

Software Requirements Specification for

Healthcare recommendation system

(Healtha)

Prepared by

Esraa Maged
Sara Samir
Iptihal Yousri

Rahma Khaled
Habiba Mohammed
Aya Elmallah

Version: (1)

Date: (10/20/2023)

Supervisors:
Dr. Ahmed Gamal
Prof. Tarek

Table of Contents

CHAPTER 1 SRS (Software Requirements Specification)

Problem Statement	i
Revision History	ii

1. Introduction

- 1.1. Purpose
- 1.2. Scope of the project
- 1.3. Definitions, Acronyms, and Abbreviations
- 1.4. Intended Audience and Reading Suggestions
- 1.5. Process Model
- 1.6. Stakeholders
- 1.7. Overview

2. Overall Description

- 2.1. Product Perspective
- 2.2. Product Functions
- 2.3. User Classes and Characteristics
- 2.4. Operating Environment
- 2.5. Design and Implementation Constraints
- 2.6. User Documentation
- 2.7. Assumptions and Dependencies
- 2.8. Functional and Non-Functional Requirements

3. System Features

- 3.1. Functional Requirements
- 3.2. Non-Functional Requirements

4. Use case Diagram

5. ERD Diagram

6. UML Class Diagram

1. Introduction

1.1.Purpose:

An SRS forms the basis of an organization's entire project. It sets out the framework that all the development teams will follow. It provides critical information to all the teams, including development, and operations.

Using the SRS helps an enterprise confirm that the requirements are fulfilled and helps business leaders make decisions about the lifecycle of their product,

In addition, writing an SRS can help developers reduce the time and effort necessary to meet their goals as well as save money on the cost of development.

1.2.Scope of the Project:

The laboratory analysis and recommendation system is designed to assist laboratory professionals and users in the accurate analysis of samples and provide appropriate recommendations based on the results. The system will streamline the laboratory processes and enhance productivity while ensuring high-quality analysis and reliable recommendations.

1. Target Users:

a. The primary beneficiaries of the system would include healthcare professionals such as doctors, physicians, and medical practitioners who require laboratory test results to make informed decisions regarding patient care.

b. Patient Benefits: The system aims to enhance patient care outcomes by facilitating timely and accurate laboratory test analysis and result interpretation, leading to improved diagnosis, treatment planning, and monitoring.

c. Researchers: Scientists and researchers who need accurate analysis and recommendations for their experiments.

d. Healthcare Professionals: Doctors, nurses, and clinicians who require lab reports and recommendations for diagnosis and treatment.

2. System Features and Functions:

The laboratory analysis and recommendation system will include the following key features:

a. Analysis Algorithms: Advanced algorithms and models will be utilized to analyze the submitted samples accurately, considering various parameters and variables.

b. Recommendation Generation: Based on the analysis results, the system will generate appropriate recommendations, suggesting further actions, treatments, or experiments.

c. Reporting: The system will generate comprehensive reports summarizing the analysis findings and recommendations, which can be shared digitally or printed.

By fulfilling the above scope, the laboratory analysis and recommendation system will provide an efficient and reliable platform for users to analyze samples accurately and receive appropriate recommendations, enhancing the overall productivity and effectiveness of laboratory operations.

1.3.Definition, Acronyms, and Abbreviations:

1. Laboratory Analysis: The systematic examination and testing of samples in a controlled environment to gather information and determine their properties, composition, and characteristics.
2. Recommendation System: A software system that provides suggestions or advice to users based on their preferences, historical data, and predefined rules or algorithms.
3. SRS: Software Requirements Specification. A document that lays out the functional and non-functional requirements of the software system being developed.
4. Sample: A representative portion or specimen used for laboratory analysis, typically collected from a larger population or substance.

1.4.Intended Audience and Reading Suggestions

The intended audience who will read this SRS and use this application are:

1. Developers
2. Testers
3. Designers
4. Project manager
5. Documentation writer
6. Patients
7. Lab doctors
8. Quality assurance engineers

1.5.Process Model

In software engineering, a software process model divides software development work into distinct phases to improve design, product management, and project management. It is also known as a software development life cycle.

The goal of a software process model is to provide guidance for controlling and coordinating the tasks to achieve the end product and objectives as effectively as possible. There are many kinds of process models for meeting different requirements.

