SOFTWARE REQUIREMENTS SPECIFICATION
For
Hospital Appointment Scheduler
Prepared by: [Epic Coders]

1. Introduction

1.1 Purpose

The purpose of this document is to outline the requirements for the development of a Hospital Appointment Scheduler. This system aims to enhance the efficiency of appointment management processes within the hospital, providing a user-friendly web-based platform for both patients and administrative staff.

1.2 Document Conventions

The document follows a standardized format:

- Main title: Times New Roman, Bold, Font Size: 14
- Subtitle: Times New Roman, Bold, Font Size: 12
- Body: Times New Roman, Font Size: 12

1.3 Scope of Development Project

The Hospital Appointment Scheduler is designed to integrate seamlessly with the existing Hospital Management System. It empowers patients to manage appointments, allows doctors to access schedules, and enables administrative staff to oversee and manage the broader scheduling ecosystem. The system will operate on common web browsers and major operating systems.

1.4 Definitions, Acronyms, and Abbreviations

- JAVA: Platform independence
- SQL: Structured Query Language
- ER: Entity Relationship
- UML: Unified Modeling Language
- IDE: Integrated Development Environment
- SRS: Software Requirement Specification

1.5 References

Books:

- "Software Requirements and Specifications: A Lexicon of Practice, Principles and Prejudices" by Michael Jackson
- "Software Requirements" (Microsoft) Second Edition by Karl E. Wiegers
- "Software Engineering: A Practitioner’s Approach" Fifth Edition by Roger S. Pressman

Websites:

- slideshare.net
- ebookily.net

2. Overall Descriptions

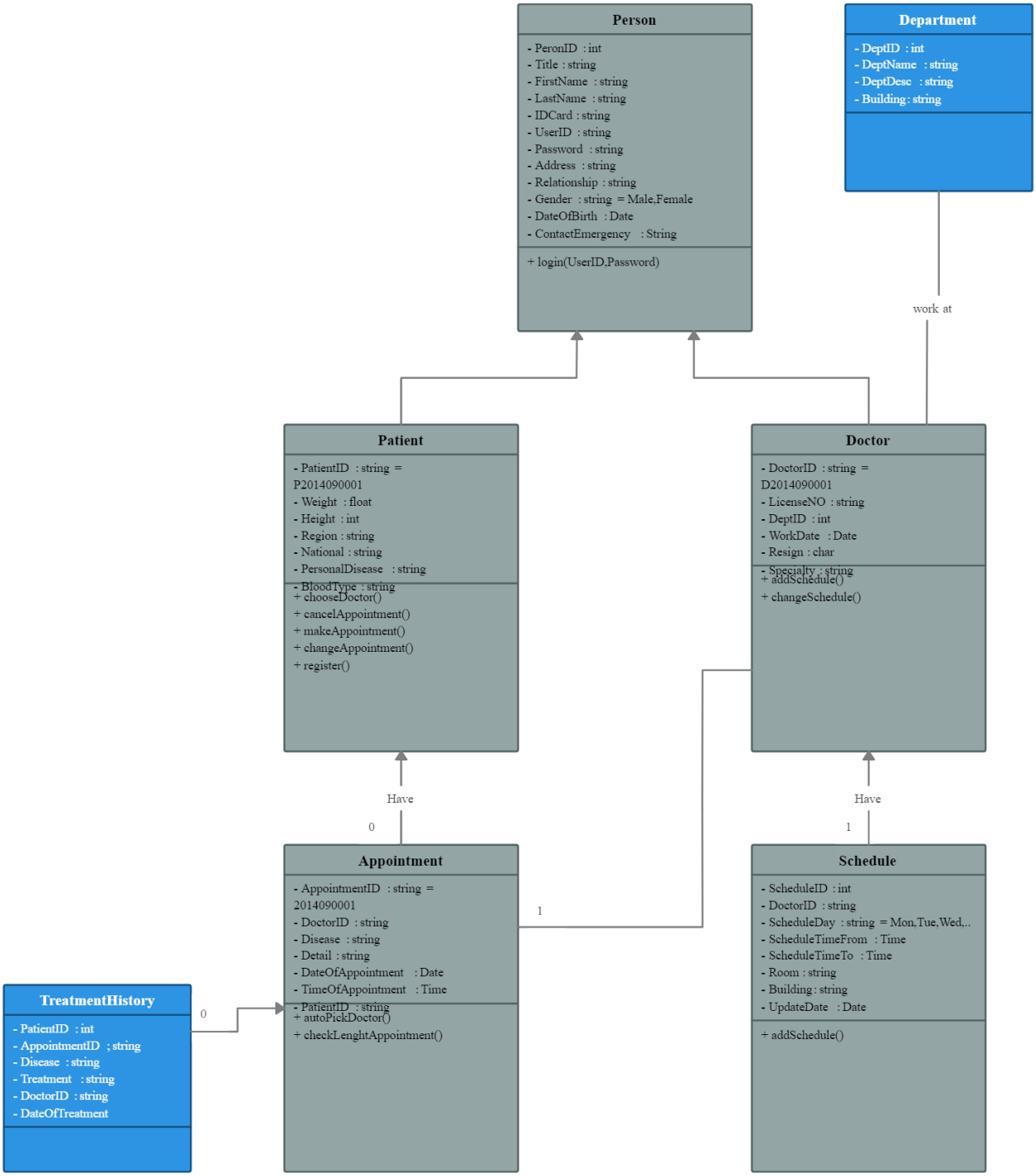
2.1 Product Perspective
The Hospital Appointment Scheduler interfaces with the existing Hospital Management System. The system accommodates patients, doctors, and administrative staff, providing features like appointment scheduling, rescheduling, and cancellation. A user-friendly interface is essential, allowing users to interact seamlessly with the system.
2.2 Product Function
The system facilitates real-time appointment management, ensuring ease of use for both patients and staff. An Entity Relationship Diagram illustrates the system's data structure, managing book issues, returns, fines, and generating reports. The librarian acts as the administrator, overseeing user and book management.
2.3 User Classes and Characteristics
Users include patients, doctors, and administrative staff. Librarians, acting as administrators, have privileges to issue books, view categories, and manage accounts. Patients can view available books, issue/renew/return, and search for specific books.
2.4 Operating Environment
The system operates in a Windows environment, compatible with major browsers. Hardware requirements include a dual-core CPU, 40GB hard disk, 15” color monitor, and 256 MB RAM. Internet connectivity is essential for system access.
2.5 Assumptions and Dependencies
Assumptions include error-free coding, user-friendly design, and 24/7 system availability. Dependencies involve specific hardware and software requirements, end-user understanding, and regular system maintenance.
2.6 Requirement
Software Configuration:

- Front End: Java (supported by Sun Microsystems)
- Back End: Microsoft SQL Server
- Operating System: Windows NT, Windows 98, Windows XP
- Language: Java Runtime Environment, NetBeans 7.0.1
- Database: MS SQL Server

Hardware Configuration:

- Processor: Pentium(R) Dual-core CPU
- Hard Disk: 40GB
- RAM: 256 MB or more

2.7 Data Requirement
The system processes queries to the database, providing solutions and user account details. Inputs include user queries (e.g., create an account), and outputs include details of user accounts and book availability.


Person
- PeronID : int
- Title : string
- FirstName : string
- LastName : string
- IDCard : string
- UserID : string
- Password : string
- Address : string
- Relationship : string
- Gender : string = Male,Female
- DateOfBirth : Date
- ContactEmergency : String
+ login(UserID,Password)
Department
- DeptID : int
- DeptName : string
- DeptDesc : string
- Building : string
work at
Patient
- PatientID : string = P2014090001
- Weight : float
- Height : int
- Region : string
- National : string
- PersonalDisease : string
- BloodType : string
+ chooseDoctor()
+ cancelAppointment()
+ makeAppointment()
+ changeAppointment()
+ register()
Doctor
- DoctorID : string = D2014090001
- LicenseNO : string
- DeptID : int
- WorkDate : Date
- Resign : char
- Specialty : string
+ addSchedule()
+ changeSchedule()
Have
0
Have
1
Appointment
- AppointmentID : string = 2014090001
- DoctorID : string
- Disease : string
- Detail : string
- DateOfAppointment : Date
- TimeOfAppointment : Time
- PatientID : string
+ autoPickDoctor()
+ checkLenghtAppointment()
1
Schedule
- ScheduleID : int
- DoctorID : string
- ScheduleDay : string = Mon,Tue,Wed,..
- ScheduleTimeFrom : Time
- ScheduleTimeTo : Time
- Room : string
- Building : string
- UpdateDate : Date
+ addSchedule()
TreatmentHistory
- PatientID : int
- AppointmentID ; string
- Disease : string
- Treatment : string
- DoctorID : string
- DateOfTreatment
0

