Department of Computer Science & Engineering

University of Dhaka

Software Requirements Specification

for the MedWise Application

Iteration 2 Submission

Document Information	
Project Name	MedWise
Document Title	Software Requirements Specification
Version	2.0
Date	July 12, 2025
Development Methodology	Rational Unified Process (RUP)

Submitted To:

Sarker Tanveer Ahmed
Associate Professor
Department of Computer Science and Engineering

Team Name:

MedWise Application

Prepared by:

Shahriar Hossen Fahim (12)
Farzana Tasnim (14)
Md. Tauseef - Ur - Rahman (24)
Amina Islam (36)
Tamzid Bin Tariq (48)

Submission Date: July 12, 2025

Contents

1 Revision History

Date	Version	Description	Author
06/July/25	1.0	• Digital Report Management: Users can log in, upload medical records via camera, AI extracts data using OCR, organizes by date/doctor/disease, displays on dashboard. • Health Chatbot Assistant: Users can ask health-related questions, AI processes queries using NLP with personalized responses based on medical data. • Export Medical History: Users can export medical data as a structured PDF for specified timeframes with download/share functionality.	1. Shahriar Hossen Fahim, 2. Farzana Tasnim, 3. Md. Tauseef - Ur - Rahman, 4. Amina Islam, 5. Tamzid Bin Tariq
12/July/25	2.0	• Find Hospitals/Treatments: Users search for hospitals or treatments by specialty or location, app shows verified listings with contact info and maps for easy access. • Extract Medicine Info from Prescription and Set Reminder on Medicine: Users upload prescription images, AI extracts medicine schedule using OCR/NLP, app sets automatic dose reminders to support adherence. • Generate Graph for Glucose Level and Blood Pressure: Users select timeframe (7/30/90 days, all reports), app generates interactive graphs from stored data to visualize trends for self-monitoring.	1. Shahriar Hossen Fahim, 2. Farzana Tasnim, 3. Md. Tauseef - Ur - Rahman, 4. Amina Islam, 5. Tamzid Bin Tariq

Table 1: Version History of MedWise Application

2 Introduction

MedWise allows users to upload scanned medical reports, from which the system extracts and organizes key clinical data to automatically maintain a structured digital medical history. Key features of the platform include secure report storage with timeline visualization, a Natural Language Processing (NLP)-powered chatbot that suggests probable diseases and provides lifestyle advice, exportable health summaries, and a searchable database of diseases and treatment options. Users also have access to a directory of top hospitals and can browse symptom and disease information with interactive chatbot support.

In addition to these core features, MedWise offers intelligent medication reminders — notifying users when to take their medicines and tracking how many doses were missed. It also provides visual insights into health trends by generating graphs for vital metrics such as blood pressure and glucose levels, helping users monitor chronic conditions like hypertension and diabetes.

By combining smart record-keeping, automated reminders, visual health monitoring, and accessible medical knowledge, MedWise empowers users to take proactive control of their health. This document defines the functional and non-functional requirements of the MedWise system and serves as a foundation for its design, development, and validation.

2.1 Purpose

2.1.1 External behavior of Application

MedWise is an intelligent Android-based application designed for managing personal medical records and providing health assistance. The application enables users to upload scanned medical documents such as prescriptions and test reports. Leveraging computer vision (OCR), MedWise automatically extracts essential data from these documents and updates the user's medical history accordingly. This facilitates the organized and efficient tracking of clinical records over time.

Key functionalities of MedWise include:

- Uploading and storing prescriptions and medical reports in digital format.
- Automatic extraction and organization of medical information.
- Generation of summarized health reports in PDF format for easy sharing.
- Access to a health assistant chatbot that provides personalized health insights and recommendations.
- Search functionality for locating nearby hospitals based on disease category (e.g., cardiology, orthopedics, etc.).
- User profile management and the ability to update personal health details.
- Notification and reminder system for scheduled medication intake and missed doses.

MedWise aims to empower users by enabling better health data management and access to intelligent health support tools. It is important to note that this application is not a substitute for professional medical advice, diagnosis, or treatment.