For this project, we selected the Agile process model to work on.

“Agile process model” refers to a software development approach based on iterative development. Agile methods break tasks into smaller iterations, or parts do long-termly involve long-term planning. The project scope and requirements are laid down at the beginning of the development process. Plans regarding the number of items, ratios, duration, and the scope of each iteration are clearly defined in advance.

We had chosen it because in the agile:

- ✓ Each iteration is considered as a short time "frame" in the Agile process model, which typically lasts from one to four weeks, so we could easily get an app version.

- ✓ The division of the entire project into smaller parts helps to minimize the project risk and to reduce the overall project delivery time requirements.

- ✓ Each iteration involves a team working through a full software development life cycle including planning, requirements analysis, design, coding, and testing before a working product is demonstrated to the client. It leads us to reach a good system.

- ✓ That would help users of this Resume Builder App who can manipulate the resume sections and add and edit features.

1.6.Stakeholders

Internal stakeholders:

1. Project manager
2. Project team (Software engineers/ developers/ testers/ designers/ database administrator)
3. Product admin

External stakeholders:

1. patients
2. Lab doctors
3. Medical practitioners

1.7. Overview:

We suppose that, in the coming sections, we are going to talk about the following:

- ✓ Overall Description
- ✓ Product Features
- ✓ Operating Environment
- ✓ Design and Implementation Constraints
- ✓ System Features

2. Overall Description

2.1: product perspective:

This software Requirements Specification (SRS) defines the requirements for a new system to replace the existing laboratory analysis system. The new system will be a mobile application that will provide a number of benefits, including:

1. Providing all medical lab tests and diseases encyclopedia.
2. Obtain AI-generated medical reports, verified by a laboratory doctor.
3. Chat with a chatbot to learn more about your medical reports and get personalized recommendations.
4. Getting a recommendation of the medical specialty and the nearest doctors on the specialist they need.

2.2: product functions (user requirements)

software will have the following core functionalities:

➤ **Registration**

- Creating an account
- Signing in to an existing account

➤ **Apply as a laboratory doctor**

- Doctors can review and confirm the ai-generated report.

➤ **Apply as a patient**

- Upload lab analysis document.
- Save or download the report.
- Listen to the screen reader.
- Chat with Healtha chatbot.
- View medical lab tests encyclopedias.
- View disease encyclopedias.
- View a recommendation of the medical specialty and the nearest doctors.

2.3 User Classes and Characteristics

Admin

Admin has full access to the system, manages all activities, and is the highest privileged user.

Key functions

- Access and modify laboratory doctors' profiles.
- Manage access for Laboratory doctors and patients.

Patients

The patient is anyone who is going to deal with the app. could be a normal patient, people with special needs, or others. They have access only to their records and could edit only their own. He can navigate within any available laboratory to generate their own report.

Key functions

- Log in as a patient
- Upload lab analysis document.
- Save or download the report.
- Listen to the screen reader.
- Chat with Healtha chatbot.
- View medical lab tests encyclopedias.
- View disease encyclopedias.
- View a recommendation of the medical specialty and the nearest doctors.

Laboratory doctors

The medical staff gives the agreement for the lab analysis report and send them to the users.

Key functions

- Log in as a laboratory doctor.
- Handle generated report.
- Confirm and send lab reports.

2.4 Operating Environment

Software Requirement: -

- Windows 10
- IOS or Android device

Hardware Requirement: -

- PC

- Core i7 Processor
- 8GB Ram
- 20GB storage

- Mobile

- CPU: 1.2GHz dual-core processor
- RAM: 1GB
- Storage: 8GB
- Operating system: Android 6.0 or later

Server Requirements: -

- Dual-Core 2 GHz or higher.
- GB RAM.
- 1 GB free disk space.
- Windows 10, 8, 7, Vista (Ultimate or Business)
- Microsoft SQL Server
- IIS versions 7+

2.5. Design and Implementation Constraints

Implementation and language restrictions

The proposed tech stack consists of:

- A MySQL database for storing data.
- A Dart backend for implementing the application's logic.
- A Flutter frontend for creating a native mobile experience on iOS and Android.
- A Node.js server for handling communication between the frontend and

backend.

- Python for developing and deploying AI models.

APIs

- **Login API**, This API is used to authenticate users and generate access tokens.

