records at once using a CSV or Excel file. The system will validate the structure of the file, check for missing or invalid fields, and insert the data into the database. This scenario supports clinics or researchers who process large amounts of patient information and require efficient data entry.

Scenario E: Generating Reports

The user shall be able to generate summaries or detailed reports based on stored patient data and prediction results. These reports may include counts of high-risk patients, trends in smoking history, or distribution of symptoms. The system shall allow exporting these reports for medical or research use. This scenario supports data monitoring, medical decision-making, and research analysis.

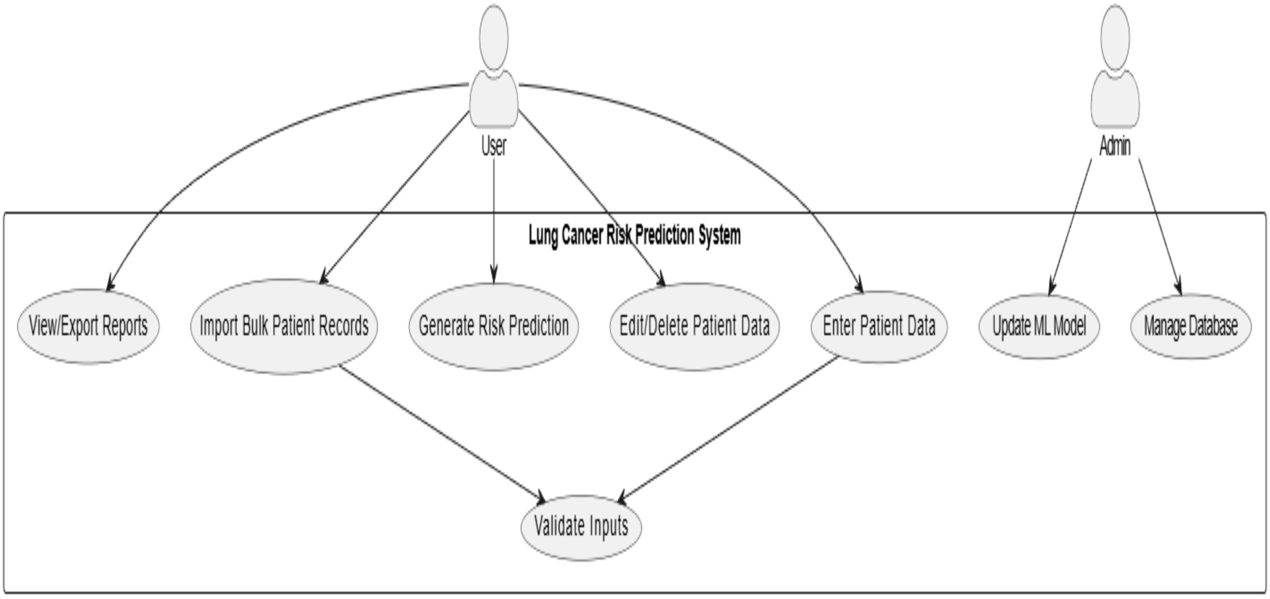
Scenario F: Model Update and Retraining

An administrator or authorized user shall update or retrain the machine learning model when new patient data becomes available or when improvements are required. The system will load updated datasets, retrain the model, and replace the previous prediction engine. This scenario ensures that the system remains accurate, up-to-date, and aligned with evolving medical data.

