Software Requirement Specification Document

For

Hospital Management System

Prepared by

Aniruddh Dwivedi & Akshaj Kumar Bhardwaj

(23BCE8304) (23BCE8277)

School of Computer Science and Engineering



10th January 2025

Table of Contents

Page Numbers

1. Introduction
   1. Purpose
   2. Intended Customers/Users
   3. Product Scope
   4. Outline of the document
2. Overall Description
   1. Product Perspective
   2. Product Functions
   3. User Classes
   4. Design and Implementation Constraints
3. Diagrams
   1. UML Diagrams
   2. ER Diagrams
   3. DFD Diagrams
   4. Assumptions and Dependencies
4. External Interface Requirements/Screenshots
   1. User Interfaces
   2. Hardware Interfaces
   3. Software Interfaces
   4. Communication Interfaces
5. Other Non-Functional Requirements
   1. Performance Requirements
   2. Safety Requirements
   3. Security Requirements
   4. Software Quality Attributes
   5. Business Policies/Rules (if any)
6. Testing
   1. Block-Box Testing
   2. White-Box Testing
7. Conclusion
8. References
9. Acknowledgements
10. **Introduction**

This Software Requirement Specification Document for the development of a hospital entails the different aspects involved in the creation and optimization of the Hospital Management System.

* 1. Purpose

The purpose of the hospital management system is to help a hospital or any other large-scale patient care system to be enabled to store patient, employee, and transport records in an organized manner over the course of its operation.

* 1. Intended Customers/Users

The intended users of the designed system are hospital staff and human resources managers who would find themselves eased in their jobs through the use of the provided software product.

* 1. Product Scope

The Hospital Management System aims to streamline the operational and administrative processes within a hospital environment by providing a centralized, user-friendly platform for managing patient records, employee details, appointments, inventory, billing, and other critical hospital functions.

* 1. Outline of the document

The document provides an overview on the workings of the Hospital Management System, including its purpose, intended users, and scope. It outlines the overall description of the product, detailing its perspective, key functions, user classes, and design constraints. Visual representations such as UML, ER, and DFD diagrams are included to illustrate the system architecture and workflows, along with assumptions and dependencies.

1. Overall Description

The Hospital Management System is a standalone software designed to digitize and centralize hospital operations, integrating seamlessly with existing hospital infrastructure, providing a cohesive platform for managing patient records, staff data, appointments, billing, and more.

* 1. Product Perspective

The Hospital Management System is designed to digitize and centralize hospital operations. It integrates with existing hospital infrastructure to provide a cohesive platform for managing patient records, staff data, appointments, billing, inventory, and more. The system is designed to be scalable and readily accessible by different levels of technically inclined individuals.

* 1. Product Functions

The hospital management system provides the following key functions:

* **Patient Management:** Registration and appointment scheduling.
* **Staff Management:** Storing employee details and performance metrics.
* **Billing and Payments:** Automating billing processes and payment tracking.
* **Inventory Management:** Monitoring stock levels of medicines and medical supplies.
  1. User Classes

Three basic classes of users are decided to be availed to the system’s potential user-base, these are:

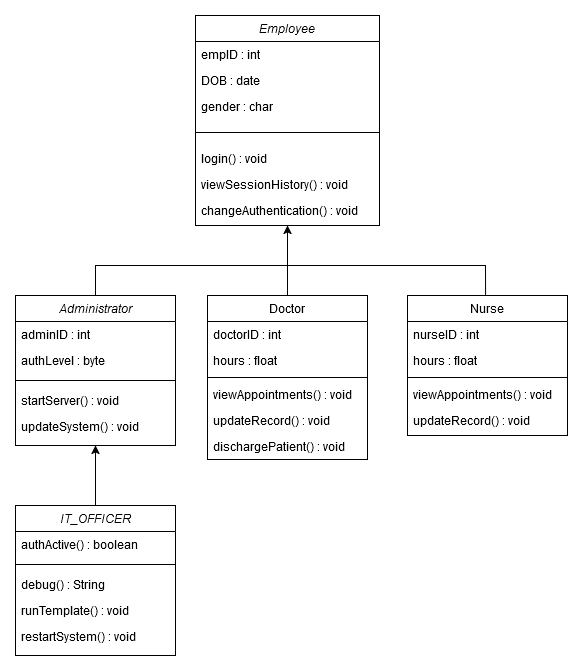
* Administrators: Monitor the use of the system by sub-classes of users and make executive decisions regarding hospital administrative processes
* Standard Employees: Manage the regular record maintenance of the system’s records.
* Technical Staff: Maintain the software and ensure data integrity.
  1. Design and Implementation Constraints