3. External Interface Requirement

3.1 GUI
The graphical user interface (GUI) of the Hospital Appointment Scheduler ensures a user-friendly experience for administrators, doctors, and patients. It facilitates efficient appointment scheduling, quick access to patient records, and customizable layouts.
3.2 Login Interface
The system incorporates a secure login interface, allowing authorized users, including administrators, doctors, and staff, to access the scheduler. Error messages are displayed for incorrect login credentials, ensuring data security.
3.3 Appointment Scheduling
Users can efficiently schedule appointments by selecting available time slots, doctors, and departments. The scheduler provides a clear and intuitive interface for managing appointments.
3.4 Patient Records
Healthcare providers have access to patient records, enabling them to view medical histories, prescriptions, and previous appointments. The interface is designed for easy navigation and retrieval of relevant patient information.
3.5 Admin Dashboard
Administrators have a control panel for user management, appointment tracking, and system configuration. The design adheres to standard templates, ensuring simplicity and ease of use.

4. System Features

The Hospital Appointment Scheduler guarantees secure user authentication, efficient appointment monitoring, and adherence to healthcare regulations. Patient confidentiality is prioritized, and the system prevents unauthorized access.

5. Other Non-functional Requirements

5.1 Performance Requirement
The system's performance is optimized for speed, accuracy, and the ability to handle a large volume of appointment data. Error handling mechanisms prevent data loss and ensure uninterrupted scheduling.
5.2 Safety Requirement
Regular backups of the database are essential to safeguard against data loss in case of system failures or crashes. A power backup (UPS/inverter) is in place to mitigate risks related to power supply interruptions.
5.3 Security Requirement
The system incorporates secure databases, user access restrictions, proper authentication, and encryption to ensure patient data integrity and confidentiality.
5.4 Requirement Attributes

Attributes include multiple admin access, open-source nature, user-friendly design, and easy download/installation. The system aims for quality, accessibility, and adherence to healthcare standards.

5.5 Business Rules

The system enforces business rules, including compliance with healthcare regulations, appointment scheduling policies, and patient confidentiality. Admin and user roles must adhere to established rules.

5.6 User Requirement

Users, including administrators, doctors, and staff, require a secure and user-friendly system. Admin facilities include backup/recovery, password recovery, and routine system maintenance.

6. Other Requirements

6.1 Data and Category Requirement

User categories (admin, doctor, staff) have specific access rights. Patient records are categorized, and data related to each patient is stored in a standardized format.