8. Preliminary Use Case Models and Sequence Diagrams

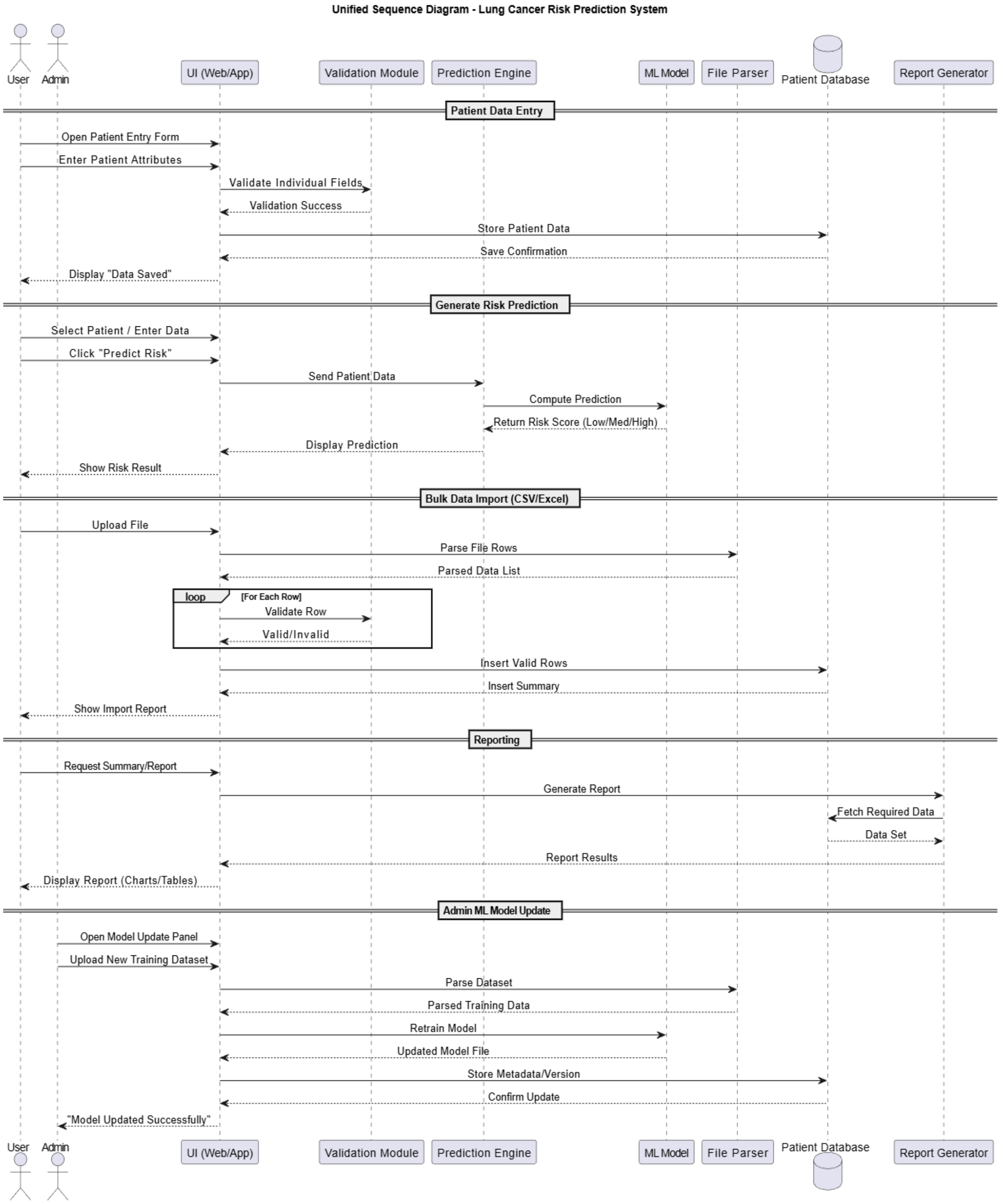
This section presents a list of the fundamental use cases and sequence diagrams that satisfy the system's core requirements. The purpose is to provide an alternative, "structural" view of how the system's functionality operates, how the users interact with the system, and how these interactions lead to the completion of system goals. These models help illustrate the logical flow and support a deeper understanding of the operations required for the Lung Cancer Risk Prediction System.

8.1 Use Case Model



8.1 Use Case Diagram

8.2 Sequence Diagrams



8.2 Sequence Diagrams

9. Updated Schedule

The updated PERT/GANTT chart is attached at the end of the document

10. Updated Budget

An updated budget is attached at the end of this document

11. Appendices

11.1 Definitions, Acronyms, Abbreviations

IDANRV- Intellectual Disabilities Agency of the New River Valley