High Level Design

Hospital Management System

Revision Number: 1.0

Poonam Merwade, Karnataka

Associate ID:844333

Change Record

|  |  |  |  |
| --- | --- | --- | --- |
| **Revision** | **Date** | **Author** | **Changes** |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Contents

# Introduction

* 1. Why this High Level Design Document?
  2. Scope
  3. Definitions
  4. Overview

# General Description

* 1. Product Perspective
  2. Tools used
  3. General Constraints
  4. Assumptions
  5. Special Design aspects

# Design Details

* 1. Main Design Features
  2. Application Architecture
  3. ER diagram
  4. Technology Architecture
     1. Web Application Architecture
  5. Error Handling
  6. Help
  7. Security
  8. Portability
  9. Reusability
  10. Major classes

# Introduction

* 1. Why this High Level Design Document?

The purpose of this High Level Design (HLD) Document is to add the necessary detail to the current project description to represent a suitable model for coding. This document is also intended to help detect contradictions prior to coding, and can be used as a reference manual for how the modules interact at a high level.

* 1. Scope

The HLD documentation presents the structure of the system, such as the database architecture, application architecture (layers), application flow (Navigation), and technology architecture. The HLD uses non-technical to mildly-technical terms which should be understandable to the administrators of the system.

* 1. Definitions
* HMS : Hospital Management System.
* MySQL server: A database management system.
* Java : A language used to interface between backend and Middleware.
* ER : Entity Relation diagram.

* 1. Overview

The HLD will:

* + - present all of the design aspects and define them in detail
    - describe the user interface being implemented
    - describe the hardware and software interfaces
    - describe the performance requirements
    - include design features and the architecture of the project
    - list and describe the non-functional attributes like:
      * security
      * reliability
      * maintainability
      * portability
      * reusability
      * application compatibility
      * resource utilization

# General Description

* 1. Product Perspective

The Hospital Management System will be comprised of several different components. Java will be used as the language. MySQL is used for retrieve, insert, delete, and update the database. The setup will allow Hospital front desk to login and provide details of the specialists like his name, specialty, available time and date to the caller or patient who called the hospital front desk and it provides the availability of beds to the patient, we are spring boot for this project.

* 1. Tools used

1. The use case diagram is used to present the details of the project and design phases of the project in a simple way.
2. The project will have a relational database backend that is SQL based. The actual software used is MySQL workbench.
3. Spring tool suite (STS) is used for middleware.
4. Postman is used for retrieving, posting, updating and deleting the data.

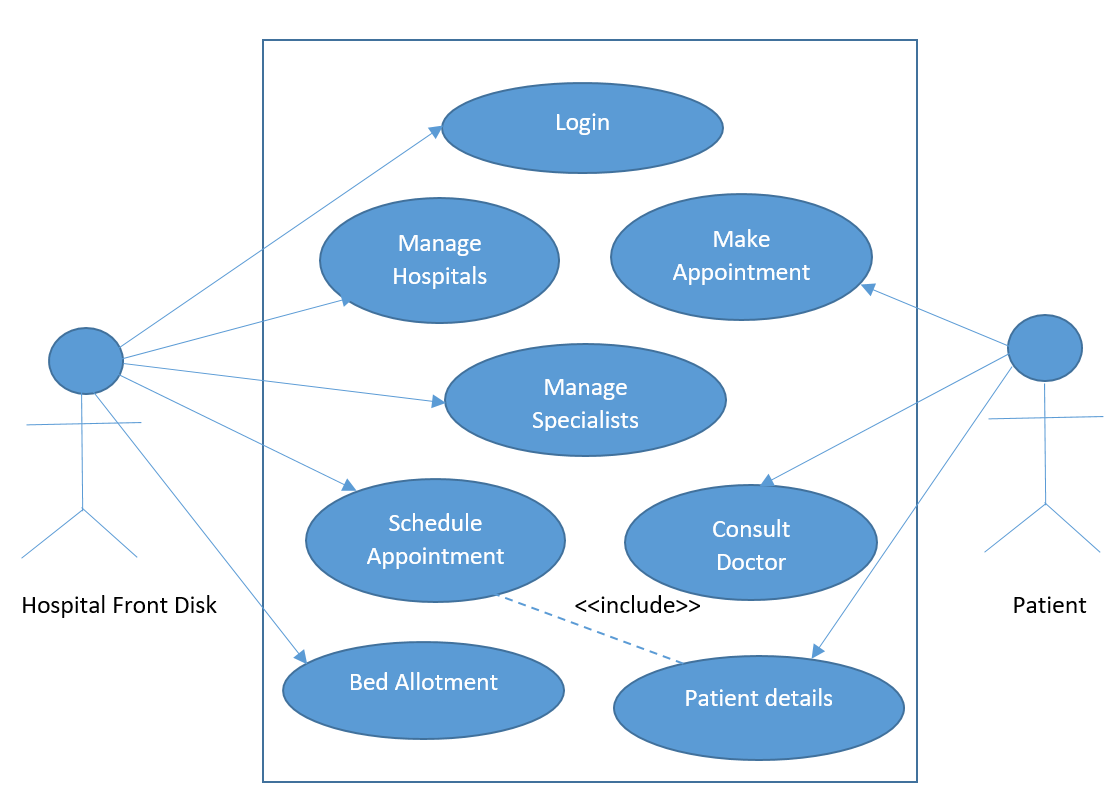
* 1. General constraints

The hospital management system is user friendly. Hospital front desk has authority to appoint the patient by observing availability of specialists on particular day and time also checks the availability of beds. Only patient provide his name and type of specialist he want to get appointment.

