WEB APPLICATION FOR RAMMOHAN CHILDREN'S HOSPITAL

A

Major Project Thesis submitted

in the partial fulfillment of the requirements for the award of the degree of

BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE & ENGINEERING

BY

SOFIA TARANNUM	**19281A0590**
U. MADHUMITHA	**19281A0598**
S. SATWIKA	**19281A0578**
SARDAR HANSRAJ SINGH	**19281A05B5**

PROJECT GUIDE

NEELAKANT DESHPANDE

(Assistant Professor)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

KAMALA INSTITUTE OF TECHNOLOGY & SCIENCE

(Approved by AICTE, New Delhi, Affiliated to JNTU, Hyderabad) Accredited by NAAC with A++ and NBA (CSE,ECE,EEE) Singapur, Huzurabad, Karimnagar,Telangana-505468.

2022 – 2023

Beside Pullela Hospital, Mancherial Chowrasta to Choppadandi Road, Karimnagar - 505001

CERTIFICATE

This is to certify that **SOFIA TARANNUM (19281A0590), U. MADHUMITHA (19281A0598), S. SATWIKA (19281A0578), SARDAR HANSRAJ SINGH (19281A05B5)** students of B. Tech (C.S.E) from kamala institute of technology and science, Singapur, Huzurabad, have continuously interacted with the management in gathering the requirements for the development of the "WEB APPLICATION FOR RAMMOHAN CHILDREN'S HOSPITAL". We are satisfied with the work done by the students, as it has satisfied our requirements. We wish all the success for their future.

Place

Date **Signature of issuing authority**

KAMALA INSTITUTE OF TECHNOLOGY & SCIENCE

(Sponsored by Kamala education society, Approved by AICTE, New Delhi, Affiliated to JNTU, Hyderabad, Accredited by NAAC with A++ and NBA (CSE,ECE,EEE))Singapur, Huzurabad, Karimnagar,Telangana-505468.

Department Of Computer Science & Engineering

CERTIFICATE

This is to certify that **SOFIA TARANNUM (19281A0590), U. MADHUMITHA (19281A0598), S. SATWIKA (19281A0578), SARDAR HANSRAJ SINGH (19281A05B5)** of the IV Year B. Tech Computer Science and Engineering have satisfactorily completed the dissertation work for Project Work entitled **"WEB APPLICATION FOR RAMMOHAN CHILDREN'S HOSPITAL"** towards the partial fulfillment of B.Tech degree in the academic year 2022-2023.

Project Guide **Head of the Department**

Principal **External Examiner**

INDEX

ACKNOWLEDGEMENT

The success of any course depends mainly on the teachers who teach us. Only good teaching can interpret the syllabus and produce desirable changes and competent citizens. This project was a team effort and many people whose names do not appear on the cover also deserve credit. First, we thank God almighty for his manifold mercies in carrying out our project successfully.

We would like to pay our respects to our Principal **Dr. K. SHANKER** and Management for providing all the facilities required for completing this project work.

We sincerely thank our Head of the Department and Professor of Computer Science & Engineering **Dr. Capt. RAVINDRA BABU KALLAM**, for encouraging us in doing this project and for his guidance.

We would like to thank our project co-ordinator **Mr. J. PAVAN KUMAR**, Assistant Professor in CSE for the continuous support in completion of this project.

We would like to thank our project guide **NEELAKANT DESHPANDE,** Assistant Professor in CSE for the valuable guidance and constant encouragement at each stage of this project.

We thank other teaching and other non-teaching staff of Kamala Institute of Technology & Science, for supporting us in every stage of our major project, entitled **"WEB APPLICATION FOR RAMMOHAN CHILDREN'S HOSPITAL".**

We thank our parents and friends for their moral support throughout the project work that helped to strengthen our will.

SOFIA TARANNUM	**19281A0590**
U. MADHUMITHA	**19281A0598**
S. SATWIKA	**19281A0578**
SARDAR HANSRAJ SINGH	**19281A05B5**

ABSTRACT

The main aim of this project is to create an online portal for a hospital that offers its services exclusively to infants, children, adolescents, and young adults. Our website helps the user to register an appointment for check-up, donate blood, get vaccinated and apply for a job. The admin provides the information about the vaccines available and job requirements needed for the hospital.

In the existing system, it was very hard to fetch the details of the in-patient,out-patient. With this project it is very easy for the admin to access the details of the patients admitted, in-patients and out-patients. The admin also provides the job opportunities and the online generated reports and bills. In order to provide security, we provide login credentials to the admin.

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

1.1 ABOUT THE PROJECT

The online portal for the hospital helps the user to book an appointment for check-up, apply for a vaccine to get vaccinated, donate blood and apply to job opportunities provided. The admin will provide the vaccines available and job requirements needed for the hospital. The user can apply to job opportunities provided if he/she wants to.

The admin will generate reports, bills and send them to the respective users. Fetching the details of patients will be easy. The admin will be able to fetch the details of in-patient and out-patient, number of appointments fixed, number of patients admitted etc.

1.2 DRAWBACKS OF EXISTING SYSTEM

- Fetching in-patient and out-patient data was not easy.
- There is no online portal which provides booking an appointment for check-up and vaccines, blood donation, job opportunities at the same place.
- No online reports and bills were generated.
- Users never got the online reports and bills.

1.3 PROPOSED SYSTEM WITH FEATURES

Our project helps the user to get information about vaccines, appointment, blood donation, job opportunities all at one place and an easy access to the admin to fetch data.

FEATURES:

- User can book an appointment for check-up.
- Users can donate blood.
- Users can apply for a vaccine.
- Users can apply for the job opportunities provided.
- The user will get the online generated reports, bills through email.
- Admin can provide the information about vaccines and job opportunities available.
- Admin can generate reports, bills and send them to the respective users.
- Admin can easily fetch the data of in-patient, out-patient, number of appointments fixed, number of patients admitted etc.

CHAPTER 2

LITERATURE SURVEY

A children's hospital is the one that offers its services exclusively to infants, children, adolescents and young adults. In any hospital, the management of the hospital will work for the betterment of its patients. The hospital provides the daily check-ups, reports, bills etc. In addition to these they also have in-patients and out-patients. But as all these are done manually they take a lot of human effort and time. To resolve these issues we have made an interactive and user friendly website for the hospital.

In the present hospital management system there is no online system to access the details of patients, fetch the in-patient and out-patient details. To make an appointment for the check-up, the user needs to come to the hospital and book his/her appointment. To take bills and reports, the user has to come to the hospital then consult the management and take the bills and reports. To know about the vaccine related information, the user needs to come to the hospital to know the details and then get vaccinated. There is no detailed information about the appointments booked, number of patients admitted, number of in-patients and out-patients etc.

In our proposed system, the user can directly book an appointment for check-up. The user need not come to the hospital to book an appointment, he/she can book it from their home itself which saves the user's time and energy. In this, the hospital also provides its additional facilities such as vaccinations and blood donation. The user can book vaccine to get vaccinated and fill the blood donation form to donate blood. The hospital also provides the job opportunities available for the hospital for which the users can apply if they want to. The admin will generate the online reports, bills and send them to the respective users. This saves a lot of time in visiting the hospital and waiting in line to get the reports. The users get a lot of easy access to booking appointments, getting reports and getting bills. The admin will get an easy access to fetch the data of patients. The problems admin faced in getting the in-patient and out-patient details will be overcomed. The admin now can fetch the data of number of patients admitted, number of appointments booked and the number of in-patients and out-patients.

In our project the admin is given the login credentials and the user can book an appointment for check-

up. All the additional facilities provided for the users such as blood donation, vaccines and job opportunities, the user can directly book and apply to them by simply filling an online form.

The main objective of our project was to first research and understand the current hospital system, then to discover their limitations and overcome them. This has enabled complete understanding and convenience in providing easy access to users and admin. Reports and bills are generated within the time and makes the management work easy.

CHAPTER 3

ANALYSIS

The goal of the system analysis is to determine where the problem is and attempt to fix it. This step involves breaking down the system into different pieces to analyse the situation. Analysing project goals and attempts to engage users are carried out so that definite requirements can be defined.

3.1 HARDWARE AND SOFTWARE REQUIREMENTS

Hardware Requirements: The following are the hardware requirements which we have used in our project.

- Processor needed : Intel dual core or above
- RAM : 4 GB or more.
- Hard disk : 320 GB or more.
- Keyboard : Standard Keyboard.
- Mouse : Two or Three Button Mouse.
- System type : 32-bit or 64-bit.

Software Requirements: The following are software requirements. Technologies are specified by the client.

- Operating System : Windows 7 or Higher.
- Scripting Language : Python (Django).
- Front – End : HTML, CSS, JavaScript and Bootstrap.
- Database : MySQL.
- Web Server : XAMPP Server.

3.2 FUNCTIONAL REQUIREMENT AND NON-FUNCTIONAL REQUIREMENTS

Functional Requirements: Functional Requirements are associated with specific functions, tasks or behaviours of the system. The functional requirements address the quality characteristic of functionality while the other quality characteristics are concerned with various kinds of non-functional requirements. A task-based functional requirements statement is a useful skeleton upon which to construct a complete requirements statement.

The following are the functional requirements of our project:

- User booking for check-up.

- Book a vaccine, donate blood, apply for a job.
- Generate reports.
- Generate Bills.
- Users get online reports and bills.

Non-Functional Requirements:

Non-functional requirements are requirements that specify criteria that can be used to judge the operation of a system, rather than specific behaviours. This should be contrasted with functional requirements that specify specific behaviour or functions.

Following are the non-functional requirements:

- **Consistency:** The application provides consistent user interface design to users. The designs of the screen are standardised and consistent that make the user feel comfortable to use it.
- **Convenience:** The application gives convenience to the user to store all the details through the application connected to the database.
- **Availability:** The content must be available to authorised users. This application provides the admin with a login facility to add, modify or view the details.
- **Security:** Administrators can only perform administrative tasks on pages they are privileged to access. Users will not be allowed to access the administrator pages. This site provides password access control to avoid unauthorized users to login.
- **Reliability:** The application provides an effective method to maintain the back-end to store all the details securely. All details are managed by this application effectively.
- **Size:** The performance of the project depends on the size of the project. We put lots of effort in reducing lines of code. In this project the storage space is utilized efficiently.
- **Scalability:** Scalable software can remain stable while adapting to changes, upgrades, overhauls and resource reduction. Scalability is an attribute of a tool or a system to increase its capacity and functionalities based on its user's demand.

3.3 MODULE DESCRIPTION

The modules used in this system are:

- Registration and Appointment
- Facilities

- Patient
- Billing and Reports
- Job Opportunities

1. Registration and Appointment: The user can book an appointment for check-up. The admin is given login credentials. Admin can provide and fetch information by logging into the website.

2.Facilities: Facilities include vaccination and blood donation. The admin will provide the information about the vaccines available. The user can book a vaccine. The user can also donate blood.

3.Patient: The admin will be able to fetch the in-patient and out-patient details. Inpatients are the patients who stay in hospital while under treatment. Outpatients are the patients who attend the hospital for treatment without staying there overnight.

4.Billing and Reports: The admin will generate the online reports, bills and send them to the respective users. The user gets the reports and bills through email.

5.Job Opportunities: The admin will provide the available job opportunities needed for the hospital and the users can apply to them if they want to.

CHAPTER 4

DESIGN

System Design is the process of defining the elements of a system such as architecture, modules and components, the different interfaces of those components and the data that goes through the system. It is meant to satisfy specific needs and requirements of a business or organization through the engineering of a coherent and well-running system.

4.1 BLOCK DIAGRAM:

A block diagram is a diagram of a system in which the principal parts or functions are represented by blocks connected by lines that show the relationships of the blocks. The figure 4.1 shows the block diagram of web application for rammohan children's hospital.

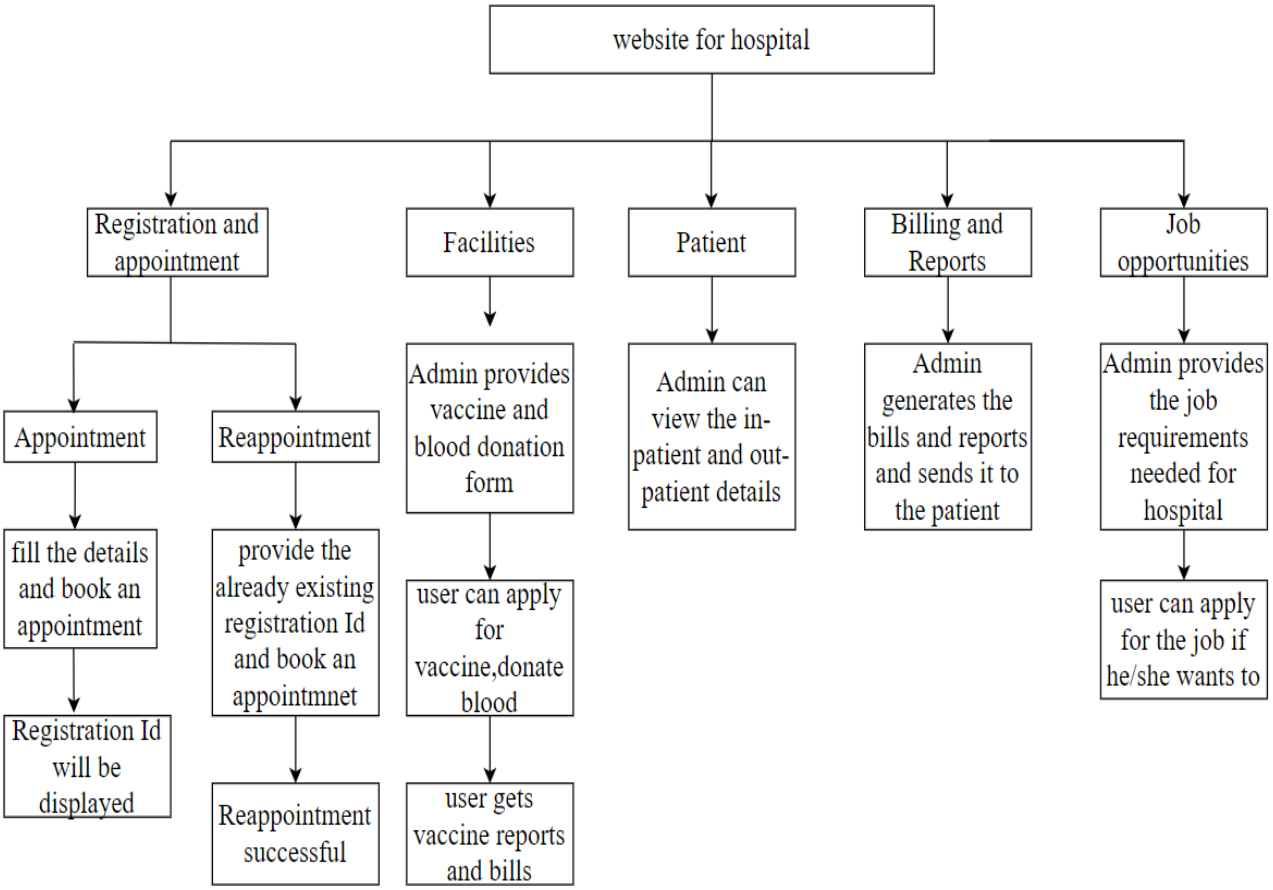


Figure 4.1: Block Diagram

They are heavily used in engineering in hardware design, electronic design, software design, and process flow diagrams. These block components are mainly five and include:

Block: it represents the logical and physical components of the system.

