

Hospital Management System

Members :

Member

1. Mrs.Varsha Shaha
2. Ms.Tejas Bhadane
3. Mr. Gurudas Mali
4. Ms. Divyshree Gutti
5. Ms. Mohommad Shafith

Guided By

➤ Prof. Indrakka Mali

Index

- ❑ Introduction
- ❑ Objective
- ❑ Technology to be Used
- ❑ Proposed System
- ❑ Snapshots
- ❑ Diagrams
- ❑ Advantages
- ❑ Conclusion

Introduction

- ❑ This project will automate the daily operations of LIFE-LIINE hospital. The project keeps track of the staff and patient (in-patient, out-patient) details.
- ❑ HOSPITAL MANAGEMENT is an integrated Hospital Information System which addresses all the major functional areas of multi-specialty hospital.
- ❑ It also takes care of the ward, medical, invoice and the doctor's appointment details.
- ❑ The Hospital Management enables better patient care, patient safety, patient confidentiality, efficiency, reduced cost and better management information system.

- ☐ It provides easy access to critical information thus enabling the management to take better decision on time.
- ☐ The project deals with processing of each and every department in the hospital.
- ☐ The details of Doctor and staff help the hospital to maintain the record of every person.
- ☐ The main aim of our project is to provide a paper less hospital up to 90%.It also aims at providing low-cost reliable automation of the existing systems.

Objective

- ❑ The project “Hospital management system is aimed to develop to maintain the day -to-day state of admission/discharge of patients,list of doctors ,reports generation,and etc.
- ❑ It is designed to achieve the following objectives-
 1. To computerize all details regarding patient and hospital details.
 2. Scheduling the appointment of patient with doctors to make it convenient for both.

- 3.Scheduling the service of specialized doctors and emergency properly so that facilities provided by hospital are fully utilized in effective and efficient manner.**
- 4.If the medical store issues medicines to patients ,it should reduce the stock status of the medical store and vice versa.**
- 5.The information of the patients should be kept upto date and there record should be kept in the system for historical purpose.**

Software & hardware used

- **Software Requirements**

- Operating System : Windows 7 or Windows 10
- Language : Java
- IDE : Spring Boot,PostMan
- Backend : Microsoft MySQL server

- **Hardware Requirements**

- CPU : Intel i3 processor
- RAM : 512 MB or above
- Hard Disk : 2 GB hard disk space or minimum

Back-End

- **Core Java**
- **Spring Boot**
- **Spring Data JPA**
- **Hibernate**
- **Spring Boot Web**
- **MySQL Database**
- **Thymeleaf**
- **JUnit**