- **Register API**, This API is used to create new user accounts.

- **Data API**, This API is used to access and manipulate data stored on the server.

- **Chatbot API**, Use real-time accurate data to interact with the chatbot.

- **Testing APIs**, APIs will be tested using a combination of unit testing, integration testing, and system testing. Unit tests will be used to test individual API methods. Integration tests will be used to test the interaction between different APIs. System tests will be used to test the entire system as a whole.

- **API Documentation**, API documentation will be generated using a tool such as Swagger or OpenAPI. The documentation will include detailed information about each API, including its request and response parameters, error codes, and examples.

- **API Security**, API security will be a top priority. All APIs will be designed using security best practices, such as input validation, output encoding, and authentication. APIs will also be regularly monitored for security vulnerabilities.

Resource limits

- The users' device should have a working data plan or wifi connection to use some features.

- Reliability, security, dependability and efficiency of the database Needs to be compatible on most devices.

User Documentation:

The software will:

- let the user aware with a whole two sections one that provide a detailed description (comprehensive encyclopedia) for all diseases (Medical conditions) and the other provide a detailed description (comprehensive encyclopedia) for all Medical Lab Tests.

- let the user to have a good report from only his/her lab test results by the use of the power of AI, not just waiting for a doctor to read the test results and write the report for his/her.
- recommend the user with a medical specialty based on the user SYMPTOMS.
- Let the user able to contact an AI-powered chatbot to ask specific Medical questions.
- And more things for easier accessibility.

2.6 Assumptions and Dependencies:

Assumptions:

- The user will not log in to Healtha software without having registered before and has already set an email and password.
- The user will use his/her phone to be able to use the app.
- Display the Result or even upload media to the app, the user needs to connect to Wi-Fi.

Dependencies:

- Wi-Fi access
- Accuracy of the data
- Accuracy of different APIs

3. System Features

3.1. Functional Requirements

- 1- [H001] Users (doctors, laboratory doctors or patients) shall register to the system.
- 2- [H002] Admins shall manage all users' accounts (add/ edit/ delete).
- 3- [H003] Admins can manage all generated reports (Lab test results interpretation).
- 4- [H004] Patients can access all medical lab tests encyclopedias.
- 5- [H005] Patients can access all Diseases encyclopedias.
- 6- [H006] Patients can upload lab analysis documents.
- 7- [H007] Patients can view the generated report.
- 8- [H008] Patients can use the chatbot to get a further understanding of the reports.
- 9- [H009] Patients can view a recommendation of the medical specialty and the nearest doctors on the specialist they need.
- 10- [H010] Laboratory doctors can handle, review, and confirm generated reports.

- 11- [H011] Doctors can spot their location on the maps.

3.2. Non-Functional Requirements

1. [H101] The system shall operate using Android, IOS, or window10 or higher devices.
2. [H102] The software shall distinguish between authorized and nonauthorized users.
3. [H103] The software shall provide protection against threats and attacks.
4. [H104] The system shall be easy to use by anyone using two main languages Arabic and English
5. [H105] The product should be designed in a suitable way for patients.
6. [H106] The product should be easy to use.
7. [H107] The software should load in less than 6 seconds at all times.
8. [H108] The software should be efficient, reliable, and attractive to all people.
9. [H109] The system should be available 24/7.
10. [H110] The system shall, on average, operate without failure for 30 days.
11. [H111] The app could contain 2 main view modes (light and dark).
12. [H112] The app should conserve the battery life.
13. [H113] The system must be able to distinguish between authorized and unauthorized access.
14. [H114] The design of the app should not use more than three colors.
15. [H115] The language used in the interface should be formal and polite.
16. [H116] The system shall be used in variable lighting conditions.
17. [H117] The system shall be able to be modified to cope with a new class of use.
18. [H118] The system shall Considerate people with special needs through:
 - Lab reader support.
 - Text size and font customization
 - High contrast mode.
 - Gesture support.

4. Use Case

Narrative use case:

Identifier and name: [H201] Medical tour and generate report

Initiator: Patient

Goal: Obtaining lab analysis medical report.

Preconditions:

- The patient and the laboratory doctor must have an internet connection.