Part: it comprises all aspects modeled using aggregation and association.

Reference: it has all the parts which were developed using aggregation and association.

Standard Port: this is the point of interaction between a system block and the corresponding environment.

Flow Port: this is the point of interaction where a block can emerge from or to.

It is essential to understand the terms used in describing the relationships within Block Diagrams. These are:

Association: it explains the communication amongst the blocks.

Aggregation: this term describes how a unit gets made of parts.

Composition: it is a proper form of aggregation in which the existence of an object which is part of unit relies on the presence of the group.

Generalization: is a leading relationship between blocks in which a designated block contains all the properties of the entire block diagram

4.2 DATA FLOW DIAGRAMS:

A data-flow diagram (DFD) is a graphical representation of the "flow" of data through an information system. DFDs can also be used for the visualization of data processing (structured design). On a DFD, data items flow from an external data source or an internal data store to an internal data store or an external data sink, via an internal process. A DFD provides no information about the timing of the processes, or about whether processes will operate in sequence or in parallel.

It is therefore quite different from a flowchart, which shows the flow of control through an algorithm, allowing a reader to determine what operations will be performed, in what order, and under what circumstances, but not what kinds of data will be input to and output from the system, nor where the data will come from and go to, nor where the data will be stored. It is common practice to draw a context-level data flow diagram first, which shows the interaction between the system and external agents which act as data sources and data sinks. On the context diagram (also known as the Level 0 DFD) the system's interactions with the outside world are modelled purely in terms of data flows across the system boundary.

The context diagram shows the entire system as a single process, and gives no clues as to its internal

organization. This context-level DFD is next "exploded", to produce a Level 1 DFD that shows some of the detail of the system being modelled. The Level 1 DFD shows how the system is divided into subsystems (processes), each of which deals with one or more of the data flows to or from an external agent, and which together provide all of the functionality of the system as a whole.

It also identifies internal data stores that must be present in order for the system to do its job, and shows the flow of data between the various parts of the system. Data-flow diagrams were invented by Larry Constantine, the original developer of structured design, based on Martin and Estrin's "data-flow graph" model of computation. Data-flow diagrams (DFDs) are one of the three essential perspectives of the structured-systems analysis and design method SSADM. The sponsor of a project and the end users will need to be briefed and consulted throughout all stages of a system's evolution. With a data-flow diagram, users are able to visualize how the system will operate, what the system will accomplish, and how the system will be implemented.

The old system's data flow diagrams can be drawn up, compared with the new system's data-flow diagrams to draw comparisons to implement a more efficient system. Data-flow diagrams can be used to provide the end user with a physical idea of where the data they input ultimately has an effect upon the structure of the whole system from order to dispatch to report.

There are different notations to draw data-flow diagrams, defining different visual representations for processes, data stores, data flow, and external entities. Data flow diagrams ("bubble charts") are directed graphs in which the nodes specify processing activities and the arcs specify data items transmitted between processing nodes.

DFD Symbols:

In the DFD, there are four symbols

- A square defines a source (originator) or destination of system data.
- An arrow identifies data flow. It is the pipeline through which the information flows.
- A circle or a bubble represents a process that transforms incoming data flow into outgoing data flows.
- An open rectangle is a data store, data at rest or a temporary repository of data.

Data flow: Data moves in a specific direction from an origin to a destination.

Process: People, procedures or devices that use or produce (Transform) data. The physical component is not identified.

Source: external sources or destination of data, which may be programs, organizations or other entities.

Data Store: Here data is stored or referenced by a process in the system.

In our project, we had to build the data flow diagrams at the very beginning of business process modelling in order to model the functions that our project has to carry out and the interaction between the functions together with focusing on data exchanges between processes.

Constructing a DFD:

Several rules of thumb are used in drawing DFD'S:

- Processes should be named and numbered for an easy interface. Each name should be representative of the process.
- The direction of flow is from top to bottom and from left to right. Data traditionally flows from source to the destination although they may flow back to the source. One way to indicate this is to draw a long flow line back to a source.
- An alternative way is to repeat the source symbol as a destination. Since it is used more than once in the DFD it is marked with a short diagonal.

4.2.1 ZERO LEVEL DATA FLOW DIAGRAM

This is the zero level DFD of the hospital system, where we have elaborated the high level process

of the hospital. It's a basic overview of whole hospital system or process being analyzed or modeled.

The figure 4.2.1 shows the zero level diagram for web application of rammohan children's hospital.

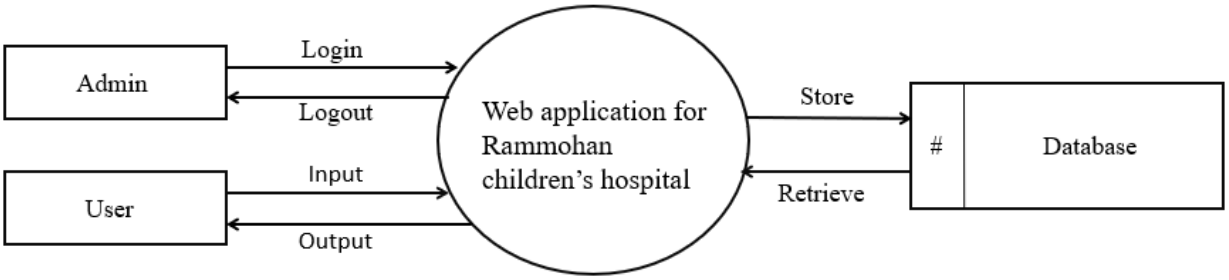


Figure 4.2.1 Zero level data flow diagram

4.2.2 FIRST LEVEL DATA FLOW DIAGRAM

First level dfd of hospital system shows how the system is divided into sub-systems(processes), each of which deals with one or more of the data flows to or from an external agent, and which together provide all of the functionality of the hospital system as a whole.

A data flow diagram is that which can be used to indicate the clear progress of a business venture. In the process of coming up with a data flow diagram, the level one provides an overview of the major functional areas of the undertaking. After presenting the values for most important fields of discussion, it gives room for level two to be drawn.

The figure 4.2.2.1 shows the first level data flow diagram for user and the figure 4.2.2.2 shows the first level data flow diagram for admin.

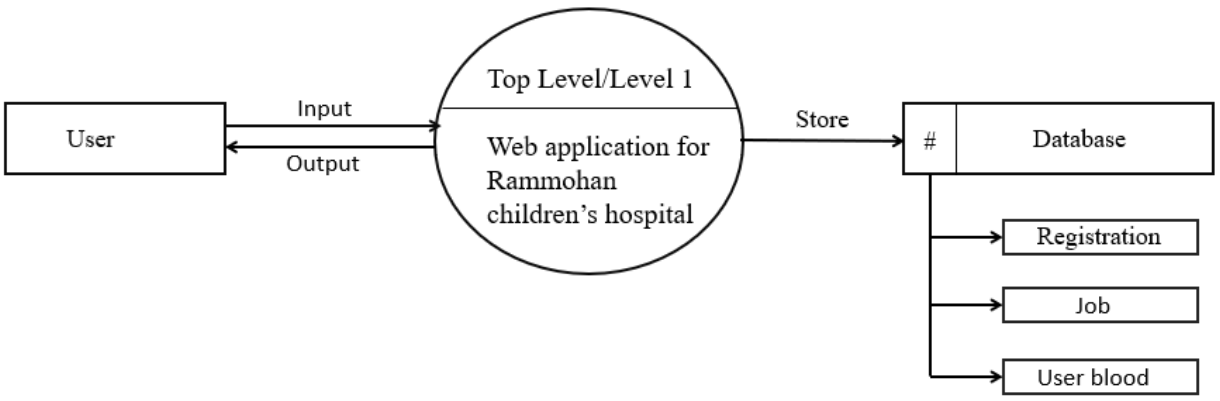


Figure 4.2.2.1 First level Data Flow Diagram for User

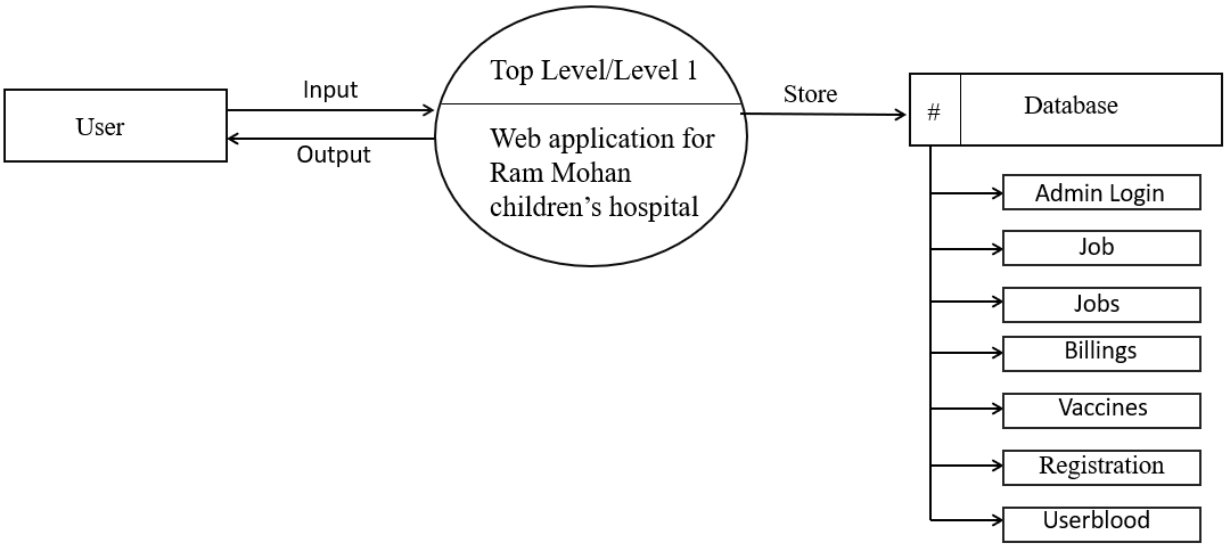


Figure 4.2.2.2 First level Data Flow Diagram for Admin

4.3 ER DIAGRAM:

In software engineering, an entity-relationship model (ER model) is a data model for describing data or information aspects of a business domain or its process requirements. In an abstract way the main components of ER model are entities and the relationships that can exist among them.

Elements in ER diagram

There are three basic elements in an ER Diagram.

- Entity
- Attribute
- Relationship

There are more elements which are based on the main elements. They are weak entities, multi valued attributes, derived attributes, weak relationships and recursive relationships. Cardinality is one of the notations used in ER diagrams.

Entity: An entity can be a person, place, event or object that is relevant to a given system. They are represented by a rectangle and named using nouns.

Weak Entity: A weak entity is an entity that depends on the existence of another entity. It can be

defined as an entity that cannot be identified by its own attributes.

Attribute: An attribute is a property, or characteristic of an entity, relationship, or another attribute.

Multi value Attribute: If an attribute can have more than one value it is called a multi valued Attribute.

Derived Attribute: An attribute derived from another attribute.

Relationship: A relationship describes how entities interact.