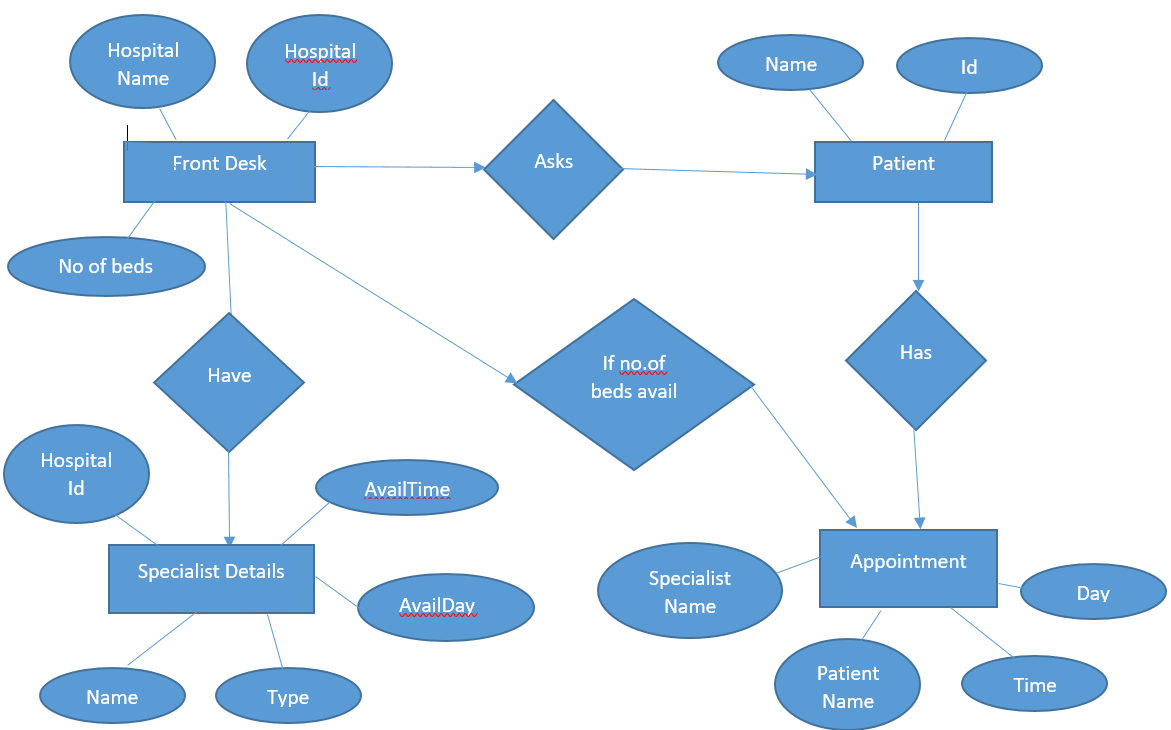
1. **Design details**
   1. Main Design features

The main design features includes the architecture, the database, process relation. In order to make these designs easier to understand, the design has been illustrated in attached diagrams (ER, Use Case).

* 1. Application architecture(Use case Diagram)



* 1. Entity Realtionship diagram



* 1. **Technology architecture**

3.4.1. Web application architecture

The front end of the program is a web application. The hospital front desk should login into the system and it has privilege to check the details of the hospitals and availability of beds and can appoint the patient at a given time, only front desk has authority to add the details of the patient details who are getting appointment.

* 1. **Error Handling**

Should errors be encountered, an explanation will be displayed as to what went wrong. An error will be defined as anything that falls outside the normal and intended usage.

* 1. **Help**

Help will come in the form of all the documentation created prior to coding, which explain the intended uses. Should time allow, detailed instructions will be written on how to create and implement the system with the intentions of publishing as an Open Source solution.

* 1. **Security**

For hospital front desk login purpose basic authentication will be used so that particular person from front desk should provide appointment to the patient.

* 1. **Portability**

This system should have the ability that, once it is together, the entire system should be able to be physically moved to any location. For everything to work properly, all components should be compiled from source.

* 1. **Reusability**

The code written and the components used should have the ability to be reused with no problems. Should time allow, and detailed instructions are written on how to create this project, everything will be completely reusable to anyone.

* 1. **Major classes**

There are majorly 4 classes.

* Front desk class – A patient can call the hospital front desk.
* Specialists class – Details of the Specialists.
* Patient class – Details of the patient.
* Appointment class- Details of the appointment will be provided.