2.1.2 Nonfunctional Requirements

- **Portability**
  - The application should run on any Android device with version 8.0 (Oreo) or higher.
  - Users should be able to reinstall the app on different devices and recover their data securely through cloud storage.
- **Security**
  - All user data, including medical reports and personal information, must be securely stored using encryption techniques.

  - Data transmission must use secure protocols (e.g., HTTPS).
  - Authentication (e.g., login credentials) must be enforced to prevent unauthorized access.
- **Maintainability**
  - The codebase should follow modular design principles to facilitate easy maintenance and future updates.
  - New features (e.g., chatbot enhancements, disease category expansion) should be integrable with minimal changes.
- **Reliability**
  - The app should operate correctly under normal and abnormal conditions without crashing.
  - Critical features like computer vision (OCR), reminders, and chatbot responses should be fault-tolerant and handle errors gracefully.
- **Scalability**
  - The backend should support increasing numbers of users and growing volumes of health data without performance loss.
  - The system architecture should allow for horizontal scaling when needed.
- **Performance**
  - Computer vision operations on reports up to 2MB should complete within 5 seconds.
  - Chatbot responses and search queries should return results within 2 seconds on average.
- **Reusability**
  - Components like the computer vision engine, chatbot, and notification system should be designed as reusable modules.
  - These components should be adaptable for integration into other health-related applications in the future.
- **Flexibility**
  - The system should support easy configuration of features like reminder times, language settings, and report categories.
  - It should allow integration with third-party APIs such as hospital databases or health monitoring tools.

2.1.3 Design Constrains

- **Screen ratio and resolution** – The MedWise app must be responsive and fully functional across common Android screen sizes and resolutions (phones and tablets), ensuring that all UI elements are visible and accessible without scrolling or distortion.

- **Device compatibility** – The application is designed primarily for Android mobile phones but should also support Android tablets with no loss of functionality or usability.
- **Client-server architecture** – MedWise will follow a client-server architecture, where the mobile app (client) interacts with a secure cloud-based backend for data storage, chatbot services, Computer vision processing, and health recommendations.
- **Mobile-based** – MedWise will be developed as a native Android application using the Android SDK and will be distributed via the Google Play Store. A web version is not part of the initial design.
- **Data privacy and security** – The application must ensure the confidentiality and integrity of sensitive medical data. It must use encrypted storage and communication channels, and user authentication must be securely implemented to protect health records.
- **Language and Localization** - The initial version of the application must support English, with flexibility for adding additional language support in the future.

2.2 Scope

MedWise is an Android application designed to enhance personal health management by enabling users to store, manage, organize, and access their health records digitally. Key features of the app include:

- **Secure storage** of health records, presented through a timeline-based view of the user's medical history.
- **Computer Vision(OCR)** support to extract relevant data from uploaded scanned medical reports.
- An **NLP-powered chatbot** offering health recommendations and insights related to diseases and symptoms.
- **Search functionality** for specific diseases and symptoms.
- **Exportable health summaries** that users can share with their doctors.
- A **listing of hospitals** and recommended treatment centers.
- **Reminders** for taking medications and attending appointments.

MedWise empowers users to access and organize their personal health data, combining intelligent tools to provide informed health support.
Note: The application is not intended to provide clinical diagnoses or replace professional medical advice.

2.3 Definitions, Acronyms, and Abbreviations

- **MedWise**: The name of the mobile application for personal healthcare management.
- **OCR**: Optical Character Recognition, used for extracting text from scanned medical documents.
- **Computer Vision**: A field of Artificial Intelligence that enables the application to interpret visual information from images or documents, such as identifying document layouts or assisting OCR.

- **NLP**: Natural Language Processing, used in the Health Chatbot Assistant for understanding and responding to user queries.
- **SRS**: Software Requirements Specification, this document.
- **AI**: Artificial Intelligence, used for data extraction, organization, and personalized recommendations.
- **PDF**: Portable Document Format, used for exporting medical history.
- **RUP**: Rational Unified Process, the iterative software development methodology used for MedWise.

2.4 References

[This subsection provides a complete list of all documents referenced elsewhere in the Software Requirements Specification. Identify each document by title, report number (if applicable), date, and publishing organization. Specify the sources from which the references can be obtained. This information may be provided by reference to an appendix or to another document.]