Cardinality: Cardinality specifies how many instances of an entity relate to one instance of another entity. Cardinality specifies the maximum number of relationsh

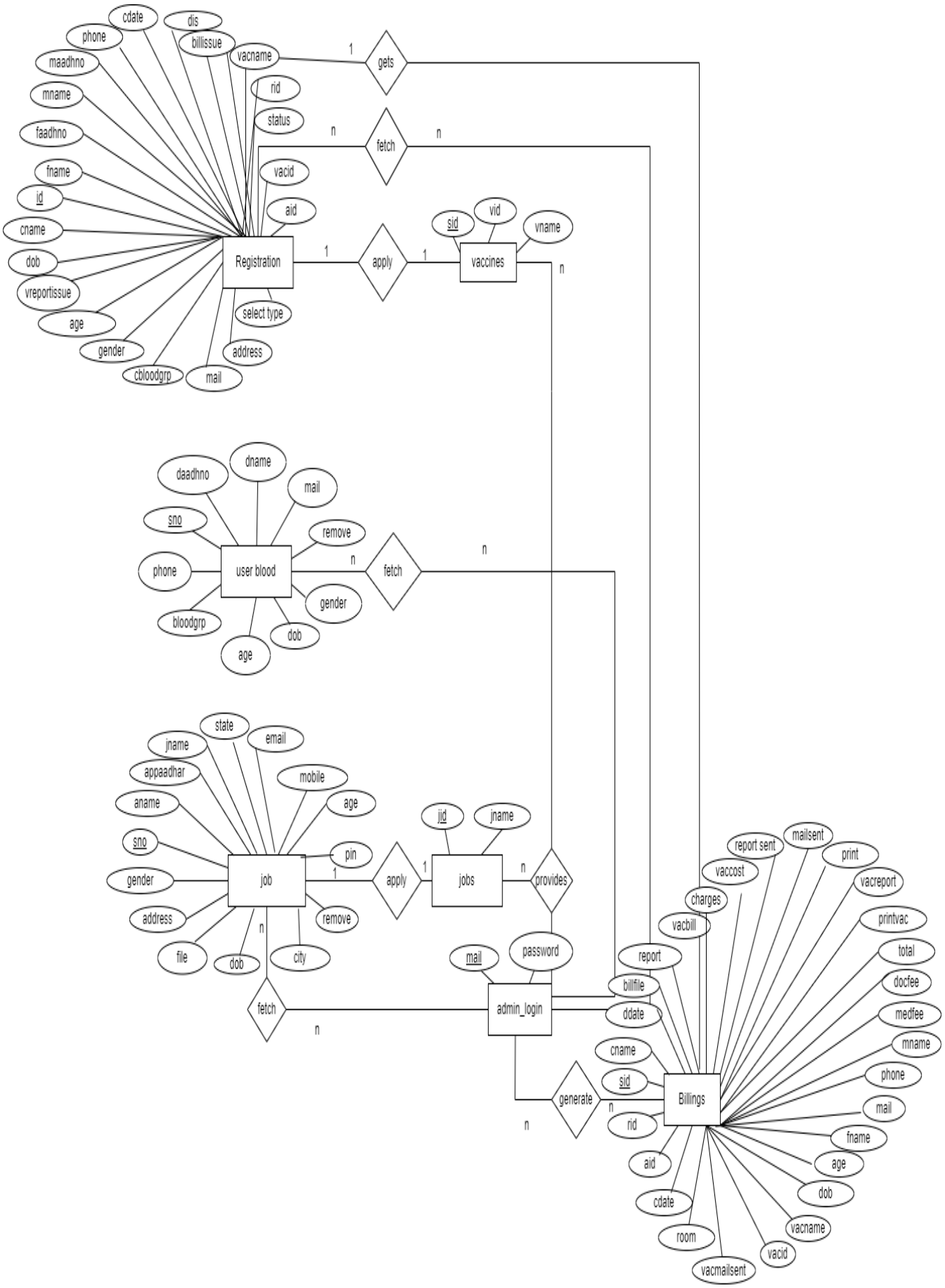


Figure 4.3 Entity Relationship Diagram Entities and their attributes

The figure 4.3 shows ER Diagram for web application for rammohan children's hospital.

Adminlogin: mail(primary key), password

Billings:sid(primary key), rid, aid, cname, cdate, dob, age, fname, mname, phone, mail, docfee, room, medcost, charges, vaccost, total, ddate, billfile, vacbill, report, vacreport, print, printvac, mailsent, reportsent, vacid, vacname, vacmailsent

Job:sno(primary key), aname, appaadhar, email, mobile, gender, dob, age, address, city, pin, state, file, remove

Jobs:jid(primary key), jname

Registration: id (primary key), cname, dob, age, gender, cbloodgrp, fname, faadhno, mname, maadhno, phone, mail, address, selecttype, rid, status, aid, vacid, vacname, cdate, billissue, vreportissue , dis

User blood: sno(primary key), daadhno, dname, mail, phone, bloodgrp, age, dob, gender, remove

Vaccines: sid(primary key), vid, vname

4.4 UML DIAGRAMS:

The Unified Modelling Language (UML) is a standard language for specifying, visualising, constructing and documenting the software system and its components. The UML focuses on the conceptual and physical representation of the system. It captures the decisions and understandings about the systems that must be constructed. Structural models represent the framework for the system and this framework is the place where all other components exist.

So the class diagram, component diagram and deployment diagrams are the part of structural modelling.

They all represent the elements and the mechanism to assemble them. But the structural model never describes the dynamic behaviour of the system. A Behavioural model describes the interaction in the system. It represents the interaction among the structural diagrams. Behavioural modelling shows the dynamic nature of the system. Architectural model represents the overall framework of the system. It contains both structural and behavioural elements of the system. Architectural model can be defined as the blueprint of the entire system. Package diagram comes under architectural modelling. The Unified Modeling Language encompasses a number of models.

- Use Case Diagram
- Class Diagram
- Sequence Diagram
- Collaboration Diagram
- State chart Diagram

- Activity Diagram
- Component Diagram
- Deployment Diagram

4.4.1 Use Case Diagram:

Use case diagrams are central to modelling the behaviour of the system, a sub-system, or a class. Each one shows a set of use cases and actors and relations.

The key points are:

- The main purpose is to show the interaction between the use cases and the actor.
- To represent the system requirement from the user's perspective.
- Use cases are the functions that are to be performed in the module.

The figure 4.4.1.1 shows use case diagram for user and the figure 4.4.1.2 shows the use case diagram for admin.

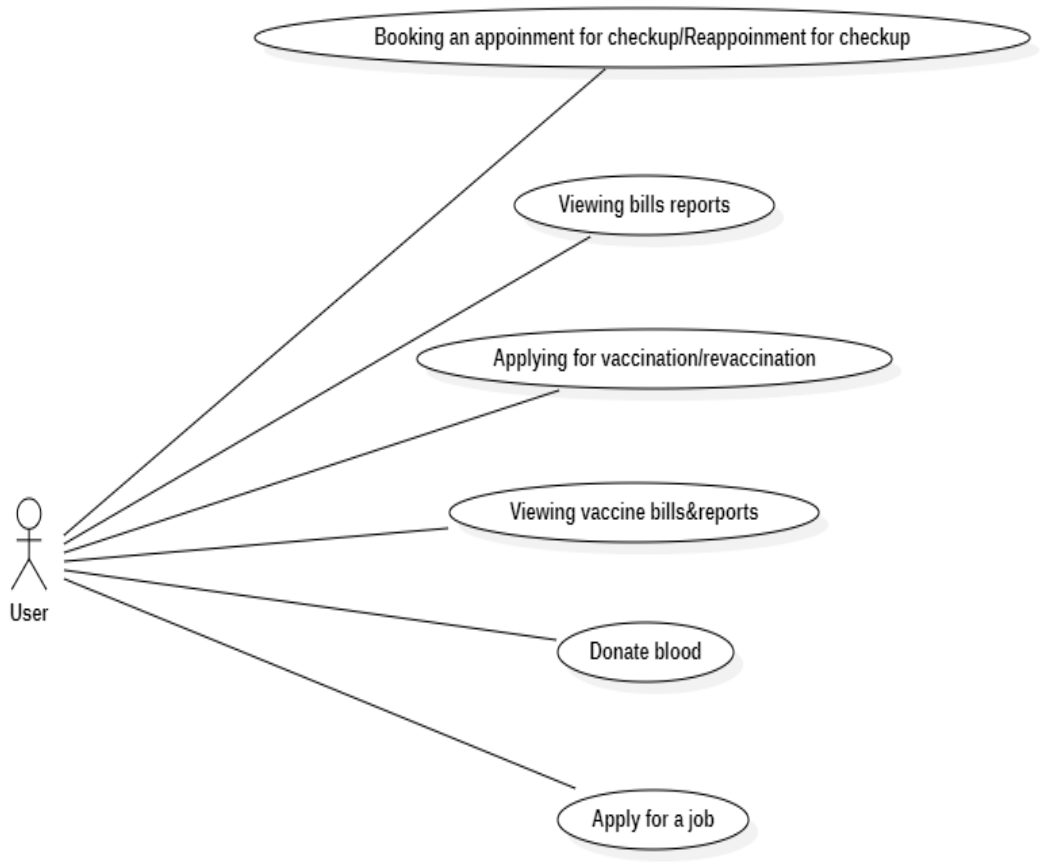


Figure 4.4.1.1 Use Case Diagram for User

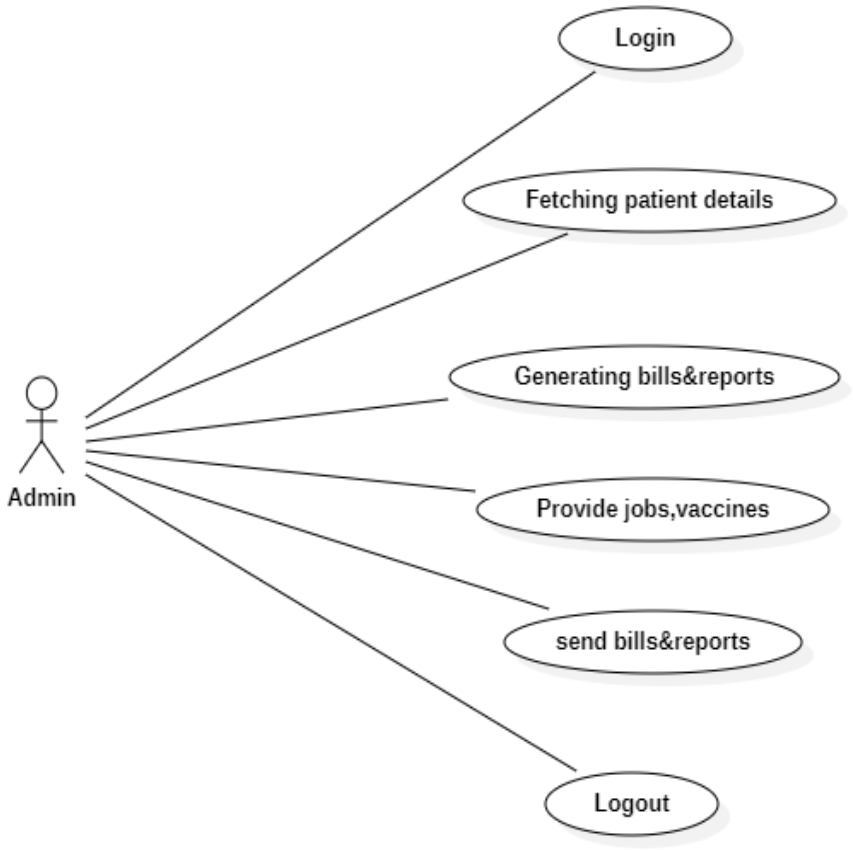


Figure 4.4.1.2 Use Case Diagram for Admin

4.4.2 Class Diagram

A “Class Diagram” shows a set of classes, interfaces and collaborations and their relationships. These diagrams are the most common diagrams in modelling object-oriented systems. The class diagram is a static diagram. It represents the static view of an application. Class diagram is not only used for visualising, describing and documenting different aspects of a system but also for constructing executable code of the software application. The class diagram describes the attributes and operations of a class and also the constraints imposed on the system. The class diagrams are widely used in the modelling of object oriented systems because they are the only UML diagrams which can be mapped directly with object-oriented languages.

The figure 4.4.2 shows the class diagram for web application of rammohan children’s hospital.

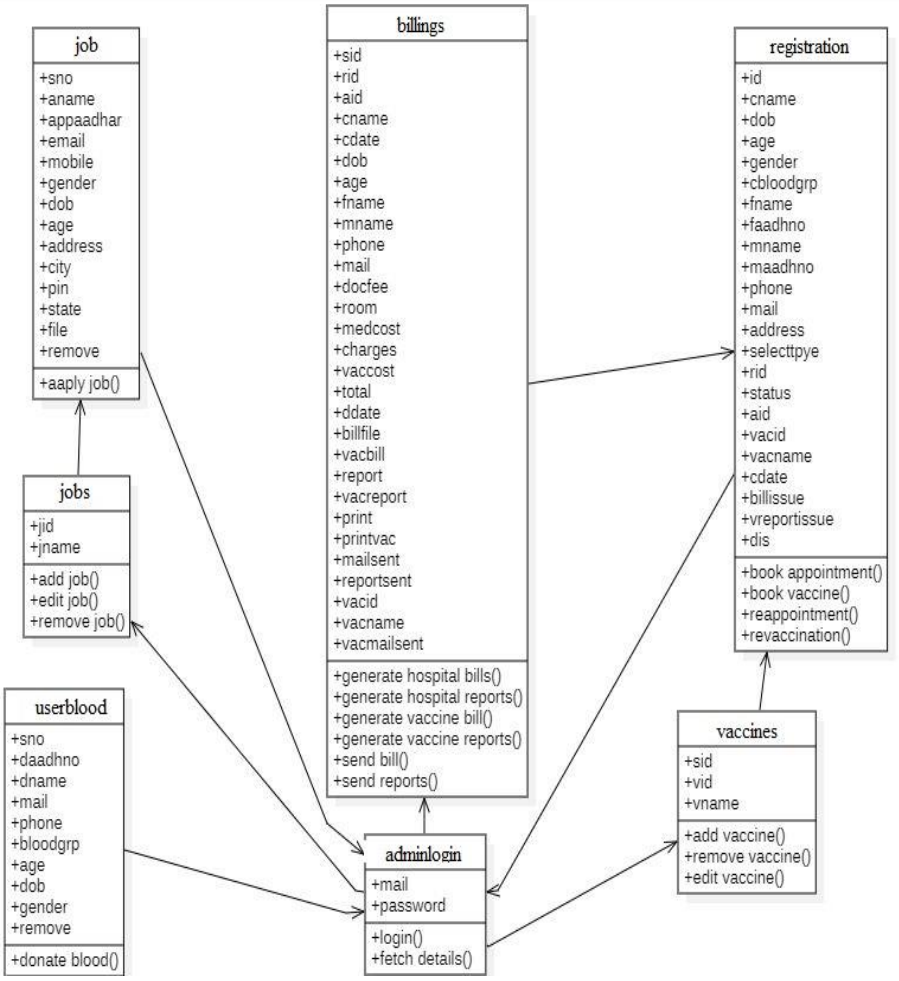


Figure 4.4.2 Class Diagram

4.4.3 Sequence Diagram

Sequence diagram is an interaction diagram which focuses on the time ordering of messages. It shows a set of objects and messages exchanged between these objects. This diagram illustrates the dynamic view of the system.

