SOFTWARE REQUIREMENTS SPECIFICATION (SRS)

(PATIENT QUEUE SYSTEM)

PREPARED BY:

LAIBA FATIMA

REG #:

2023-BS-AI-047

SUBMITTED TO:

MISS IRSHA QURESHI

DEGREE / SEM :

BS AI 3RD SEMESTER

Table of Contents

1. Introduction

1.1 Purpose
This document specifies the requirements for the Patient Queue System (PQS). The PQS streamlines patient management, doctor assignment, medicine prescription, and consultation logging.

1.2 Intended Audience and Reading Suggestions
- **Primary Audience:** Developers, testers, and project supervisors.
- **Secondary Audience:** Hospital administrators and IT staff.

1.3 Project Scope
The PQS focuses on:

- Adding patients to a queue.
- Randomly assigning doctors and medicines.
- Logging consultations to a file.

2. Overall Description

2.1 Product Perspective
PQS is a standalone desktop application with a terminal-based interface.

2.2 Product Features
- **Patient Queue Management:** Add patients with their information.
- **Doctor Management:** List available doctors.
- **Consultation Processing:** Assign doctors, prescribe medicines, and log details.

2.3 User Classes and Characteristics
- **Receptionist:** Adds patients and views doctor lists.
- **System:** Processes patient consultations.

2.4 Operating Environment
- Platform: Any system supporting C++.
- Interface: Command-line interface (CLI).

2.5 Design and Implementation Constraints

- Standalone application.
- Limited to terminal interaction.

2.6 Assumptions and Dependencies

- Patients provide accurate information.
- A hospital_log.txt file is accessible for logging.

3. System Features

3.1 Patient Queue Management

- **Description:** Add patients with their name, age, disease, and symptoms.
- **Functional Requirements:**
    - Prompt for patient details.
    - Display confirmation of queue addition.

3.2 Doctor Management

- **Description:** List all available doctors.
- **Functional Requirements:**
    - Assign unique IDs to doctors.
    - Display doctor names and IDs.

3.3 Consultation Processing

- **Description:** Process a patient by assigning a doctor, prescribing medicine, and assigning a room.
- **Functional Requirements:**
    - Randomly assign a doctor and medicine.
    - Generate a room number.

3.4 File Logging

- **Description:** Save consultation details to a file.
- **Functional Requirements:**
    - Log patient, doctor, medicine, and room information.

4. External Interface Requirements

4.1 User Interfaces
- Text-based CLI.

4.2 Hardware Interfaces
- Standard PC hardware (keyboard and monitor).

4.3 Software Interfaces
- Local file system for logging.

4.4 Communications Interfaces
- None required.

5. Other Nonfunctional Requirements

5.1 Performance Requirements
- Response time for user actions should not exceed 2 seconds.

5.2 Safety Requirements
- Ensure file access is error-free to avoid data loss.

5.3 Security Requirements
- Restrict log file access to authorized personnel.

5.4 Software Quality Attributes
- **Usability:** Easy-to-use CLI.
- **Reliability:** Minimal crash risk during standard operations.

6. Modules

1. **Patient Module:** Handles patient data input and display.
2. **Doctor Module:** Manages doctor assignment and listing.
3. **Hospital Module:** Oversees overall operations, including queue and consultation management.

7. Other Requirements

- Maintain a consistent coding style for maintainability.

Appendices

Appendix A: Glossary

- **CLI:** Command Line Interface.
- **Queue:** A collection of patients waiting for consultation.

Appendix B: Issues List

- **Data Security:** Log files stored in plain text.
- **Scalability:** Limited to single-user operations.

