**Project Documentation**

**Of**

**ClinApp**

**Submitted by**

**Dhruv Barke 220343120022**

**Ajinkya Hatwar 220343120006**

**Punam Londhe 220343120054**

**Ajinkya Bhutkar 220343120007**

Guide(s):

**Harshita Maheshwari**



**Infoway Technologies Private Limited**

**March 2022**

**Certificate**

This is to certify that the project entitled “**ClinApp**

”is a bonafide work of **“Dhruv Barke (220343120022), Ajinkya Hatwar(220343120006), Punam Londhe (220343120054), Ajinkya Bhutkar (220343120007).”** Submitted to Infoway Technologies Private Limited Pune in partial fulfillment of the requirement for the award of the Post Graduate Diploma in Advanced Computing.

**Aditya Hatwar Harshita Maheshwari**

**Supervisor Guide Faculty Guide**

**Declaration**

I declare that this written submission represents our ideas in our own words and where others' ideas or words have been included, we have adequately cited and referenced the original sources. We also declare that we have adhered to all principals of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in my submission. We understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

**Table of Contents:**

|  |  |  |
| --- | --- | --- |
| Sr. No. | Content | Page No. |
| 1.1 | Introduction | 5 |
| 1.2 | Problem Definition | 6 |
| 1.3 | Purpose and Scope | 6 |
| 2 | System Analysis | 7 |
| 2.1 | Functional Requirements | 7 |
| 2.1 | Doctor Account | 7 |
| 2.2 | Registration and creation of user profile | 8 |
| 2.3 | Admin Account | 9 |
| 3 | Design | 12 |
| 4 | Analysis Diagram | 15 |
| 4.1 | Class Diagram | 15 |
| 4.2 | Use Case Diagram | 16 |
| 5 | Implementation Technologies | 17 |
| 6 | Hardware and Software Requirements | 20 |

* 1. **INTRODUCTION TO PROJECT**

The “ClinApp” has been developed to override the problems prevailing in the practicing manual system. This project is supported to eliminate and reduce the hardships faced by the manual work. Moreover, this project is made for clinics to carry out management of their clinic in smooth and effective manner.

The project is reduced as much as possible to avoid errors while entering the data. It also provides message while entering invalid data such as wrong username etc.

No formal knowledge is needed for user to use this system. Thus by this all it proves user-friendly. ClinApp, as described above, can lead to error free, secure, reliable and fast management system. It can assist the user to concentrate on their other activities rather than to concentrate on the record keeping work. Thus it will help clinics in better utilization of services.

Every organization, whether big or small, has challenges to overcome and managing the information of Doctor, Clinic, Timeslot, Appointments, Calendar. Every Clinic Management System has different Clinic needs, therefore we design exclusive

employee management systems that are adapted to your managerial requirements.

This is designed to assist in strategic planning, and will help you ensure that your

organization is equipped with the right level of information and details for your future

goals. Also, for those busy executive who are always on the go, our systems come with

remote access features, which will allow you to manage your workforce anytime, at all

times. These systems will ultimately allow you to better manage resources.

**1.2 Problem Definition:**

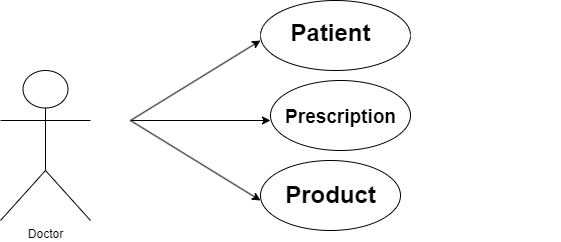
* In our country most of the large hospitals have new technologies but small clinics don’t have the technology to reduce their manual record effort
* Large space usage. Use of paper to record data requires large space for storage over time. This can be seen in many places where analogue systems are in use
* It is manual and time consuming.
* There are more human errors.
* There is difficult to manage the large amount of data in a register or a book.
* Searching becomes more difficult.
* Maximum chances of losing data.
* No security as data can be misplaced or damaged.
* Lastly, it was not offers a user friendly work environment.

**1.3 Purpose and Scope :**

* Working with current system is quite slower, complicated and time consuming task. Operations such as searching, adding, removing, deleting cannot be done efficiently. Hence organization needs new system with enhanced features. Features such as adding patients, clinic, employee, prescription etc.
* It will make easy to save huge data base in computer.
* Unauthorized person cannot access the Database in computer.
* Easily can add details of a patient of clinic update and delete details.
* Easy to search details about patient
* Security and Backup.