The key points are:

- The main purpose is to represent the logical flow of data with respect to a process
- A sequence diagram displays the objects and not the classes.

The figure 4.4.3.1 shows the sequence diagram for appointment, figure 4.4.3.2 shows the sequence diagram for vaccination, figure 4.4.3.3 shows the sequence diagram for blood donation and figure 4.4.3.4 shows the sequence diagram for job opportunities.

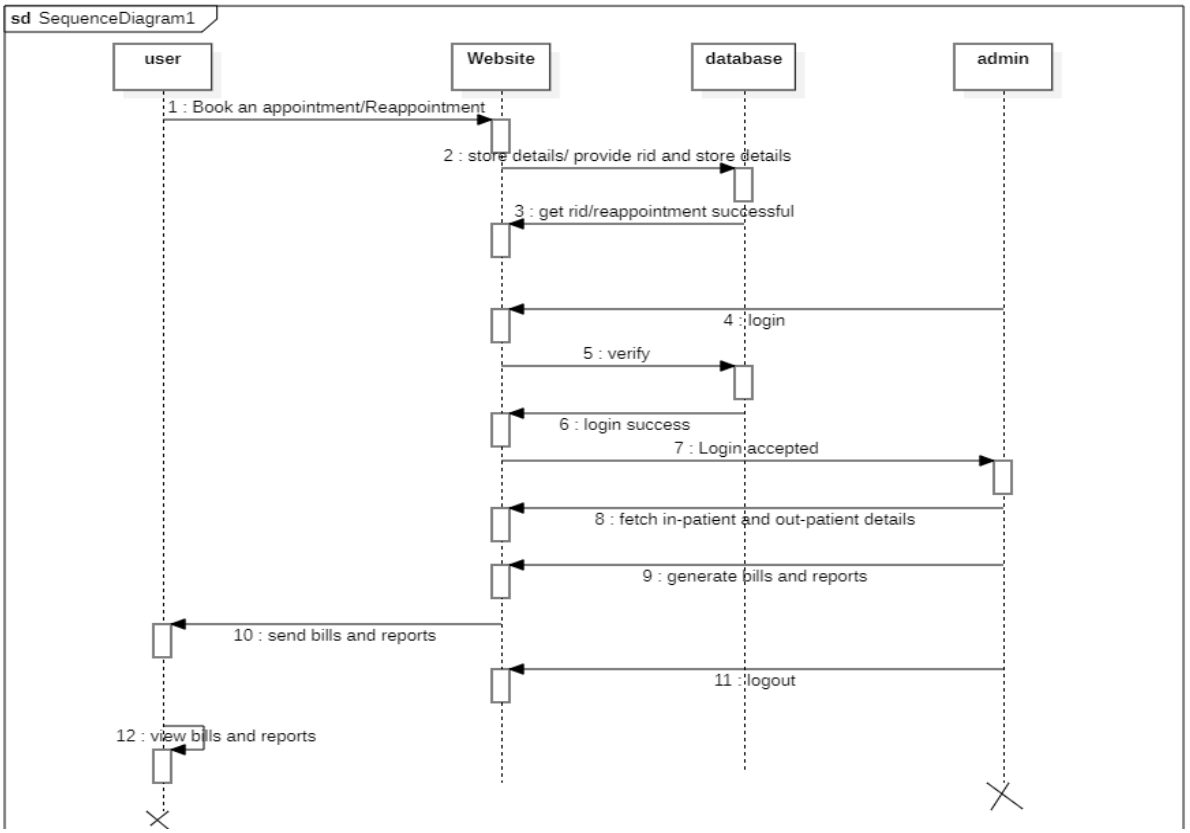


Figure 4.4.3.1 Sequence Diagram for Appointment

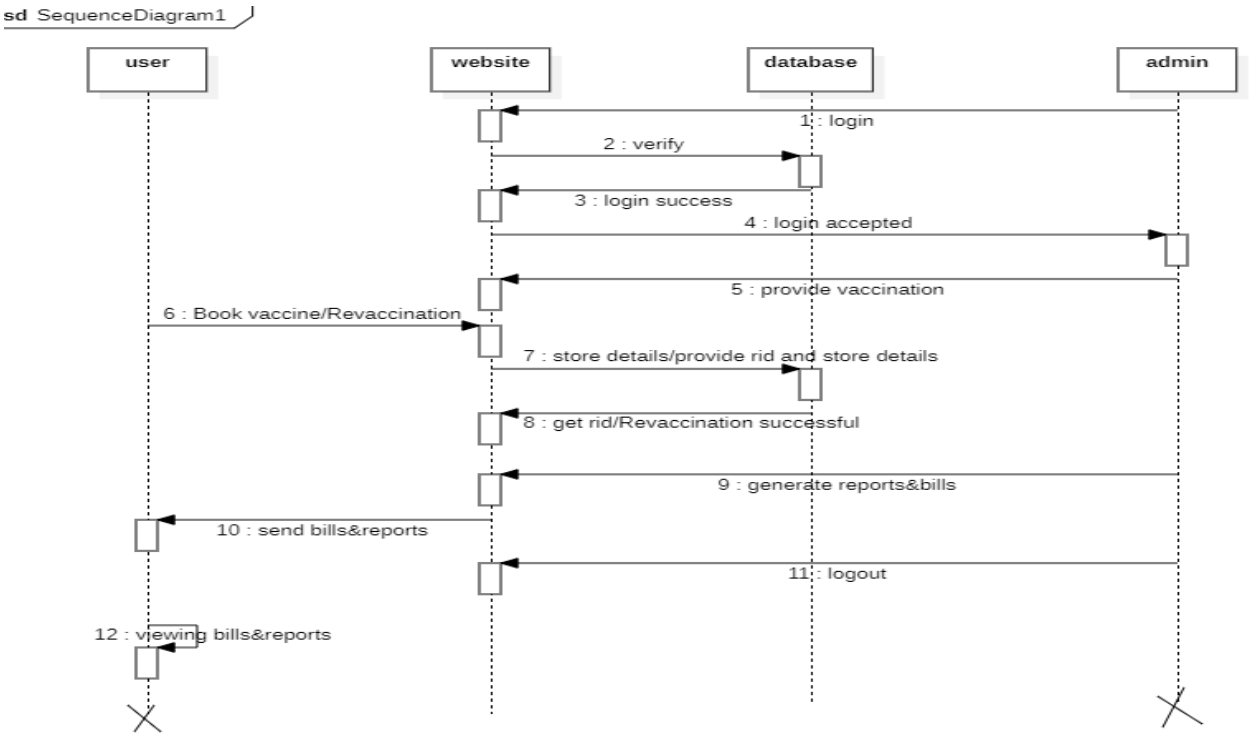


Figure 4.4.3.2 Sequence Diagram for Vaccination

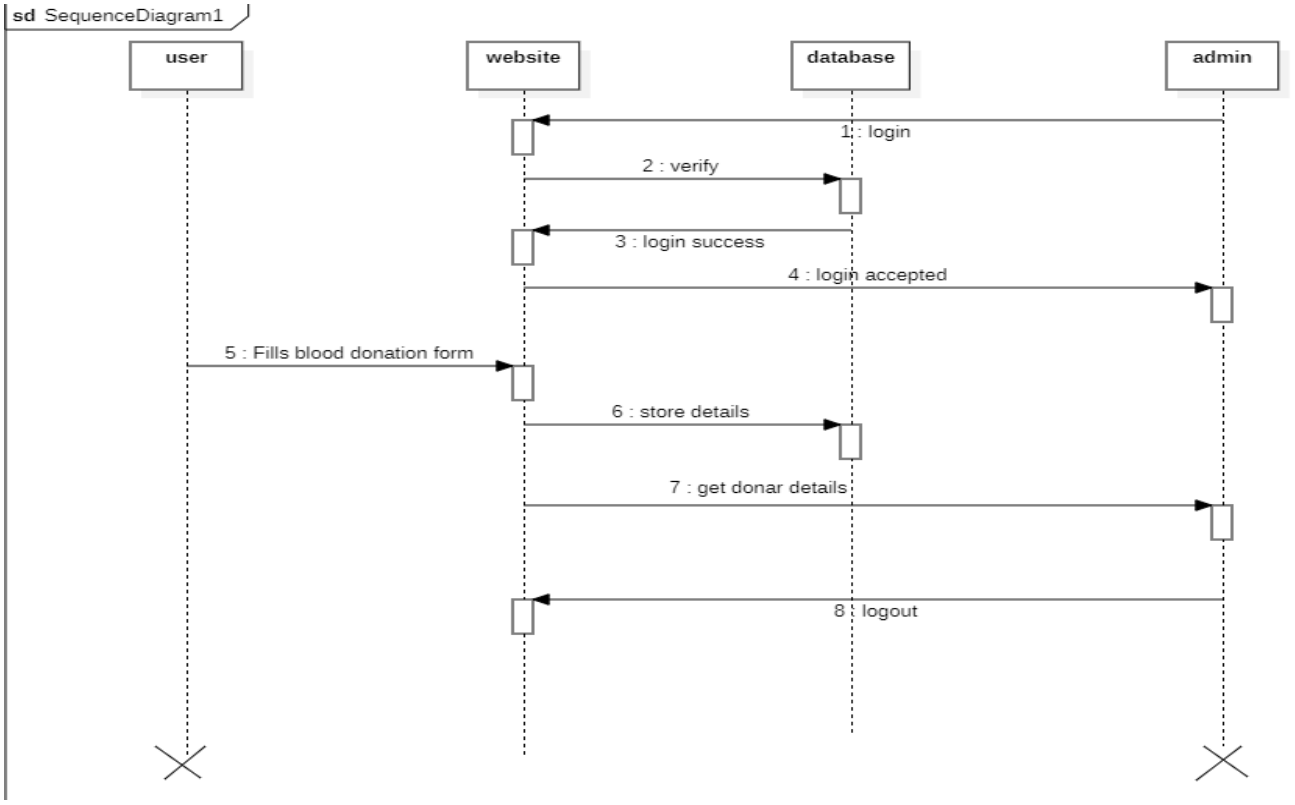


Figure 4.4.3.3 Sequence Diagram for Blood Donation

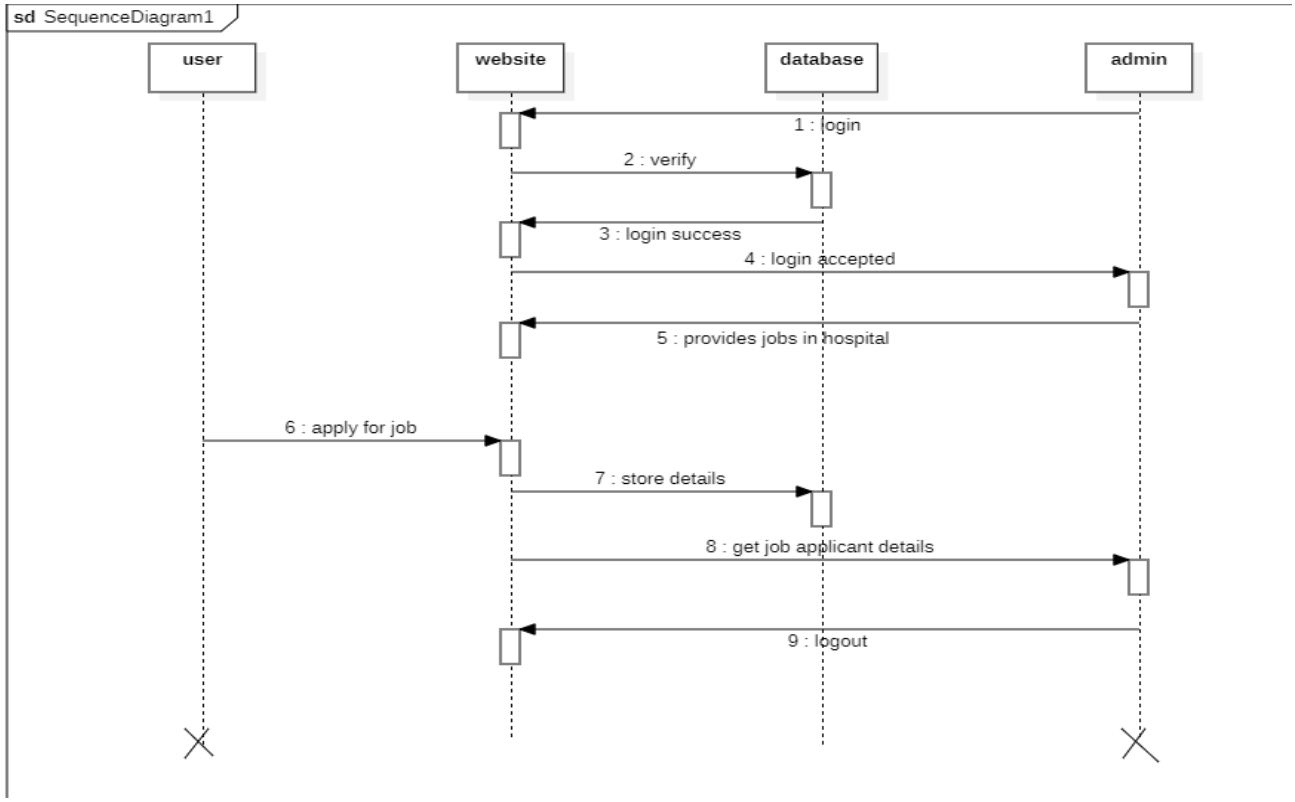


Figure 4.4.3.4 Sequence Diagram for Job Opportunities

4.4.4 Collaboration Diagram

A collaboration diagram is an interaction diagram that emphasises the structural organisation of the objects that send and receive messages. It is also called a communication diagram. Collaboration diagrams are isomorphic, meaning that you can take one and transform it into the other. Collaboration diagrams convey the same information as sequence diagrams, but they focus on object roles instead of times that messages are sent. In the sequence diagram, object roles are the vertices and messages are the connecting links. The object-role rectangles are labelled with either class or object names (or both). The figure 4.4.4.1 shows the collaboration diagram for appointment, figure 4.4.4.2 shows the collaboration diagram for vaccination, figure 4.4.4.3 shows the collaboration diagram for blood donation and figure 4.4.4.4 shows the collaboration diagram for job opportunities.