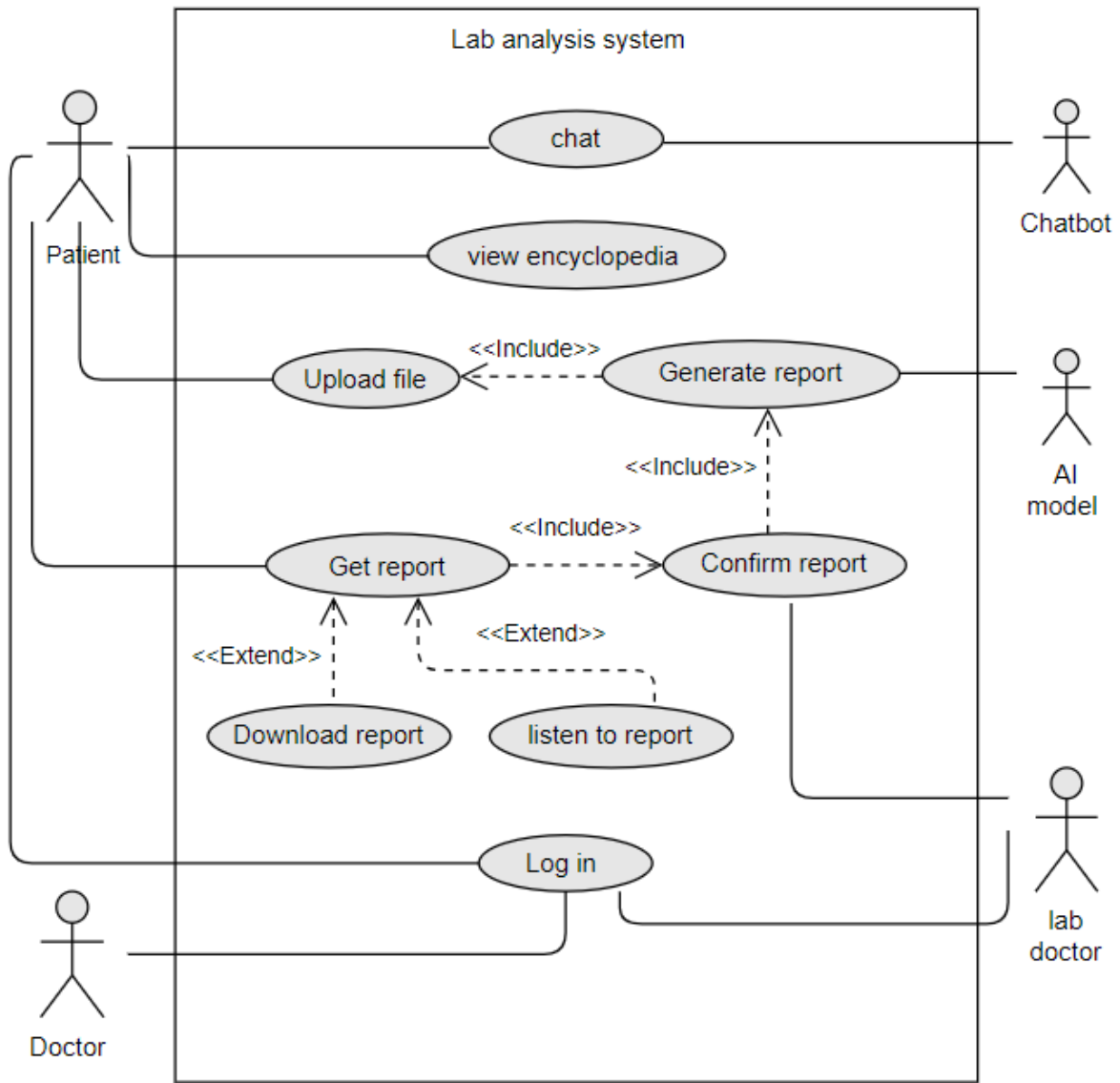
Postconditions:

- The patient has received the report of their lab analysis results.