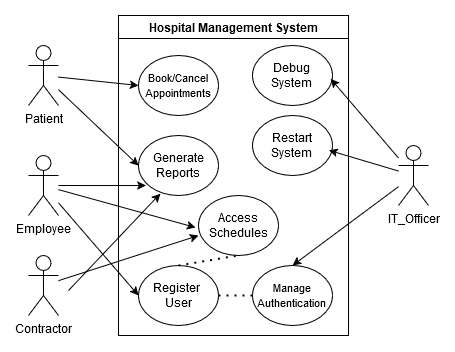
Data Security: Encryption and secure access are needed control mechanisms to protect sensitive data.

Hardware Limitations: The system must operate efficiently on existing hospital hardware.

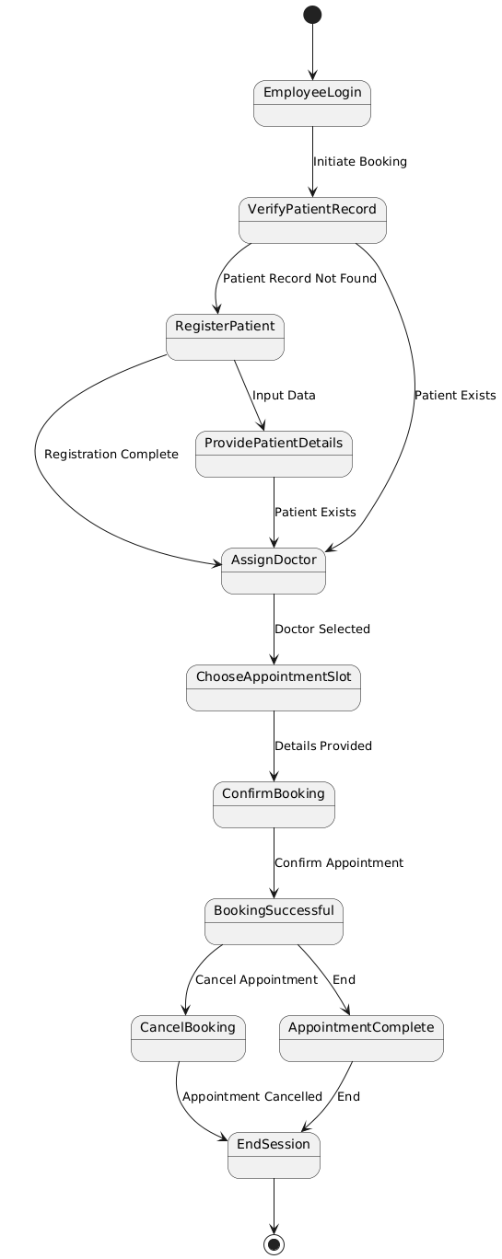
Technical Expertise: Hospital staff may have limited technical knowledge, which may lead to them being unable to manage the workings of the system.