This subsection provides a list of documents and sources referenced in the preparation of the MedWise Software Requirements Specification.

- Python Software Foundation, *Python 3.11 Documentation*, 2023. Available at: `https://docs.python.org/3.11/`
- MongoDB Inc., *MongoDB Documentation*, 2023. Available at: `https://www.mongodb.com/docs/`
- ReactJS Contributors, *React – A JavaScript Library for Building User Interfaces*, Meta, 2023. Available at: `https://reactjs.org/`
- GitHub, *Github* , 2025. Available at: `https://github.com/`

2.5 Overview

This SRS is organized as follows: Section 1 provides an introduction, including purpose, scope, definitions, references, and overview. Section 2 offers an overall description of the product, including its perspective, functions, user characteristics, constraints, assumptions and dependencies, and requirements subsets. Section 3 details the specific requirements, including the use-case model survey, use-case reports, and supplementary specifications for the Digital Report Management, Health Chatbot Assistant, and Export Medical History subsystems.

3 Overall Description

3.1 Product Perspective

MedWise is a novel mobile application that integrates AI-driven features to streamline personal healthcare management. It operates as a standalone system with potential integrations with local pharmacies and wearable health devices. The application addresses the lack of a unified platform in Bangladesh for managing medical records, providing personalized health advice, and facilitating pharmacy interactions.

3.2 Product Functions

The MedWise application provides the following key functions for the specified subsystems:

- **Digital Report Management**: Scan, extract, and organize medical reports using computer vision (OCR) and AI, sortable by date, doctor, or disease.
- **Health Chatbot Assistant**: Enable users to ask health-related questions and receive instant, personalized responses based on their medical data using NLP.
- **Export Medical History**: Generate and export a PDF of a user's medical history for a specified timeframe with a single tap.
- **Find Hospitals/Treatments**: Search for hospitals or treatments by specialty or location with verified listings including contact information and maps.
- **Extract Medicine Info from Prescription and Set Reminder on Medicine**: Upload prescription images to automatically extract schedules using OCR/NLP and set dose reminders to improve adherence.
- **Generate Graph for Glucose Level and Blood Pressure**: Visualize trends with interactive graphs over selected timeframes (7, 30, 90 days, or all reports) for self-monitoring.

3.3 User Characteristics

The primary users are urban adults aged 25-50 in Dhaka, Bangladesh, particularly:

- Chronic patients (e.g., diabetes, hypertension) needing regular medical tracking.
- Elderly individuals requiring medication reminders.
- Families managing multiple prescriptions.

Users are assumed to have basic smartphone proficiency and access to medical documents for scanning.

3.4 Constraints

- The application must comply with data privacy regulations in Bangladesh.
- Computer vision(OCR) accuracy depends on the quality of scanned documents.
- The Health Chatbot Assistant's responses are limited to the user's stored medical data and general medical knowledge.
- Internet connectivity is required for pharmacy integration and data synchronization.

3.5 Assumptions and Dependencies

This section describes any key technical feasibility, subsystem or component availability, or other project related assumptions on which the viability of the software described by this Software Requirements Specification may be based.

- Users have access to smartphones with cameras for document scanning.
- Local pharmacies support API integration for order processing.

- The application depends on third-party OCR and NLP libraries for functionality.
- RUP phases (Inception, Elaboration, Construction, Transition) are adhered to for iterative development.

4 Specific Requirements

This section of the Software Requirements Specification contains all software requirements to a level of detail sufficient to enable designers to design a system to satisfy those requirements and testers to test that the system satisfies those requirements. When using use-case modeling, these requirements are captured in the use cases and the applicable supplementary specifications. If use-case modeling is not used, the outline for supplementary specifications may be inserted directly into this section.

4.1 Use-Case Model Survey

Actors:

- **User**: An individual using the MedWise app to manage healthcare data.
- **AI System**: Internal system component handling computer vision, NLP, and data processing.