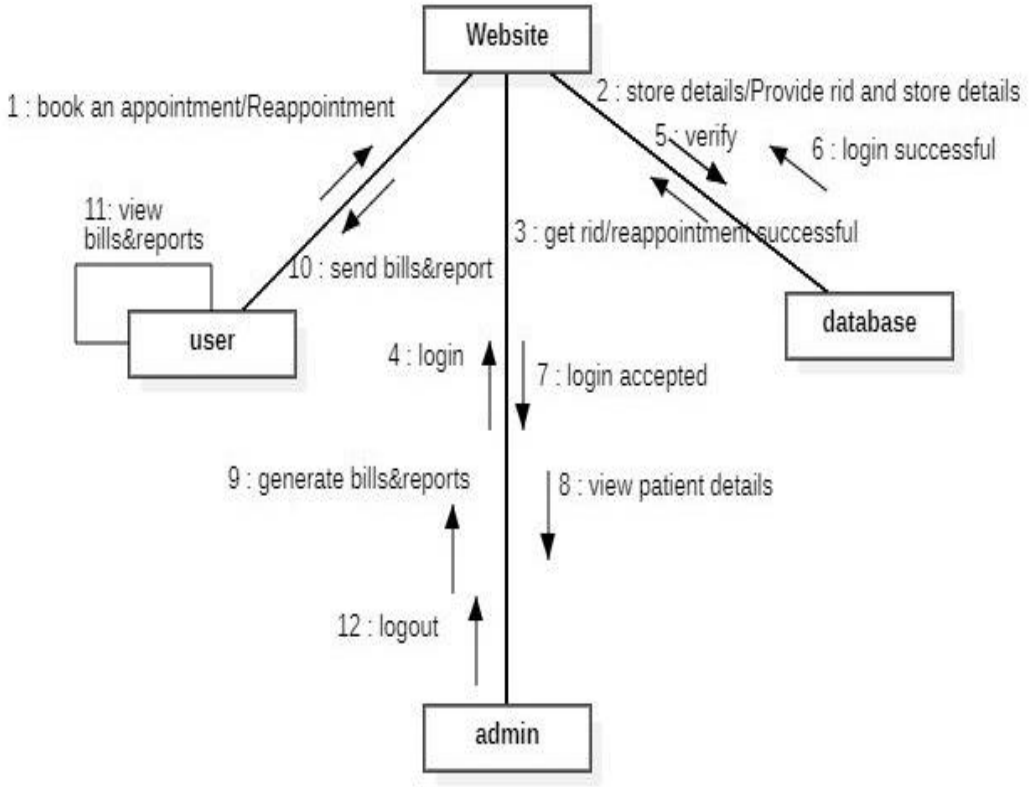


Figure 4.4.4.1 Collaboration Diagram for Appointment

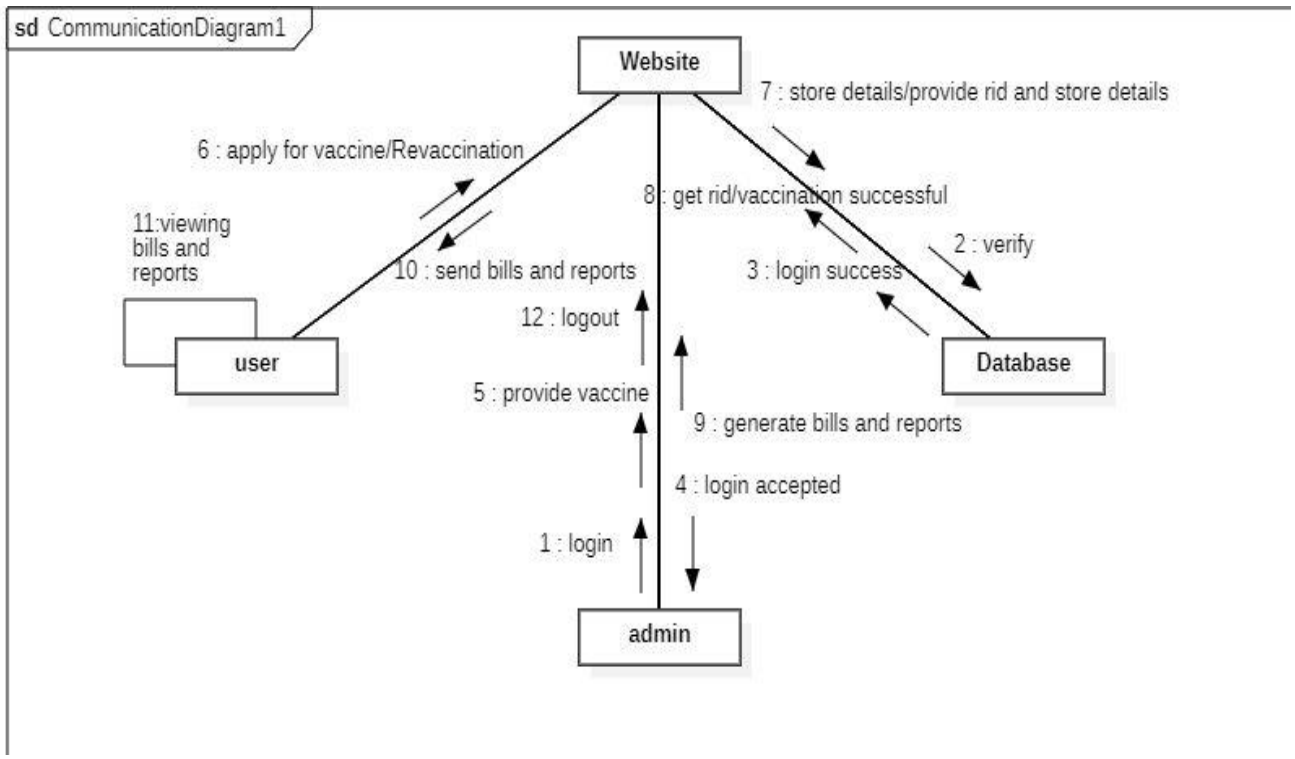


Figure 4.4.4.2 Collaboration Diagram for Vaccination

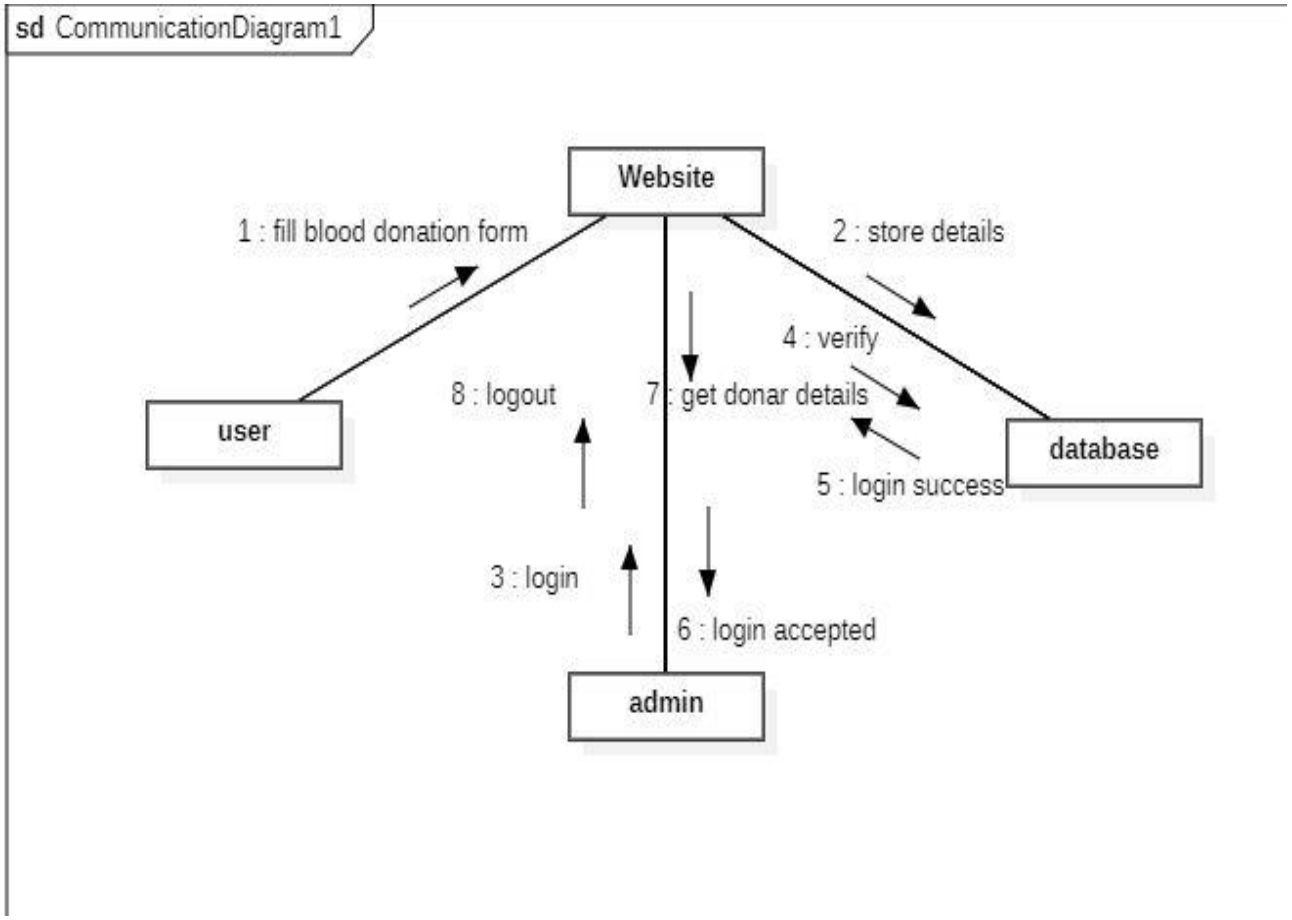


Figure 4.4.4.3 Collaboration Diagram for Blood Donation

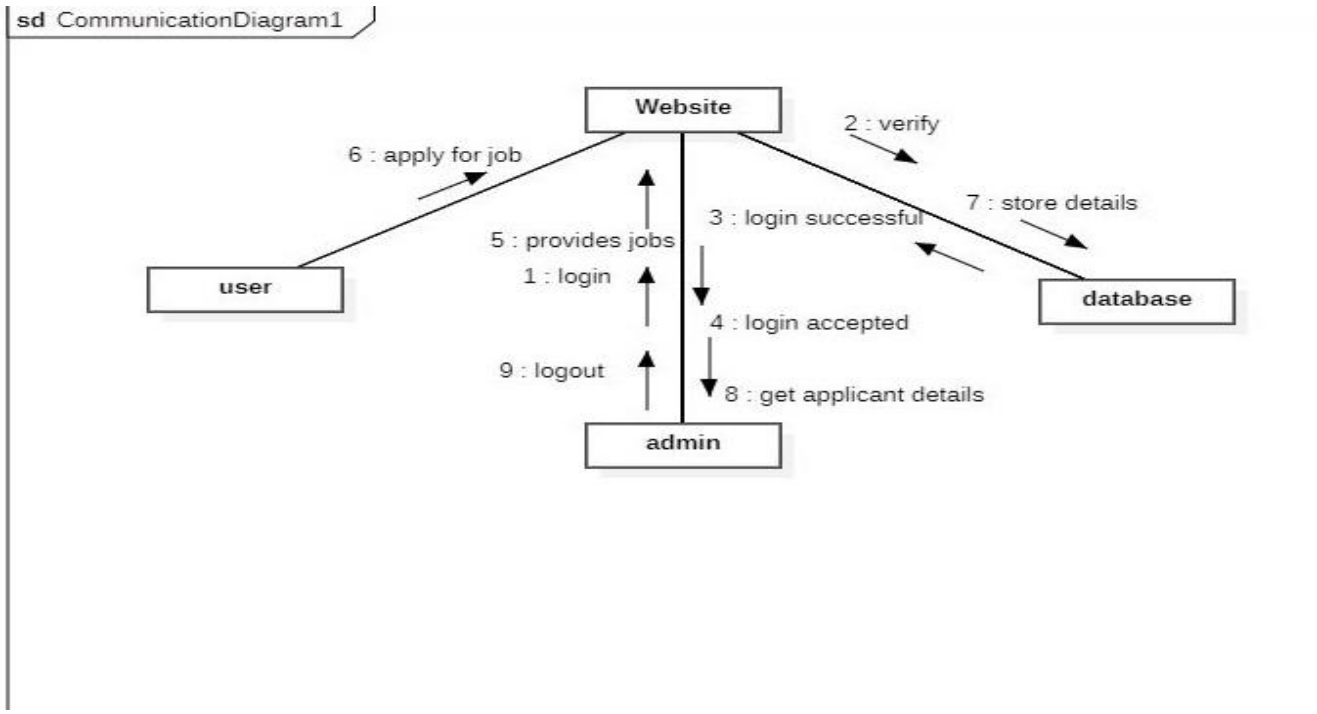


Figure 4.4.4.4 Collaboration Diagram for Job Opportunities

4.4.5 Activity Diagram:

An Activity diagram shows the flow from activity to activity within a system; it emphasises the flow of control among objects. The figure 4.4.5.1 and figure 4.4.5.1 shows the activity diagram for user and admin.