Proposed System

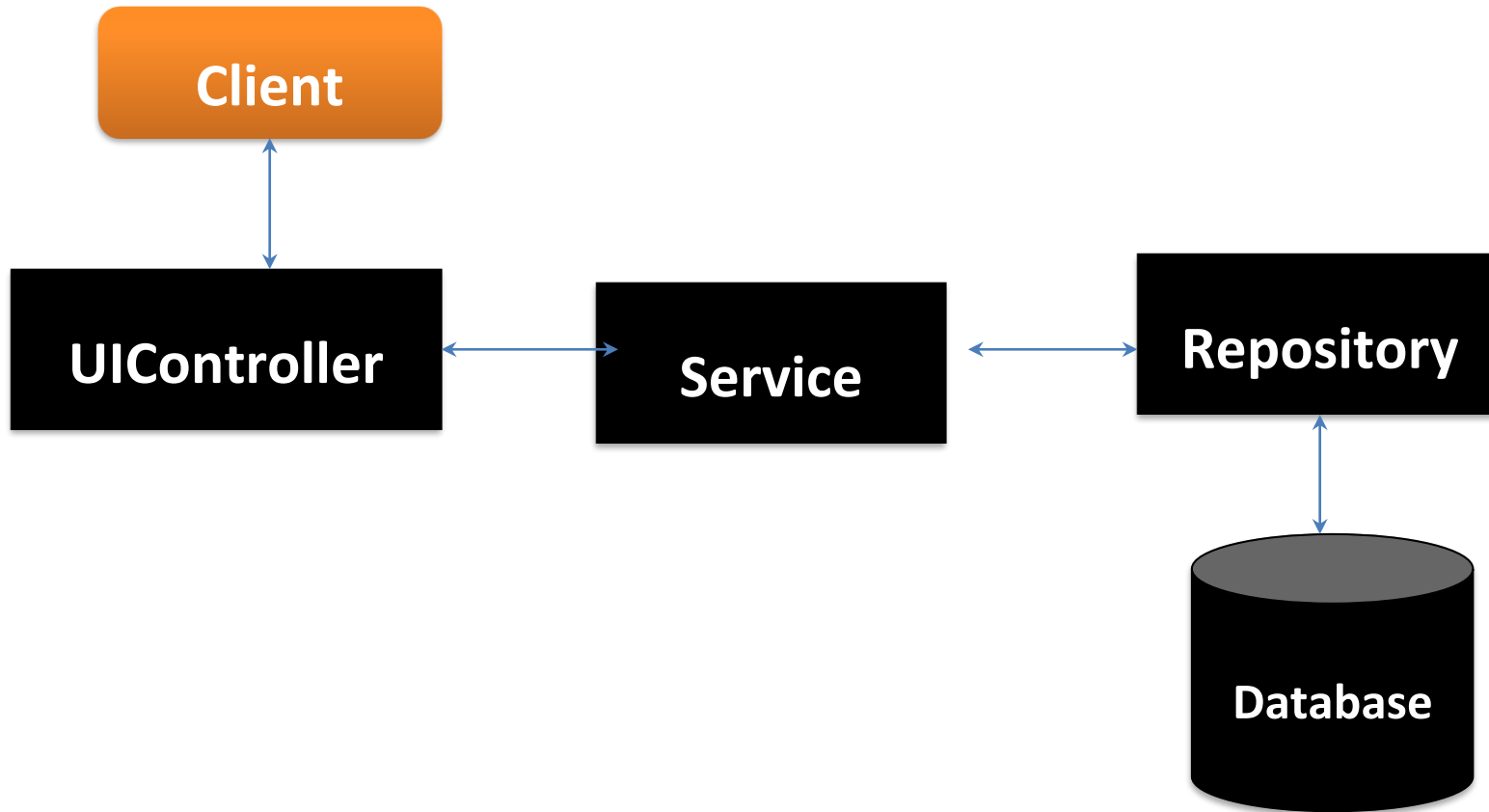
- ❑ First we have to run the Application on the browser, the index page will be shown after that if the user is registered then he/she can login the page and he/she can entering in the application.
- ❑ If the user is not registered then he/she needs to register themselves and then he/she can continuing to login and he/she can access any time.
- ❑ After that related pages will be there like if user is admin so he/she can access all the operations like insert, delete, update and search the details.
- ❑ If the user is customer then he/she can view the related record and he/she can update the record.
- ❑ And after that he/she can logged out by themselves.

About Back-end:

- ☐ There are mainly four operations will be performed by the user and admin Ex. Insert, update, retrieve/fetch, delete.
- ☐ These operation will be performed using the spring boot framework, core java and spring boot web , spring data Jpa , and hibernate.
- ☐ And thymeleaf is the design pattern for the user interface.
- ☐ For connectivity purpose we are using MySQL database. It is connected to the java and database.

Working Of Back-End

- ❑ There are mainly three stages –



- ❑ **UIController** - In Spring Boot, the UIController class is responsible for processing incoming requests, preparing a model, and returning the view to be rendered as a response.
- ❑ **Service** - Components are the class file which contains @Service annotation. These class files are used to write business logic in a different layer.
- ❑ **Repository** - Repository is a specialization of @Component annotation which is used to indicate that the class provides the mechanism for storage, retrieval, update, delete and search operation on objects.
- ❑ Repository is directly connected with the database and then it can return to the repository and response to the service

and it can return to the controller and then response send to the client.

Snapshots

Log in page

2.Inserting doctor(/addDoctor)

The screenshot displays the Postman application interface. At the top, there's a browser-like address bar and navigation tabs. Below this, a yellow banner indicates 'Working locally in Scratch Pad. Switch to a Workspace'. The left sidebar shows a 'Scratch Pad' with a collection named 'HospitalManagement' containing a 'DELETE' request. The main workspace shows a 'POST' request to 'http://localhost:8889/addDoctor'. The request body is a JSON object with the following details:

```
{  "doctorName": "Anamika Gupta",  "doctorPhone": "8778771234",  "doctorDegree": "MBBS, MD",  "doctorDepartment": "Pshycatrist",  "doctorCollege": "AIIMS Bhopal"}
```

The response is shown in the bottom panel, indicating a 'Status: 200 OK' with a time of 14.53 s and a size of 324 B. The response body is a JSON object:

```
{  "doctorId": 2,  "doctorName": "Anamika Gupta",  "doctorPhone": "8778771234",  "doctorDepartment": "Pshycatrist",  "doctorDegree": "MBBS, MD",  "doctorCollege": "AIIMS Bhopal"}
```

The bottom of the image shows the Windows taskbar with the search bar and various application icons.