Use Cases

1. **Digital Report Management**
2. **Health Chatbot Assistant**
3. **Export Medical History**
4. **Find Hospitals/Treatments**
5. **Extract Medicine Info from Prescription and Set Reminder on Medicine**
6. **Generate Graph for Glucose Level and Blood Pressure**

4.2 Use-Case Diagram

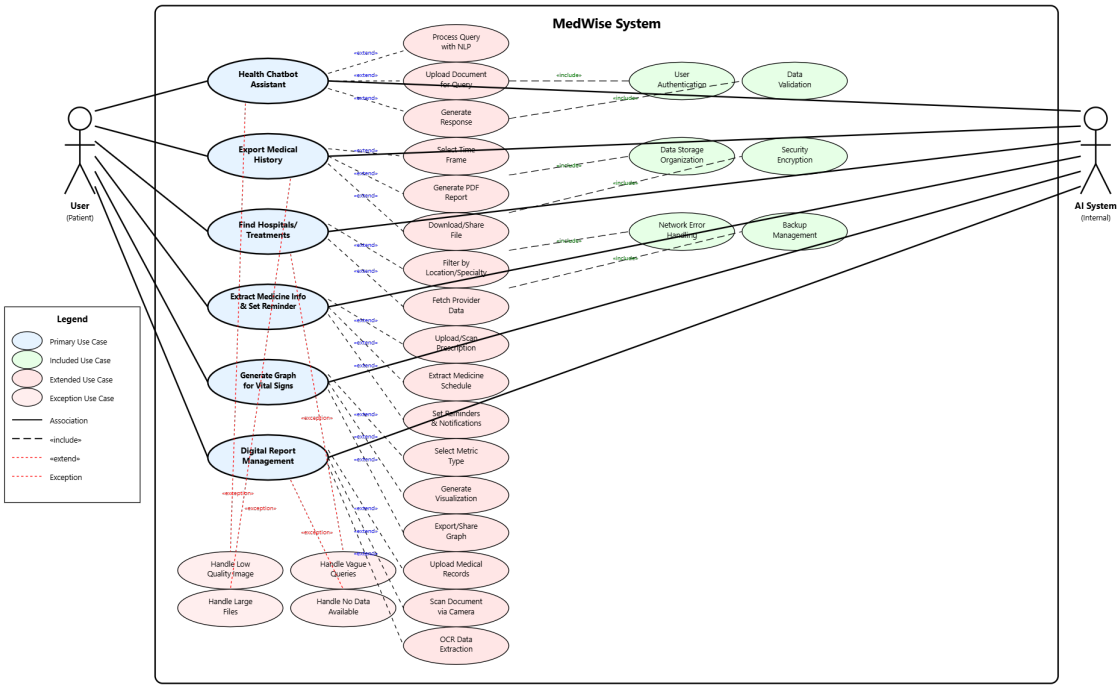

Figure 1: Use-Case Diagram

4.3 Use-Case Reports

1. **Use Case 1: Digital Report Management**
    - **Scope:** MedWise Application.
    - **Level:** User goal
    - **Primary Actor:** User (Patient)
    - **Stakeholders and Interests:**
        (a) **User:** Wants to store and track medical history easily.
        (b) **MedWise:** Ensures accurate data extraction and secure storage.
    - **Preconditions**:
        (a) User has a smartphone with a camera and the MedWise app installed.
        (b) User is authenticated.
    - **Postconditions**: Medical data is stored and organized by date, doctor, or disease.
    - **Main Success Scenario(Basic Flow)**:
        (a) User logs into MedWise.
        (b) Navigates to “Upload Medical Records.”
        (c) User scans a medical document (e.g., blood test report).

(d) AI System uses computer vision (OCR) to extract data (e.g., hemoglobin levels, date, doctor name).

(e) AI System organizes data and stores it in the user's profile.

(f) User views organized data in the app dashboard.

- **Extensions(Alternative Flows)**

(a) System displays network error:

1. System queues report for retry.

(b) c1. User can upload medical records as files.

