Table Of Contents:

[Hospital Management System Report 2](#_Toc171431153)

[Introduction 2](#_Toc171431154)

[System Overview 2](#_Toc171431155)

[Functionalities 2](#_Toc171431156)

[Patient Management: 2](#_Toc171431157)

[Doctor Management: 2](#_Toc171431158)

[Appointment Management: 2](#_Toc171431159)

[File Management: 2](#_Toc171431160)

[Technologies Used 2](#_Toc171431161)

[ Programming Language 2](#_Toc171431162)

[ Libraries 2](#_Toc171431163)

[System Design 2](#_Toc171431164)

[Class Structure 3](#_Toc171431165)

[Patient Class: 3](#_Toc171431166)

[Doctor Class: 3](#_Toc171431167)

[Appointment Class: 3](#_Toc171431168)

[HospitalManagementSystem Class: 3](#_Toc171431169)

[File Management 3](#_Toc171431170)

[Data Files: 3](#_Toc171431171)

[Functional Flow 3](#_Toc171431172)

[Initialization: 3](#_Toc171431173)

[User Interactions: 3](#_Toc171431174)

[Data Management: 4](#_Toc171431175)

# Hospital Management System Report

## Introduction

The Hospital Management System (HMS) is designed to facilitate efficient management of patient records, doctor schedules, and appointments within a hospital setting. This system aims to streamline administrative tasks, enhance patient care, and improve overall operational efficiency.

## System Overview

### Functionalities

### Patient Management:

* + Record patient details including ID, name, age, gender, and medical condition.
  + Add new patients to the system.
  + Retrieve and update patient information.

### Doctor Management:

* + Manage doctor information such as ID, name, age, gender, specialty, and shift time.
  + Add new doctors to the system.
  + View and modify doctor details.

### Appointment Management:

* + Schedule appointments between patients and doctors.
  + Maintain appointment records with details like date and time.
  + Support for adding, viewing, and canceling appointments.

### File Management:

* + Load data (patients, doctors, appointments) from text files.
  + Save data back to respective files after modifications.

## Technologies Used

* Programming Language**:** Java
* Libraries**:** Standard Java libraries for file handling (java.io.\*, java.util.\*)

## System Design

## Class Structure

### Patient Class:

* + Attributes: ID, name, age, gender, medical condition
  + Methods: Constructors, getters, setters

### Doctor Class:

* + Attributes: ID, name, age, gender, specialty, shift time
  + Methods: Constructors, getters, setters

### Appointment Class:

* + Attributes: ID, patient ID, doctor ID, date, time
  + Methods: Constructors, getters, setters

### HospitalManagementSystem Class:

* + Attributes: Lists for patients, doctors, appointments
  + Methods: Loading data from files, saving data to files, managing operations for patients, doctors, appointments

## File Management

### Data Files:

* + patientFile.txt: Stores patient information (ID, name, age, gender, medical condition)
  + doctorFile.txt: Stores doctor information (ID, name, age, gender, specialty, shift time)
  + appointmentFile.txt: Stores appointment information (ID, patient ID, doctor ID, date, time)

## Functional Flow

### Initialization:

* + Upon system initialization, data from files (patientFile.txt, doctorFile.txt, appointmentFile.txt) is loaded into respective lists (patients, doctors, appointments).

### User Interactions:

* + Users (administrators or hospital staff) can interact with the system through a user interface or command-line interface.
  + They can add new patients, doctors, and appointments.
  + View existing records of patients, doctors, and appointments.
  + Modify or delete records as needed.

### Data Management:

* + Changes made to patient, doctor, and appointment records are updated in memory and saved back to the respective text files using file I/O operations.