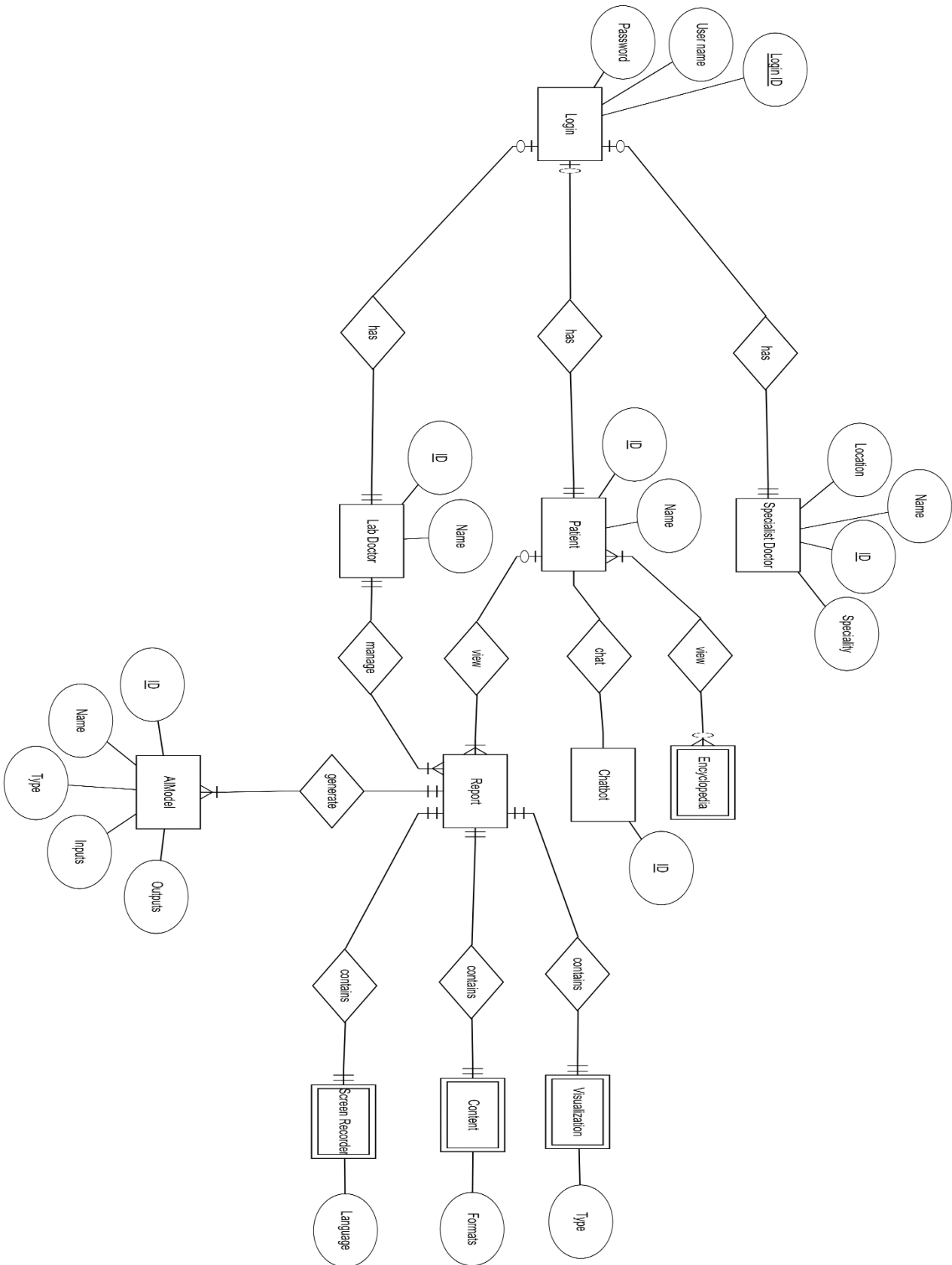
Main successful scenario:

- 1.1 The patient selects to view the medical lab tests encyclopedia.
- 1.2 The patient selects to view the diseases encyclopedia.
- 1.3 After log In, The patient selects to upload their lab analysis document.
 - 1.3.1 The AI model generates the report and recommended specialty.
 - 1.3.2 The laboratory doctor confirms the generated report.
 - 1.3.3 The system displays the report and the recommended specialty to the patient.
 - 1.3.4 The system displays to the user the nearest 10 doctors in the recommended specialty.
 - 1.3.5 The patient selects to save or download their report.
- 1.4 The patient selects to chat with the chatbot to learn more about their medical reports and get personalized recommendations.

Use Case Diagram:



5. ERD Diagram



6. UML Class Diagram