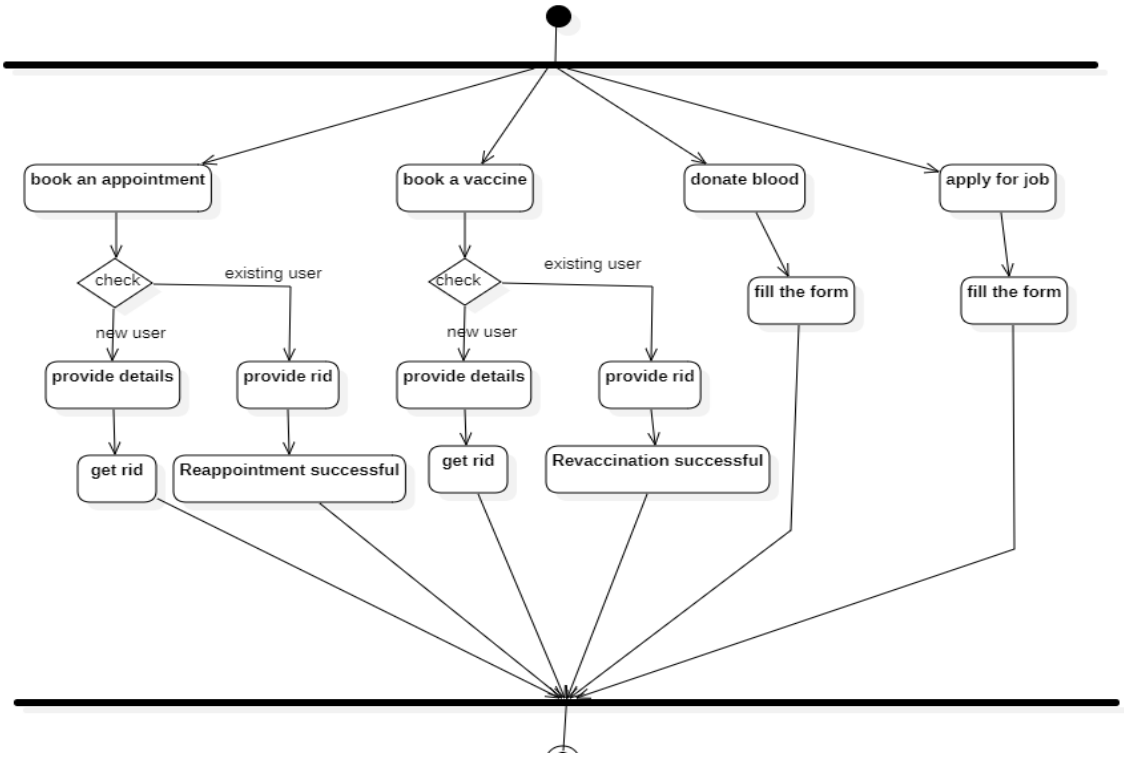


Figure 4.4.5.1 Activity Diagram for User

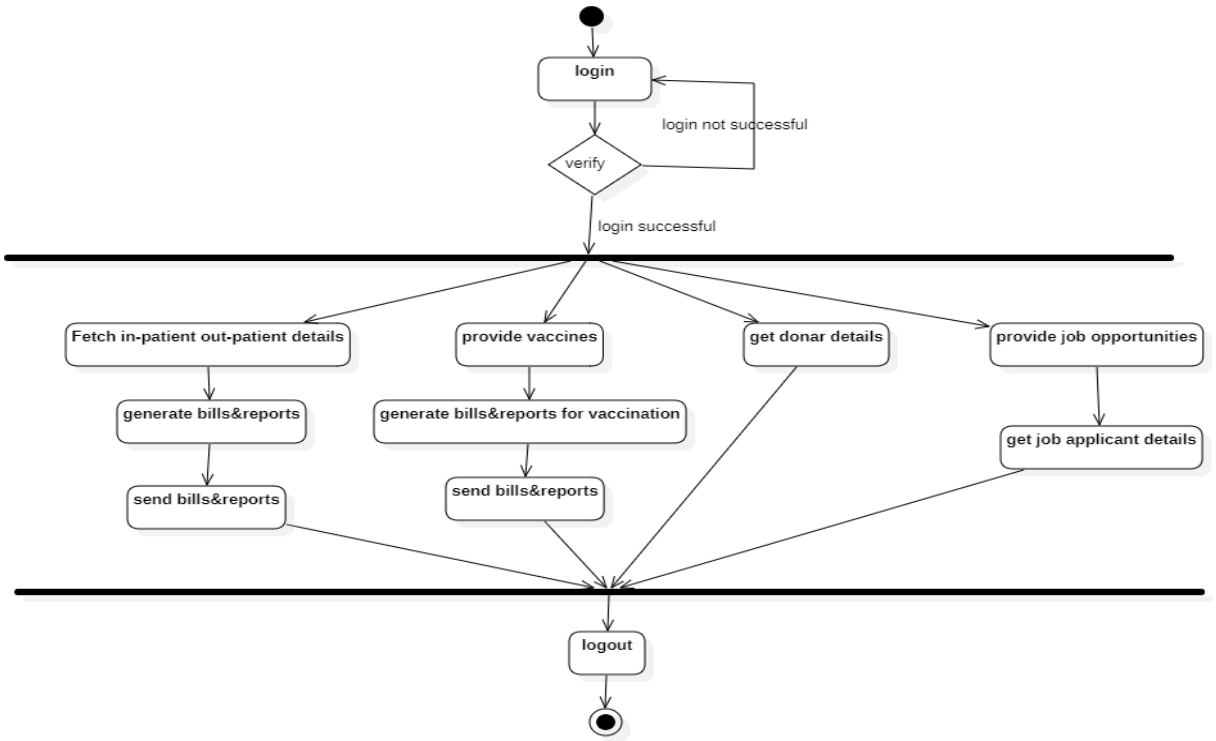


Figure 4.4.5.2 Activity Diagram for Admin

4.5 DATA DICTIONARY:

A Data Dictionary is a collection of names, definitions, and attributes about data elements that are being used or captured in a database, information system, or part of a research project. It describes the meanings and purposes of data elements within the context of a project, and provides guidance on interpretation, accepted meanings and representation. A Data Dictionary also provides metadata about data elements. The meta data included in a Data Dictionary can assist in defining the scope and characteristics of data elements, as well the rules for their usage and application. The design of tables which store details of Admin, User Registration, User vaccine, User blood, User Job, Report, Bills, Jobs, Vaccines, Appointment, Patient.

Admin:

Table 4.5.1 Database for admin

	#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
☐	1	**mail** 🔑	varchar(30)	utf8mb4_general_ci		No	*None*			Change Drop More
☐	2	**password**	varchar(20)	utf8mb4_general_ci		No	*None*			Change Drop More

User Registration:

Table 4.5.2 Database for User Registration

	#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
☐	1	**id**	int(10)			No	*None*		AUTO_INCREMENT	Change Drop More
☐	2	**cname**	char(20)	utf8mb4_general_ci		No	*None*			Change Drop More
☐	3	**dob**	text	utf8mb4_general_ci		No	*None*			Change Drop More
☐	4	**age**	int(2)			No	*None*			Change Drop More
☐	5	**gender**	char(7)	utf8mb4_general_ci		No	*None*			Change Drop More
☐	6	**cbloodgrp**	varchar(10)	utf8mb4_general_ci		No	*None*			Change Drop More
☐	7	**fname**	char(25)	utf8mb4_general_ci		No	*None*			Change Drop More
☐	8	**faadhno**	bigint(12)			No	*None*			Change Drop More
☐	9	**mname**	char(25)	utf8mb4_general_ci		No	*None*			Change Drop More
☐	10	**maadhno**	bigint(12)			No	*None*			Change Drop More

Vaccine:

Table 4.5.3 Database for Vaccine

	#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
☐	1	**sid**	int(9)			No	*None*		AUTO_INCREMENT	Change Drop More
☐	2	**vid**	varchar(3)	utf8mb4_general_ci		No	*None*			Change Drop More
☐	3	**vname**	char(15)	utf8mb4_general_ci		No	*None*			Change Drop More

User blood:

Table 4.5.4 Database for User blood

	#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
☐	1	**sno**	int(10)			No	*None*		AUTO_INCREMENT	Change Drop More
☐	2	**daadhno**	bigint(12)			No	*None*			Change Drop More
☐	3	**dname**	char(25)	utf8mb4_general_ci		No	*None*			Change Drop More
☐	4	**mail**	varchar(11)	utf8mb4_general_ci		No	*None*			Change Drop More
☐	5	**phone**	int(10)			No	*None*			Change Drop More
☐	6	**bloodgrp**	varchar(2)	utf8mb4_general_ci		No	*None*			Change Drop More
☐	7	**age**	int(2)			No	*None*			Change Drop More
☐	8	**dob**	text	utf8mb4_general_ci		No	*None*			Change Drop More
☐	9	**gender**	char(7)	utf8mb4_general_ci		No	*None*			Change Drop More
☐	10	**remove**	int(1)			No	*None*			Change Drop More

User job:

Table 4.5.5 Database for User Job

	#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
☐	1	**sno** 🔑	int(10)			No	*None*		AUTO_INCREMENT	Change Drop More
☐	2	**aname**	char(25)	utf8mb4_general_ci		No	*None*			Change Drop More
☐	3	**appaadhar**	int(12)			No	*None*			Change Drop More
☐	4	**email**	varchar(20)	utf8mb4_general_ci		No	*None*			Change Drop More
☐	5	**mobile**	int(10)			No	*None*			Change Drop More
☐	6	**gender**	char(8)	utf8mb4_general_ci		No	*None*			Change Drop More
☐	7	**dob**	text	utf8mb4_general_ci		No	*None*			Change Drop More
☐	8	**age**	int(2)			No	*None*			Change Drop More
☐	9	**address**	varchar(30)	utf8mb4_general_ci		No	*None*			Change Drop More
☐	10	**city**	char(10)	utf8mb4_general_ci		No	*None*			Change Drop More

Jobs:

Table 4.5.8 Database for jobs

	#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
☐	1	**jid** 🔑	int(10)			No	*None*		AUTO_INCREMENT	Change Drop More
☐	2	**jname**	varchar(10)	utf8mb4_general_ci		No	*None*			Change Drop More

Bills:

Table 4.5.7 Database for bills

	#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
☐	1	**sid** 🔑	int(10)			No	*None*		AUTO_INCREMENT	Change Drop More
☐	2	**rid**	int(10)			No	*None*			Change Drop More
☐	3	**aid**	int(10)			No	*None*			Change Drop More
☐	4	**cname**	char(25)	utf8mb4_general_ci		No	*None*			Change Drop More
☐	5	**cdate**	text	utf8mb4_general_ci		No	*None*			Change Drop More
☐	6	**dob**	text	utf8mb4_general_ci		No	*None*			Change Drop More
☐	7	**age**	int(2)			No	*None*			Change Drop More
☐	8	**fname**	char(25)	utf8mb4_general_ci		No	*None*			Change Drop More
☐	9	**mname**	char(25)	utf8mb4_general_ci		No	*None*			Change Drop More
☐	10	**phone**	int(10)			No	*None*			Change Drop More

CHAPTER-5

IMPLEMENTATION

5.1 TECHNOLOGIES USED:

5.1.1 : PYTHON

Python is a popular high-level programming language that was first released in 1991by Guido van Rossum. It is a versatile language that can be used for a variety of purposes, including web development, data analysis, scientific computing, artificial intelligence, and more. Python is known for its clean syntax, readability, and ease of use, which makes it a popular choice among beginners and experienced developers alike. It also has a large and active community of developers who contribute to its development and maintenance. Python supports multiple programming paradigms, including object-oriented, functional, and procedural programming. It also has a vast standard library that provides a wide range of modules for various purposes, such as web development, data analysis, and networking. One of the main reasons why Python is so popular is its flexibility and versatility. It is widely used in many industries and has become the language of choice for many data scientists and machine learning engineers. It is also widely usedin education and is a great language for beginners to start learning programming.

Features of Python

- Object-oriented
- Large standard library
- High-level language
- Third-party libraries

Object-oriented

Python supports object-oriented programming, which makes it easy to create reusable and modular code. It allows developers to define classes and objects, encapsulate data and functions, and organize code into logical units.

Large standard library

Python comes with a comprehensive standard library that provides developers with a widerange of modules and functions to accomplish common programming tasks.

High-level language

Python is a high-level language, which means that it abstracts away low-level details suchas memory management and hardware dependencies. This feature makes it easier to write code, as developers don't have to worry about these details.

Third party libraries

Python has a vast ecosystem of third-party libraries and frameworks that can be used tobuild complex applications and perform specialized tasks. This feature allows developers to leverage existing code and focus on their specific requirements.

5.1.2 : DJANGO FRAMEWORK

Django is a high-level, open-source web framework that is built on top of Python. It was firstreleased in 2005 and is maintained by the Django Software Foundation.
Django provides developers with a set of tools and libraries that simplify the process of building web applications.

Here are some key features of Django:

1. Object-Relational Mapping (ORM): Django includes a powerful ORM that allows developers to map database tables to Python objects. This makes it easy to interact with databases using Python code.
2. Admin interface: Django provides a built-in admin interface that allows developers to manage the content of their web applications. The admin interface is customizable and provides a wide range of features such as filtering, search, and sorting.
3. URL routing: Django provides a powerful URL routing system that allows developers to map URLs to views. This makes it easy to create clean and user-friendly URLs for web applications.
4. Template engine: Django includes a template engine that allows developers to create dynamic HTML pages. The template engine supports features such as template inheritance, filters, and tags.
5. Security: Django provides a set of tools and libraries that help developers build secure web applications.