**2.System Analysis:**

**2.1 FUNCTIONAL REQUIREMENTS**

****

**2.2 Doctor Account**

First doctor needs to sign in to make account. Doctor has to fill their name, password and roll after successful sign in doctor need to add patient, prescription and product

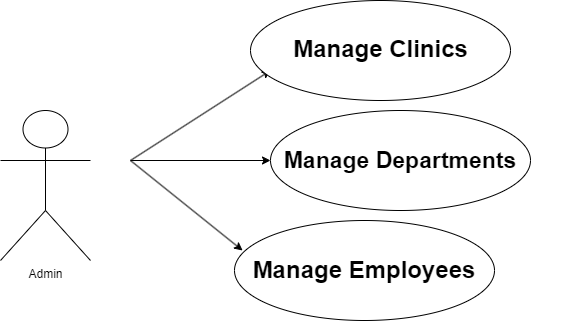
existing doctor can perform add delete update operations. Doctor can view patients’ history and manage prescription.

**2.2 Registration and creation of user profile**

.

****

* 1. **Admin Account**

****

With the help of the login page admin can login with email-id and password. Admin is the one handles the clinic and has all the permissions to see the clinic details as well as doctor, employee, patient details and also the prescription details which are very important part of the clinic.

After successful login admin can perform add, delete and update on all fields. Admin can also view ‘Patient list’ which includes all the Patient who have registered to a clinic.

**3. DESIGN**

**3.1 Database Design**

The following table structures depict the database design.

# Table1: Clinic

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Key Type/ Constraint** | **Column Name** | **Data Type** | **Length** | **Allow Null** |
| PRI | Id | int | 4 | NO |
|  | clinic\_name | varchar | 100 | NO |
|  | clinic\_email | varchar | 100 | YES |
|  | clinic\_address | varchar | 300 | NO |
| UNI | clinic\_phone | varchar | 10 | YES |
|  | morning\_time | varchar | 30 | YES |
|  | afternoon\_time | varchar | 30 | YES |
|  | evening\_time | varchar | 30 | YES |
|  | registration\_date | date | - | YES |
| MUL | user\_username | varchar | 100 | YES |

## **Table2: Department**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Key Type/ Constraint** | **Column Name** | **Data Type** | **Length** | **Allow Null** |
| PRI | id | int | 4 | NO |
|  | dept\_name | varchar | 100 | NO |
| Mul | clinic\_id | int | 4 | YES |

## **Table3: Doctor Qualification**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Key Type/ Constraint** | **Column Name** | **Data Type** | **Length** | **Allow Null** |
| PRI | id | int | 4 | NO |
|  | graduation | varchar | 100 | YES |
|  | graduation\_completion\_year | year | - | YES |
|  | post\_grad | varchar | 100 | YES |
|  | post\_grad\_completion\_year | year | - | YES |
|  | medical\_registration\_no | mediumtext | - | YES |
|  | additional\_qualification | varchar | 100 | YES |
|  | additional\_qualification\_completion\_year | year | - | YES |
|  | experience | tinyint | 1 | YES |
| UNI | employee\_id | int | 4 | YES |

## **Table4: Employee**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Key Type/ Constraint** | **Column Name** | **Data Type** | **Length** | **Allow Null** |
|  | emp\_first\_name | varchar | 50 | YES |
|  | emp\_last\_name | varchar | 50 | YES |
|  | mobile | varchar | 10 | NO |
|  | gender | varchar | 6 | NO |
|  | email | varchar | 100 | YES |
|  | dob | date |  | YES |
|  | address | varchar | 300 | YES |
|  | salary | int | 4 | YES |
|  | hiring\_date | date |  | YES |
|  | designation | varchar | 50 | YES |
| MUL | dept\_id | int | 4 | YES |

## **Table5: Patient**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Key Type/ Constraint** | **Column Name** | **Data Type** | **Length** | **Allow Null** |
| PRI | id | int | 4 | NO |
|  | first\_name | varchar(50) | 50 | NO |
|  | last\_name | varchar(50) | 50 | YES |
|  | age | tinyint | 1 | YES |
|  | mobile | varchar(10) | 10 | YES |
|  | email | varchar(100) | 100 | YES |
|  | address | varchar(300) | 300 | YES |
|  | gender | varchar(6) | 6 | NO |
| MUL | dept\_id | int | 4 | YES |
| MUL | employee\_id | int | 4 | YES |