6.2 Appendix

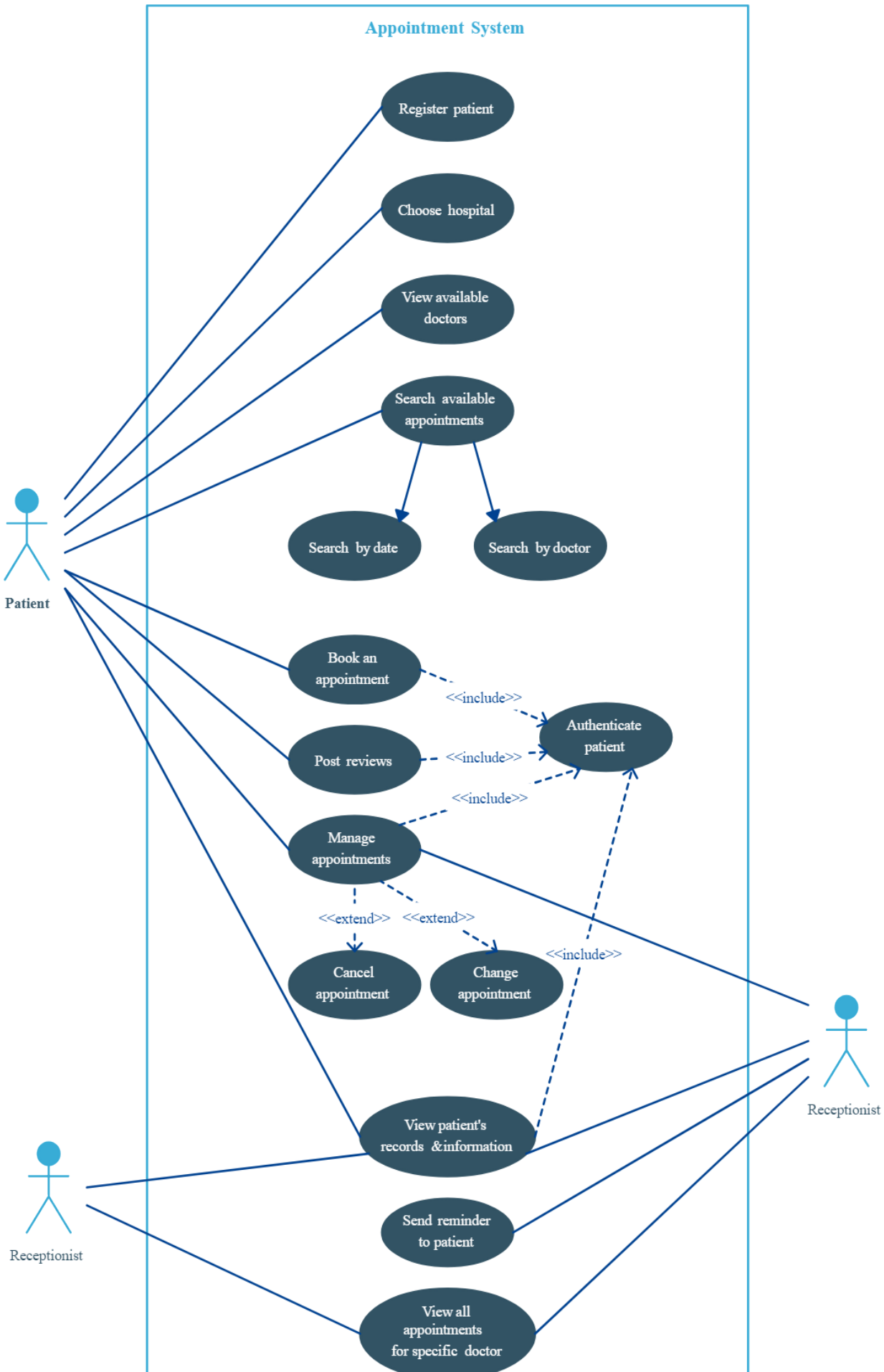
A: Admin, Abbreviation, Acronym, Assumptions; D: Data requirement, Dependencies; G: GUI; L: Login Interface; M: Medical Records, Admin Dashboard; N: Non-functional Requirement; O: Operating environment; P: Performance, Purpose; R: Requirement, Requirement attributes; S: Safety, Security, System features; U: User, User class and characteristics, User requirement.

6.3 Glossary

Includes conventions and acronyms such as administrator, user, GUI, login interface, medical records, admin dashboard.

6.4 Class Diagram

Illustrates main classes (Admin, Doctor, Staff, Patient) and their relationships, showcasing associations, aggregations, and generalizations.


Appointment System
Register patient
Choose hospital
View available doctors
Search available appointments
Search by date
Search by doctor
Patient
Book an appointment
<<include>>
Authenticate patient
Post reviews
<<include>>
<<include>>
Manage appointments
<<extend>>
<<extend>>
Cancel appointment
Change appointment
<<include>>
Receptionist
View patient's records &information
Send reminder to patient
View all appointments for specific doctor
Receptionist