1. **Diagrams**
   1. UML Diagrams

3.1.1. Class-Diagram for representation of the system’s user profiles

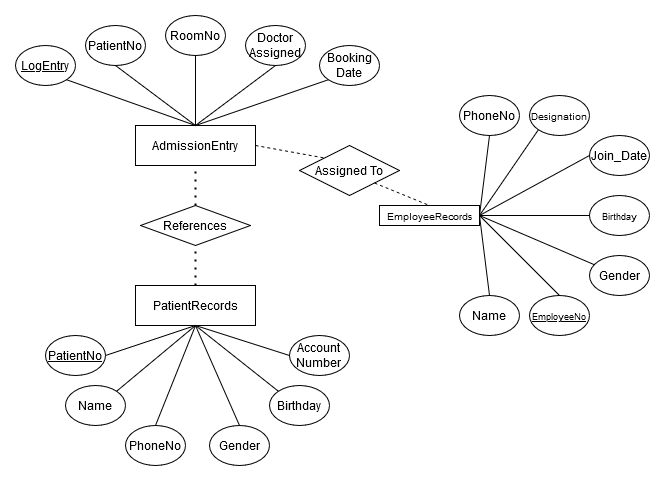


3.1.2. Use-Case diagram describing the different roles played by different user types

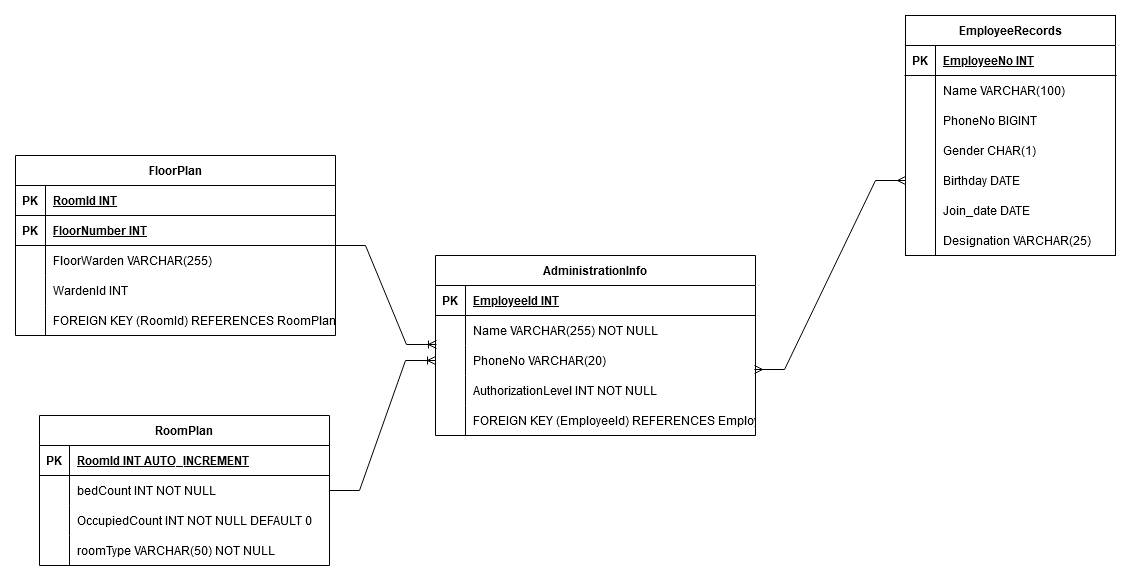


3.1.3. state-machine diagram describing the system’s patient booking interface