## **Table6: Prescription**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Key Type/ Constraint** | **Column Name** | **Data Type** | **Length** | **Allow Null** |
| PRI | id | int | 4 | NO |
|  | prescription\_date | date | '1000-01-01' to '9999-12-31' | YES |
|  | diagnosis | longtext | 4,294,967,295 | YES |
|  | weight | tinyint | 1 | YES |
|  | bp | varchar | 10 | YES |
| MUL | patient\_id | int | 4 | YES |

## **Table7: Product**

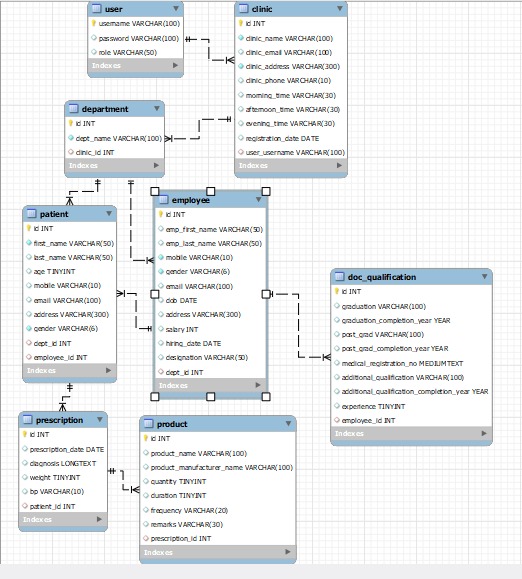
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Key Type/ Constraint** | **Column Name** | **Data Type** | **Length** | **Allow Null** |
| PRI | id | int | 4 | NO |
|  | product\_name | varchar | 100 | YES |
|  | product\_manufacturer\_name | varchar | 100 | YES |
|  | quantity | tinyint | 1 | YES |
|  | duration | tinyint | 1 | YES |
|  | frequency | varchar | 20 | YES |
|  | remarks | varchar | 30 | YES |
| MUL | prescription\_id | int | 4 | YES |

## **Table8: User**

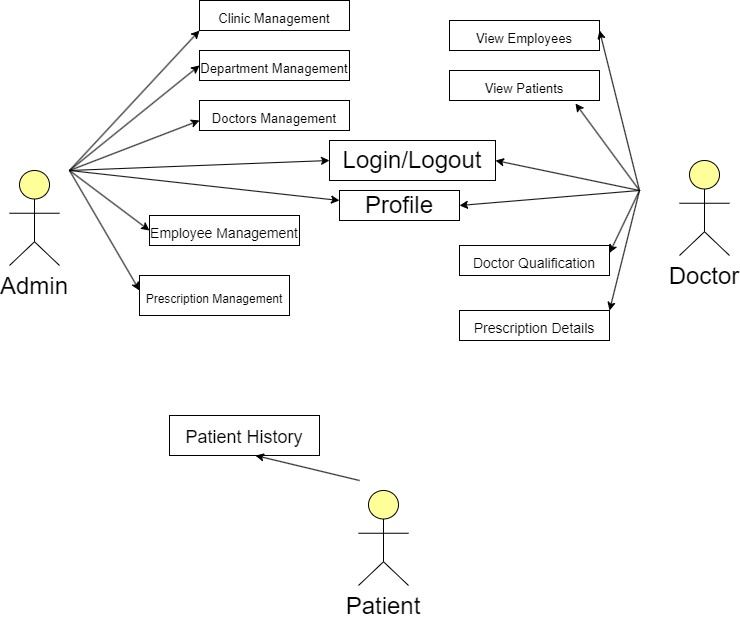
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Key Type/ Constraint** | **Column Name** | **Data Type** | **Length** | **Allow Null** |
| PRI | username | varchar | 100 | NO |
|  | password | varchar | 100 | YES |
|  | role | varchar | 50 | YES |

### 4. Analysis Diagram:

### ER Diagram



**4.1 Use Case Diagram**



**5 Implementation Technologies**

1. **Spring Framework:**

Spring Framework is a Java platform that provides comprehensive infrastructure support for developing Java applications. Spring handles the infrastructure so you can focus on your application.