8. Use Case Diagram

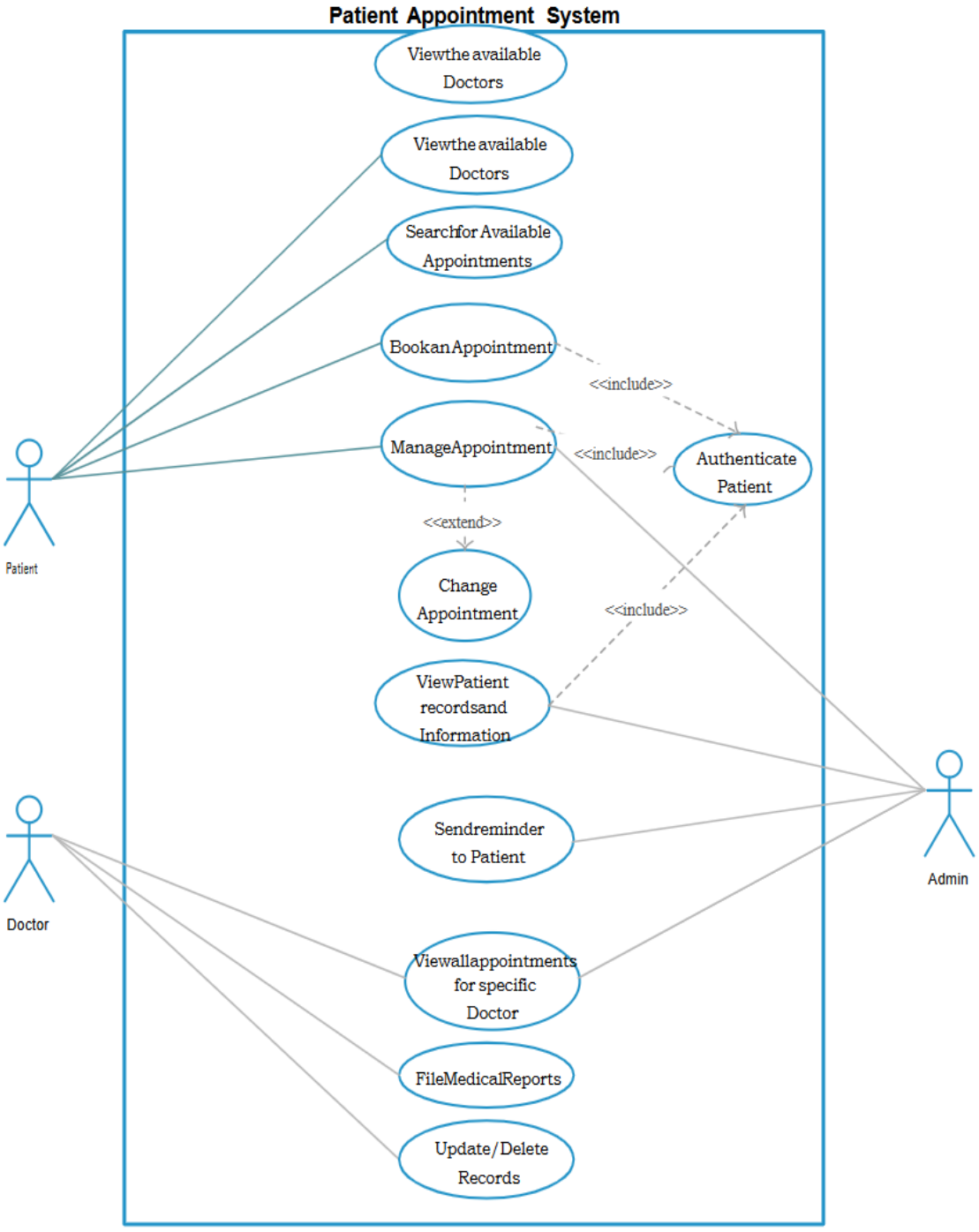
Actors:

1. Receptionist (User adding patients and viewing doctors).
2. System (Processes patients and logs details).

Use Cases:

- Add Patient.
- View Doctors.
- Process Patient (assign doctor, prescribe medicine, assign room).
- Log Consultation Details.

DIAGRAM UML


Patient Appointment System
Viewthe available Doctors
Viewthe available Doctors
Searchfor Available Appointments
Bookan Appointment
<<include>>
ManageAppointment
<<include>>
Authenticate Patient
<<extend>>
Change Appointment
<<include>>
ViewPatient recordsand Information
Sendreminder to Patient
Viewallappointments for specific Doctor
FileMedicalReports
Update/Delete Records
Patient
Doctor
Admin