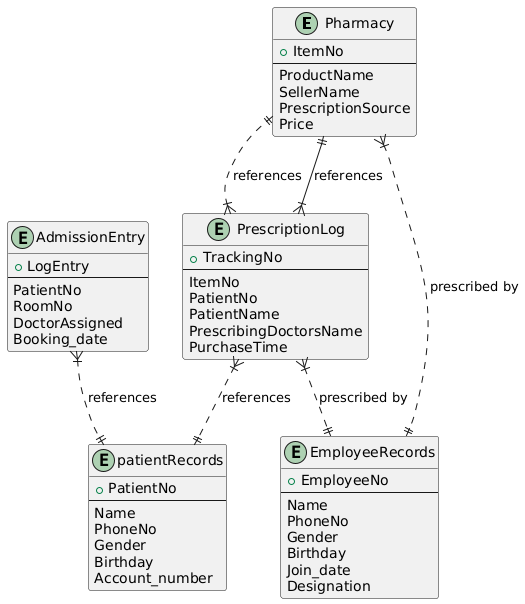
3.2. ER Diagrams

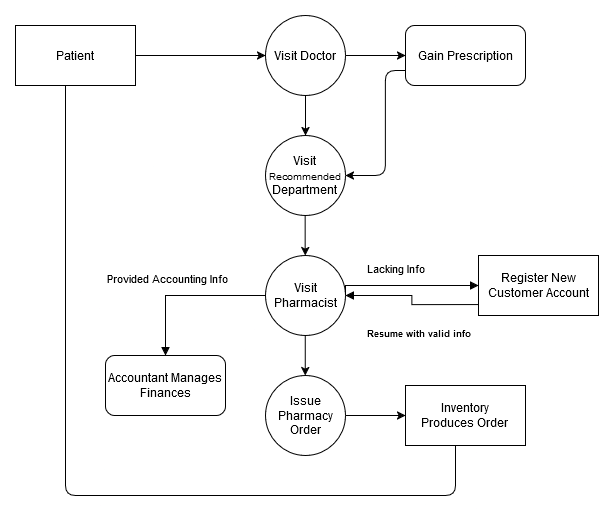


3.2.1. Chen’s ER diagram notation for patient record management

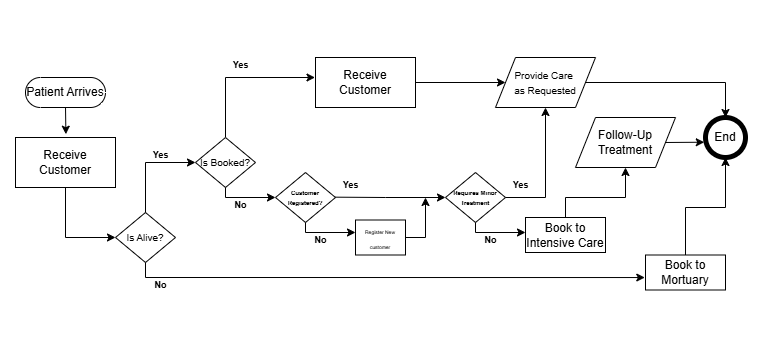


3.2.2. Barker’s ER diagram notation for Room Assignment Detail

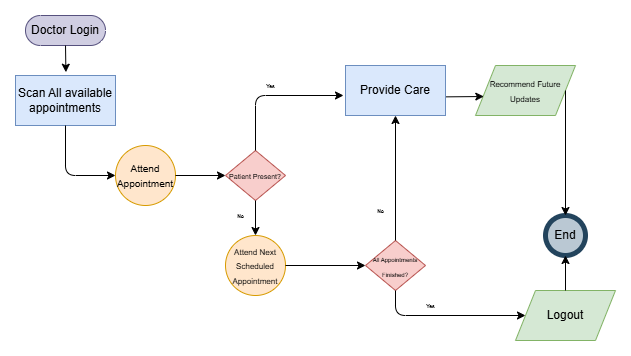
 3.2.3. Crow’s Foot ER diagram for representing Pharmacy Records