1. Medical records are too large:
   - Suggest user to compress the file.
   - Recommend uploading in parts.

c2. Low-quality image:

1. Prompt user to re-upload.

c3. AI does not extract the required fields from medical records:

1. Prompt user to re-upload.

2. **Use Case 2: Health Chatbot Assistant**

- **Scope:** MedWise Application.
- **Level:** User goal
- **Primary Actor:** User (Patient)
- **Stakeholders and Interests:**

(a) **User:** Wants relevant, trustworthy health suggestions.

(b) **MedWise:** Provides smart and useful responses via NLP.

- **Preconditions**:

(a) User is logged in and connected to internet.

- **Postconditions**: User receives a relevant response to their query.
- **Main Success Scenario(Basic Flow)**:

(a) User opens the chatbot interface in the MedWise app.

(b) User inputs a health-related question (e.g., "What should I eat for diabetes?").

(c) AI System processes the query using NLP and the user's medical data.

(d) Chatbot provides a personalized response (e.g., dietary recommendations).

(e) User ends session.

- **Extensions(Alternative Flows)**

(a) * System displays network error:

1. System queues report for retry.

(b) * NLP unavailable:

1. Display error and suggest retry.

(c) b1. User can upload files or take photo of documents via camera and query about the documentation.

(d) c1. Vague queries:

1. Bot requests clarification.

3. **Use Case 3: Export Medical History**
   - **Scope:** MedWise Application.
   - **Level:** User goal
   - **Primary Actor:** User (Patient)
   - **Stakeholders and Interests:**
     (a) **User:** Needs summary for self-tracking or doctors.
     (b) **MedWise:**Ensures clear and formatted data sharing.
   - **Preconditions**: User has stored medical data in the app.
   - **Postconditions**: A PDF file is generated and available for download or sharing.
   - **Main Success Scenario(Basic Flow)**:
     (a) User selects the export option in the MedWise app.
     (b) User specifies the timeframe for the medical history.
     (c) AI System compiles relevant data into a structured PDF format.
     (d) User downloads or shares the PDF file.
   - **Extensions(Alternative Flows)**
     (a) System displays network error:
         1. System queues report for retry.
     (b) No data available to export summary :
         1. System notifies user.
4. **Use Case 4: Find Hospitals/Treatments**
   - **Scope:** MedWise Application.
   - **Level:** User goal
   - **Primary Actor:** User (Patient)
   - **Stakeholders and Interests:**

(a) **User:** Wants quick access to trusted healthcare providers.

(b) **MedWise:**Offers curated data from verified sources.

- **Preconditions**: Location enabled or search input provided.
- **Postconditions**: Listings shown with key info and maps.
- **Main Success Scenario(Basic Flow)**:
  (a) User opens "Hospitals".
  (b) Filters by specialty or location.
  (c) System fetches matching providers.
  (d) List shown with contact info, specialties, and maps.
- **Extensions(Alternative Flows)**
  (a) b1. Location off:
    1. User must search manually.
  (b) c1. No results:
    1. Suggestions shown.

5. **Use Case 5: Extract Medicine Info from Prescription and Set Reminder on Medicine**
   - **Scope:** MedWise Application.
   - **Level:** User goal
   - **Primary Actor:** User (Patient)
   - **Stakeholders and Interests:**
     (a) **User:** Wants to avoid missing doses by getting timely reminders.
     (b) **MedWise:** Ensures accurate extraction and effective medication adherence support.
   - **Preconditions**: User has a scanned or photographed prescription to upload.
   - **Postconditions**: Medicine schedule is extracted, reminders are set in the app.
   - **Main Success Scenario(Basic Flow)**:
     (a) User selects the "Add Prescription" option in the MedWise app.
     (b) User uploads or scans the prescription image.
     (c) AI system uses OCR and NLP to extract medicine names, dosages, and timing instructions.
     (d) System saves the medicine schedule and automatically sets reminders for doses.
     (e) User receives notifications as per the schedule.
   - **Extensions(Alternative Flows)**
     (a) b1. Low-Quality Image or Unreadable Prescription:

1. System notifies the user about poor quality or extraction failure.
2. User is prompted to upload a clearer image.

(b) c1. System displays network error:

1. System queues report for retry.

c2. NLP unavailable:

1. Display error and suggest retry.

6. **Use Case 6: Generate Graph for Glucose Level and Blood Pressure**

- **Scope:** MedWise Application.
- **Level:** User goal
- **Primary Actor:** User (Patient)
- **Stakeholders and Interests:**
  (a) **User:** Wants clear visual trends for self-monitoring or to show healthcare providers.
  (b) **MedWise:**Provides user-friendly, insightful visualizations for better health management.
- **Preconditions**:User has manually entered or uploaded glucose/BP data previously, or is entering new data now.
- **Postconditions**: App displays an interactive graph based on user-selected timeframe.
- **Main Success Scenario(Basic Flow)**:
  (a) User navigates to the "Health Graphs" section in the MedWise app.
  (b) User selects the metric to visualize (Glucose Level or Blood Pressure).
  (c) User chooses the desired timeframe (7 days, 30 days, 90 days, or All Reports).
  (d) System generates a line or bar graph with:
    i. X-axis: dates of reports within the selected timeframe.
    ii. Y-axis: measured values (glucose level, systolic/diastolic BP).
  (e) Graph is displayed with interactive features (zoom, tooltip with exact values).
  (f) User can save, export, or share the graph as an image or PDF if desired.
- **Extensions(Alternative Flows)**
  (a) c1. No Data Available for Selected Timeframe:
    1. System displays a message: "No data available for this period."
    2. User is offered the option to input new data manually.
  (b) d1. Network Error During Graph Generation:

1. System displays a network error message.
2. Option to retry loading the graph once connection is restored.

4.4 Supplementary Requirements

- **Performance Requirements**:
  - Computer vision processing for Digital Report Management should complete within 60 seconds for a standard medical document.
  - Health Chatbot Assistant responses should be generated within 5 seconds.
  - PDF export should complete within 5 seconds for up to 10 documents.
- **Security Requirements**:
  - User data must be encrypted during storage and transmission.
  - Access to medical data requires user authentication.
- **Usability Requirements**:
- The app must achieve 99.9% uptime for online features.
- Computer vision(OCR) accuracy must exceed 95% for legible documents.
  - The app interface must be intuitive, requiring no more than 3 steps to complete any use case.
  - Support for English and Bangla(Optional) languages.
- **Reliability Requirements**:
  - The app must achieve 99.9% uptime for online features.
  - Computer vision(OCR) accuracy must exceed 95% for legible documents.

5 Supporting Information

The supporting information enhances the usability of the Software Requirements Specification (SRS) for the MedWise Application. It includes:

- **Table of Contents:** A detailed table of contents is provided at the beginning of the SRS to facilitate navigation through the document's sections, including Introduction, Overall Description, Specific Requirements, and Supporting Information.
- **Index:** An index of key terms such as "MedWise," "OCR," "NLP," "Computer Vision," "AI System," "Digital Report Management," "Health Chatbot Assistant," and "Export Medical History" is included in Appendix A to assist readers in locating specific topics quickly.
- **Appendices:** The following appendices are included to provide additional context and details:
  - **Appendix A: Index of Key Terms** – Lists key terms and their locations within the SRS for quick reference.

- **Appendix B: Use-Case Storyboards** – Contains storyboards illustrating the user interactions for Digital Report Management, Health Chatbot Assistant, and Export Medical History use cases. These storyboards visually depict the main success scenarios and alternative flows.
- **Appendix C: User-Interface Prototypes** – Includes wireframes and mockups of the MedWise app's key interfaces, such as the dashboard, upload screen, chatbot interface, and export feature, to clarify the user experience.

The appendices are considered part of the requirements and should be reviewed alongside the main SRS document to ensure a comprehensive understanding of the MedWise Application's specifications.

5.1 Index of Key Terms

5.2 Use-Case Storyboards

[Placeholder: Detailed storyboards for Digital Report Management, Health Chatbot Assistant, and Export Medical History use cases will be included here. These will illustrate user interactions, including main success scenarios and alternative flows, as described in Section 3.4.]

5.3 User-Interface Prototypes

[Placeholder: Wireframes and mockups for the MedWise app's dashboard, medical record upload screen, chatbot interface, and export feature will be included here to provide a visual representation of the user interface.]