Spring enables you to build applications from “plain old Java objects” (POJOs) and to apply enterprise services non-invasively to POJOs. This capability applies to the Java SE programming model and to full and partial Java EE.

**2.Spring Advantages:**

1. **Light Weight:** Spring is a lightweight framework because of its POJO implementation. It does not force the programmer to inherit any class and implement any interface. With the help of Spring, we can enable powerful, scalable applications using POJOs (Plain Old Java Object).
2. **Flexible:** It provides flexible libraries trusted by developers all over the world. The developer can choose either XML or Java-based annotations for configuration options. The IoC and DI features provide the foundation for a wide-ranging set of features and functionality. It makes the job simpler.
3. **Loose Coupling:** Spring applications are loosely coupled because of dependency injection. It handles injecting dependent components without a component knowing where they came from.
4. **Powerful Abstraction:** It provides a powerful abstraction to JEE specifications such as JMS, JDBC, JPA, and JTA.
5. **Declarative Support:** It provides declarative support for caching, validation, transaction, and formatting.
6. **Portable:** We can use server-side in web/EJB app, client-side in swing app business logic is completely portable.
7. **Cross-cutting behaviour:** Resource management is a cross-cutting concern, easy to copy and paste everywhere.

**MySQL**

MySQL tutorial provides basic and advanced concepts of MySQL. Our MySQL tutorial is designed for beginners and professionals.

MySQL is an easy to use and simple database system, but in terms of productivity, it is also an efficient database software and is much less complex to set up and administer.

**Client-Server Architecture**

MySQL is based on client-server architecture and can be accessed from anywhere as it can work on distributed networks. Every client machine can communicate to the server using an internet connection. The server processes client requests and returns the desired output back to the client machine

**Compatible and Portable**

The MySQL server is multi-threaded, so many clients can request data at the same time. Each client can use multiple databases simultaneously. MySQL provides several interfaces like command-line interface, GUI interface, or Web browsers that let you enter queries and view the desired results.

**Reliable and Fast**

MySQL stores data efficiently in the database ensuring that it should not contain redundant data. MySQL supports multithreading, it responds to each request with the desired result very quickly by utilizing all the CPU power available.

**Connectivity and Security**

MySQL can be distributed over the network and databases can be accessed from anywhere on the Internet. Only authorized users can access the database by giving proper credentials.

**ReactJS**

ReactJS is a declarative, efficient, and flexible JavaScript library for building reusable UI components. It is an open-source, component-based front-end library which is responsible only for the view layer of the application. It was initially developed and maintained by Facebook and later used in its products like WhatsApp & Instagram

**ReactJS Features**

**Components**

ReactJS is all about components. ReactJS application is made up of multiple components, and each component has its own logic and controls. These components can be reusable which help you to maintain the code when working on larger scale projects.

**One-way Data Binding**

ReactJS is designed in such a manner that follows unidirectional data flow or one-way data binding. The benefits of one-way data binding give you better control throughout the application. If the data flow is in another direction, then it requires additional features. It is because components are supposed to be immutable and the data within them cannot be changed. Flux is a pattern that helps to keep your data unidirectional. This makes the application more flexible that leads to increase efficiency.

**Virtual DOM**

A virtual DOM object is a representation of the original DOM object. It works like a one-way data binding. Whenever any modifications happen in the web application, the entire UI is re-rendered in virtual DOM representation. Then it checks the difference between the previous DOM representation and new DOM. Once it has done, the real DOM will update only the things that have actually changed. This makes the application faster, and there is no wastage of memory.

**Simplicity**

ReactJS uses JSX file which makes the application simple and to code as well as understand. We know that ReactJS is a component-based approach which makes the code reusable as your need. This makes it simple to use and learn.

**5 Hardware and Software Requirements**

**Hardware:**

1. Intel i3 processor 3rd generation or later / AMD Ryzen 200 2nd generation or later

2. 2 GB ddr3 ram.

3. Windows 7 Home edition or later.

4. 200 GB Sata HDD Space

5. Data Connection 200 kbps

**Software:**

1. IntelliJ IDEA Community Edition 2022
2. MySQL 5.7 with Workbench 8.0
3. Google Chrome version 79.0
4. Apache Tomcat Server 8.5
5. Maven Dependencies
6. Visual Studio Code

**Thank You!!!!**