3.3. DFD Diagrams

3.3.1. DFD diagram for representing Customer to Department interactions



3.3.2. DFD diagram for representing Customer Intake Procedure



3.3.3. DFD diagram for representing Doctor’s daily work procedure

3.4. Assumptions and Dependencies

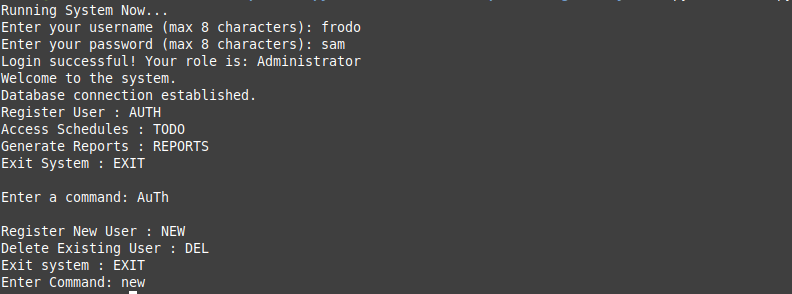
The current model of the system expects a decent level of technical know-how being available to all members of staff and may as such prove quite challenging to have to learn from scratch. The System relies on well-maintained infrastructure that must be regularly updated and be readily accessible for all members of staff at their respective authorization levels.

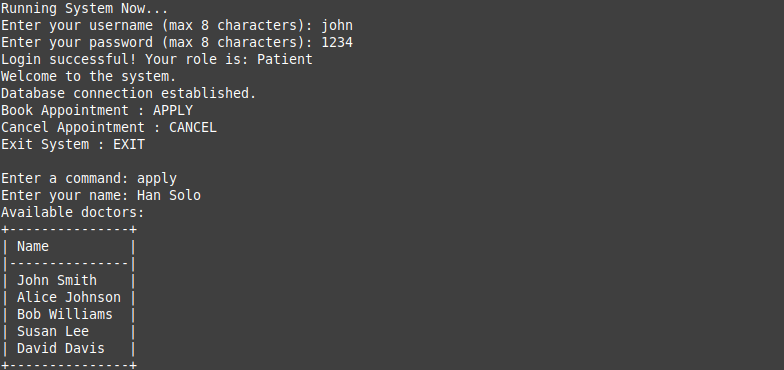
1. **External Interface Requirements / Screenshots**

The Hospital Management System is a standalone software designed to digitize and centralize hospital operations, as part of its implementation interface, certain design prospects for different software components and their interactivity are detailed thus.

* 1. User Interfaces

The Present iteration of the system uses a command-line interface for displaying menus and performing system tasks. This allows the user to easily gain access to different system functionalities while keeping up high-level data abstraction as different users have access to only specific sets of commands and only ever access part of the data.



 Commands for administrators in the system and their implements for accessibility

Commands for patients in the system and their implements for accessibility

* 1. Hardware Interfaces

The present iteration of the system relies on locally stored data accessed via simple and easy to organise computer architecture, a user need only connect to the physical server instance to access all records and manipulate them, these changes may in future iterations reflect across other concurrently running user instances. The basic system specifications for running the system are as follows:

* 64-bit processor with CISC configuration
* 2 GB RAM
* 16 MB dedicated VRAM
* 100 MB available disc space
* 1920x1080 or equivalent 16:9 aspect capable displays

Most modern computer systems would qualify these requirements, making the system very easy to access.

* 1. Software Interfaces

Commensurate to the previously specified hardware requirements, the peripheral media may run the system if it qualifies the following software requirements:

* 64-bit Linux or Windows Operating Systems released after 2024
* MySQL Server version: 8.0.41-0ubuntu0.24.04.1 (Ubuntu) or equivalent
* Python 3.12.3 with [GCC 13.3.0] on linux or equivalent
* Python modules:

-tabulate -pymysql -enum

Most modern machines should qualify the stated software requirements and those which do not may be brought up to standard quite accessibly.

* 1. Communication Interfaces

The Present iteration of the system need only to have basic communication peripherals as availed in most present-day computers provided to it, including:

* Microphone-Based single-channel P.A. systems
* Digital display and alert systems
* Building wide well-organised intercoms

With the software front being configured to work seamlessly alongside the necessary hardware, communication interfaces in the system may help to reliably provide users access to different system peripherals.