3. Get record:/ finddocbyname

The screenshot shows the Postman application interface. At the top, there's a browser-like address bar with "Search the web...". Below it, the Postman header includes navigation links (Home, Workspaces, Explore), a search bar, and user options (Sign In, Create Account). A yellow banner indicates "Working locally in Scratch Pad. Switch to a Workspace".

The main workspace is divided into a left sidebar and a main panel. The sidebar contains "Scratch Pad" with "New" and "Import" buttons, and a "Collections" list with "HospitalManagement". The main panel shows a GET request to "http://localhost:8889/finddocbyname/devika". The request is configured with "GET" method and the URL. The "Params" tab is active, showing "Query Params" with a table:

KEY	VALUE	DESCRIPTION
Key	Value	Description

Below the params, the "Body" tab is active, showing a JSON response in "Pretty" format:

```
1 {
2   "doctorId": 3,
3   "doctorName": "Devika",
4   "doctorPhone": "8778753634",
5   "doctorDepartment": "Gynecologist",
6   "doctorDegree": "MBBS",
7   "doctorCollege": "GMC"
8 }
```

The status bar at the bottom shows "Status: 200 OK", "Time: 76 ms", and "Size: 305 B". The Windows taskbar at the very bottom shows the search bar, taskbar icons, and system clock (12:27:05, 26-09-2022).

4.find doctor: /Doctorbyid

The screenshot shows the Postman interface with a GET request to `http://localhost:8889/finddocbyid/2`. The response is a JSON object with the following data:

```
{
  "doctorId": 2,
  "doctorName": "Anamika Gupta",
  "doctorPhone": "8778771234",
  "doctorDepartment": "Pshycatrist",
  "doctorDegree": "MBBS, MD",
  "doctorCollege": "AIIMS Bhopal"
}
```

The interface includes a sidebar with 'Scratch Pad' and 'Collections', a top navigation bar with 'Home', 'Workspaces', and 'Explore', and a bottom status bar showing the time as 12:23:28 on 26-09-2022.

5. Delete doctor: /doctor by id

The screenshot shows the Postman application interface. At the top, there's a browser-like address bar and navigation tabs. The main workspace is titled "Scratch Pad" and shows a "DELETE" request to the URL "http://localhost:8889/deletedoctor/2". The request is saved and ready to be sent. The response body is visible, showing a success message: "1 Doctor record deleted Successfully".

Request Details:

- Method: DELETE
- URL: http://localhost:8889/deletedoctor/2
- Params: None
- Headers: 8
- Body: None

Response Details:

- Status: 200 OK
- Time: 1787 ms
- Size: 198 B
- Body: 1 Doctor record deleted Successfully

6.Update Doctor:/update/doctor

The screenshot displays the Postman application interface. At the top, there's a browser-like address bar and navigation links. Below that, a yellow banner indicates 'Working locally in Scratch Pad. Switch to a Workspace'. The left sidebar shows the 'Scratch Pad' section with a 'HospitalManagement' collection. The main workspace is configured for a PUT request to 'http://localhost:8889/updatedoctor/1'. The 'Body' tab is selected, showing a JSON payload:

```
{  "doctorName": "Shivpriya"}
```

. The 'Headers' tab shows 5 headers. The 'Test Results' tab shows a status of 200 OK, time of 983 ms, and size of 322 B. The response body is displayed in the 'Pretty' view as a JSON object:

```
{  "doctorId": 1,  "doctorName": "Shivpriya",  "doctorPhone": "0202345765",  "doctorDepartment": "Dermatologist",  "doctorDegree": "MBBS,MD",  "doctorCollege": "GandhiMedical"}
```

Advantages

Advantages:

- **Duplication of the hospital data is avoided.**
- **Avoid errors and track every single detail.**

Conclusion

- ❑ Effectiveness, efficiency, and reliability are the key aspects that make this web-based Hospital management system .
- ❑ The proposed project is very flexible to handle new modules and features as per user requirements in future.
- ❑ Well tuned Hospital Management System involves lots of important decision that should be made in the most efficient an quick way.

THANK YOU