It includes features such as cross-site request forgery protection, password hashing, and user authentication.

6. Scalability: Django is designed to be scalable and can handle high-traffic web applications. It includes features such as caching, database optimization, and load balancing.
7. Third-party packages: Django has a vast ecosystem of third-party packages that can be used to add functionality to web applications. These packages include libraries for tasks such as authentication, content management, and social media integration.

   Overall, Django is a popular choice for building web applications due to its robust set of features, ease ofuse, and scalability. It is widely used in industries such as e-commerce, media, and education, and has a large and active community of developers.

5.1.3: HTML

HTML is a hypertext mark-up language that is in reality a backbone of any website. Every website can't be structured without the knowledge of HTML. If we make our web page only with the help of HTML, then we can't add many of the effective features in a web page, for making a web page more effectivewe use various platforms such as CSS. So here we are using this language to make our web pages more effective as well as efficient. And to make our web pages dynamic we are using JavaScript.

5.1.4: CSS

CSS Stands for "Cascading Style Sheet." Cascading style sheets are used to format the layoutof Web pages. They can be used to define text styles, table sizes, and other aspects of Web pages that 35 previously could only be defined in a page's HTML. The basic purpose of CSS is to separate the content of a web document (written in any mark-up language) from its presentation (that is written using Cascading Style Sheets CSS gives the option of selecting various style schemes and rules according to the requirements and italso allows the same HTML document to be presented in more than one varying style.

5.1.5: JAVASCRIPT

JavaScript is considered to be one of the most famous scripting languages of all time. JavaScript, by definition, is a Scripting Language of the World Wide Web. The main usage of JavaScript is toadd various Web functionalities, Web form validations, browser detections, creation of cookies, and so on.JavaScript is one of the most popular scripting languages and that is why it is supported by almost all webbrowsers available today like Firefox.We used the browser Opera or Internet Explorer. JavaScript is considered to be one ofthe most powerful scripting languages in use today. It is often used for the development of client-

side web development. JavaScript is used to make web pages more interactive and dynamic. JavaScript is a lightweightprogramming language and it is embedded directly into the HTML code. JavaScript, as the name suggests, was influenced by many languages, especially Java.

The advantages of JavaScript: -

- An interpreted language.
- Embedded within HTML.
- Embedded within HTML.
- Performance sign for simple, small programs.

5.1.6: MYSQL

MYSQL is a fast, easy-to-use RDBMS being used for many small and big businesses. MYSQL is developed, marketed, and supported by MYSQL, which is a Swedish company. MYSQL isbecoming so popular because of many good reasons.

- MYSQL is released under an open-source license. So, you have nothing to pay to use it.
- MYSQL is avery powerful program in its own right. It handles a large subset of the functionality of the most expensive and powerful database packages.
- MYSQL uses a standard form of the well-known SQL data language.
- MYSQL works on many operating systems and with many languages including Python, PERL, C, C++, JAVA, etc.
- MYSQL works very quickly and works well even with large

5.1.7: BOOTSTRAP

Bootstrap is a free and open-source tool collection for creating responsive websites and webapplications. It is the most popular HTML, CSS, and JavaScript framework for developing responsive, mobile-first websites. Nowadays, the websites are perfect for all the browsers (IE, Firefox, and Chrome) andfor all sizes of screens (Desktop, Tablets, Phablets, and Phones.

- It is Faster and Easier way for Web-Development.
- It creates Platform-independent web-pages.
- It creates Responsive Web-pages.
- It designs the responsive web pages for mobile devices too.
- It is Free and open-source framework.

5.2 FRONTEND DESIGN:

Homepage:

The homepage of the application (figure 5.2.1) is common to all the system users. This interface is available through the web application at the time of login. This page shows the navigating categories. There are four categories as shown in figure-5.2.1:

- Home
  - Make an appointment
- Our Doctors
- Our Facilities
- Registration
- Contact
- Admin Dashboard

Home:

This is our home page. It includes Make an Appointment, Donate blood, Apply for a job.

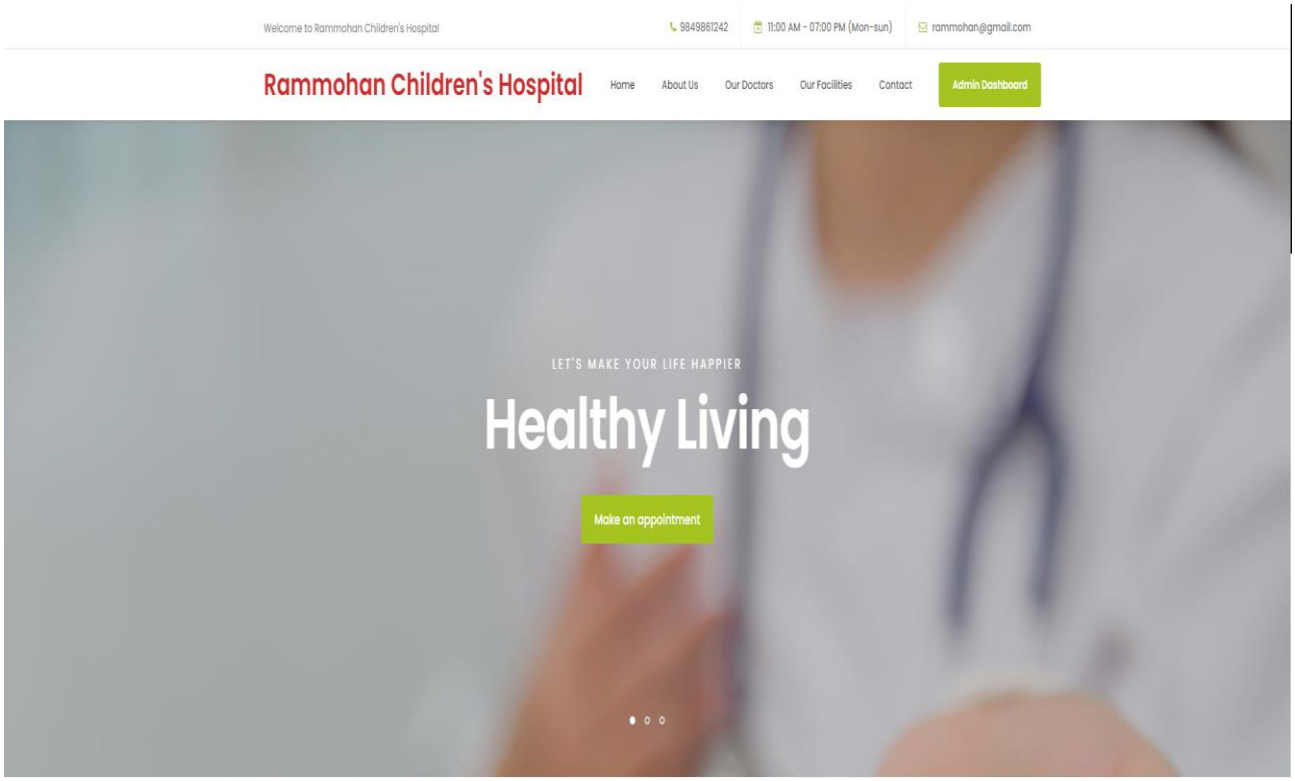


Figure-5.2.1 Screenshot of Homepage

Our Doctor:

This figure tells how well-trained our doctor is.

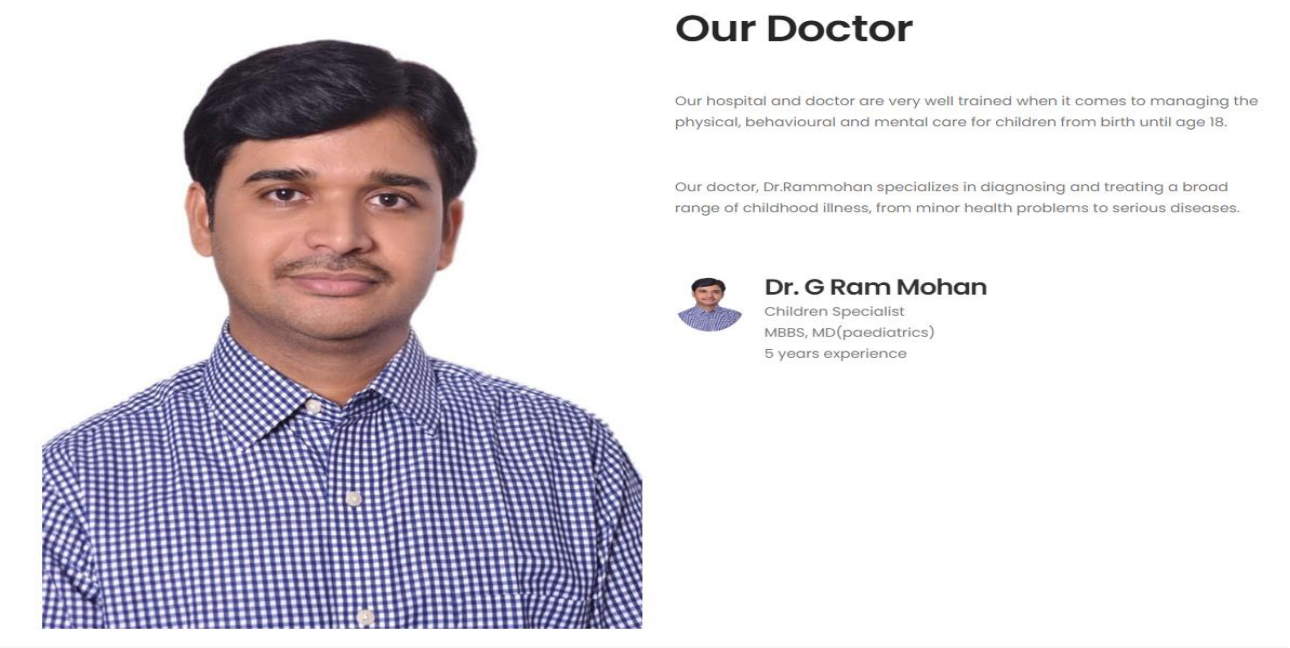


Figure-5.2.2 Screenshot of Doctor Information

Our Facilities:

This figure tells about the hospital facilities provided.

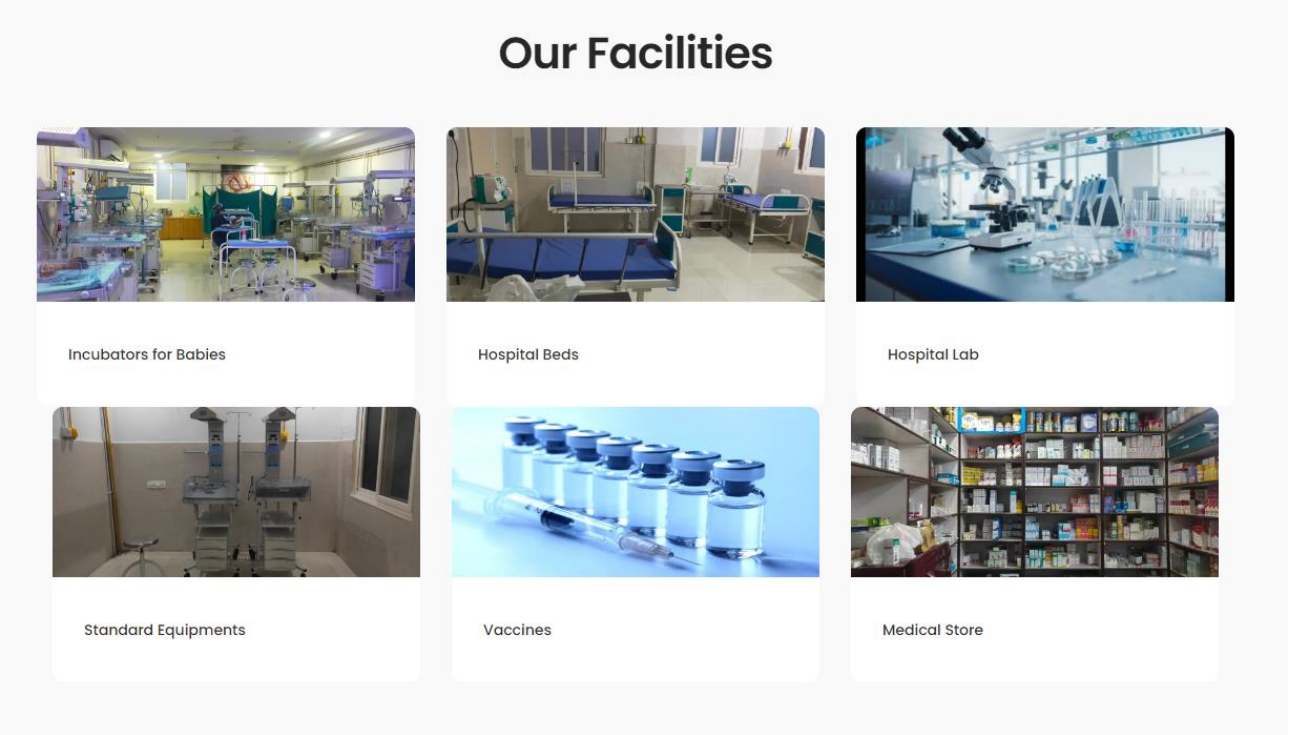


Figure- 5.2.3 Screenshot of Facilities

Registration:

This figure tells about the registration i.e appointment, reappointment, vacccination, revaccination

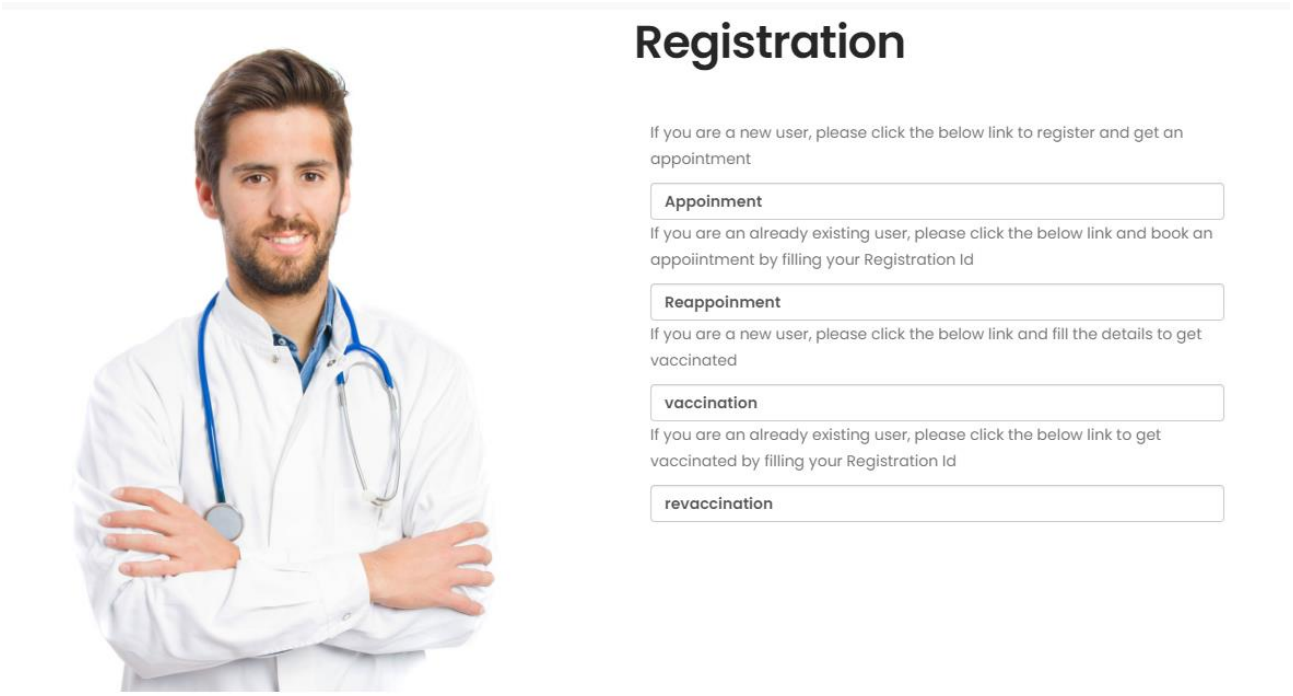


Figure-5.2.4 Screenshot of User Registration page

Contact:

This figure shows the route map of the hospital.

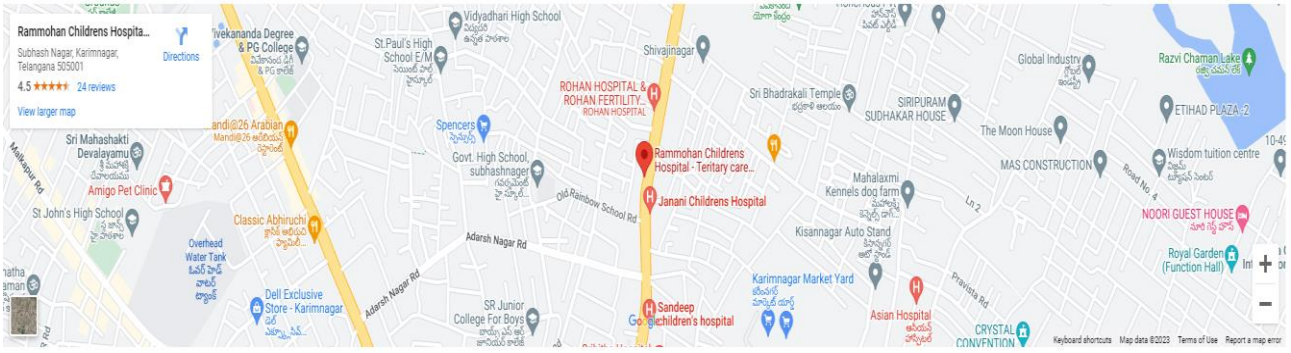


Figure-5.2.5 Screenshot of location

This figure shows the contact details of the hospital.

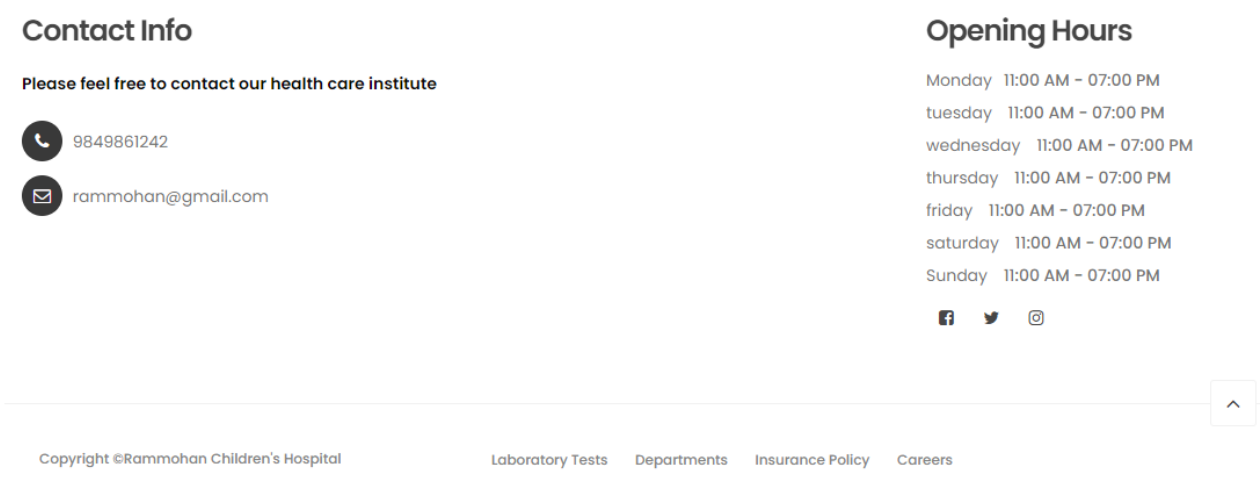


Figure-5.2.6 Screenshot of contact information

Admin Login:

This figure shows the admin login page.

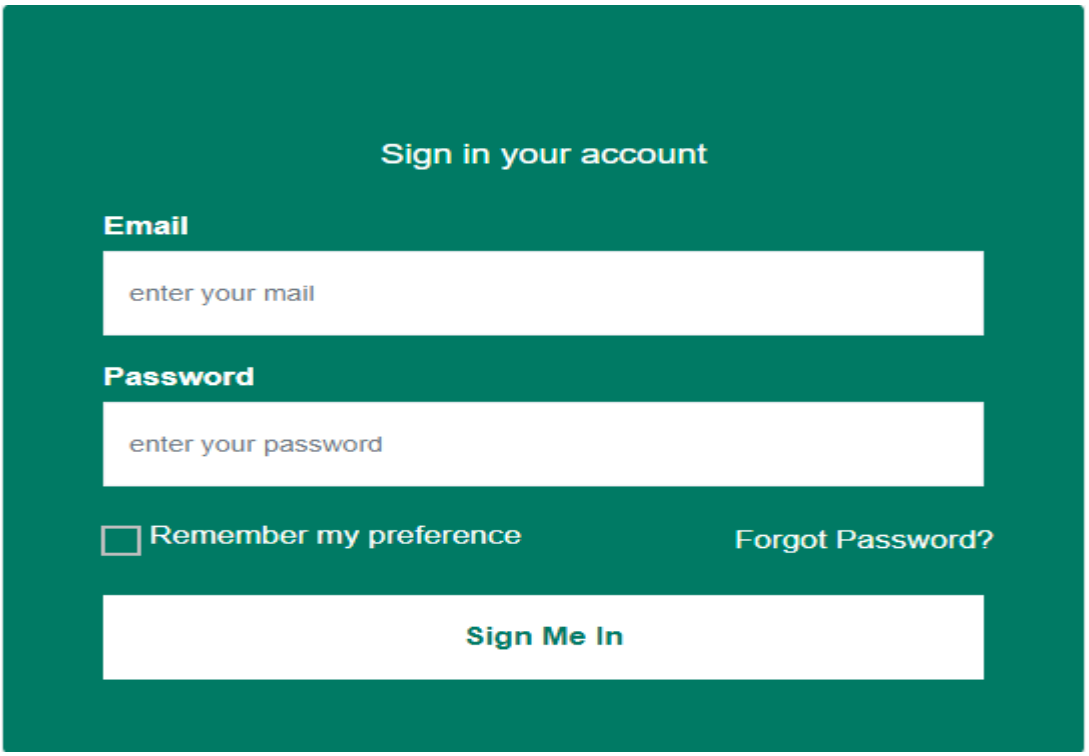


Figure-5.2.7 Screenshot of Admin login page

CHAPTER-6

TESTING

It is the process of testing the functionality and executing a program with the intent of finding an error. Testing is a crucial element of software quality assurance and presents the ultimate review of specification, design and coding. System testing is an important phase. Testing represents an interesting anomaly for the software. Thus a series of testing are performed before the system is ready for user acceptance testing. A good test case is one that has a high probability of finding an undiscovered error. A successful test is one that uncovers an undiscovered error. Software testing is usually performed for one of two reasons:

- Defect Detection
- Reliability estimation

Testing is a process,which reveals errors in the program. It is the major quality measure employed during software development. During testing, the program is executed with a set of conditions known as test cases and the output is evaluated to determine whether the program is performing as expected. Software testing is the process of testing the functionality and correctness of software by running it.

6.1 Testing

6.1.1 Admin Login Page

1. Figure 6.1.1 shows trying to login with empty.fields.The expected output is to show “Please enter your email”.
2. Figure 6.1.2 shows trying to login with email.and leaving the password field empty.
   The expected output is to show “Please enter your password”.
3. Figure 6.1.3 shows trying to login with an email of invalid format.
   The expected output is to show “Please include an @ in the email address”.
4. Figure 6.1.4 shows trying to login with invalid details. The expected output is to show “Invalid credentials!”
5. Figure 6.1.5 shows trying to login with the correct username and correct password.
   The expected output is “View of Admin Dashboard”, if username and password are valid.

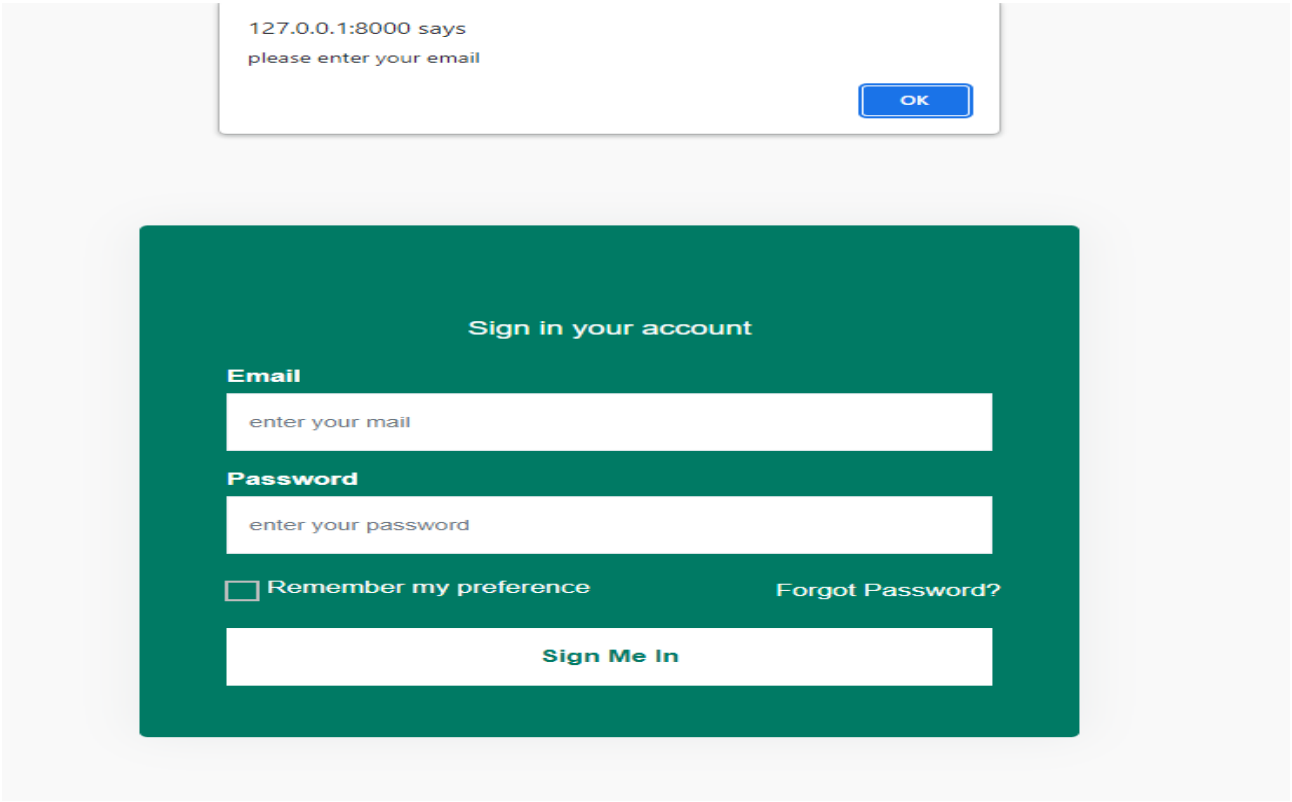


Figure 6.1.1 Email field is empty

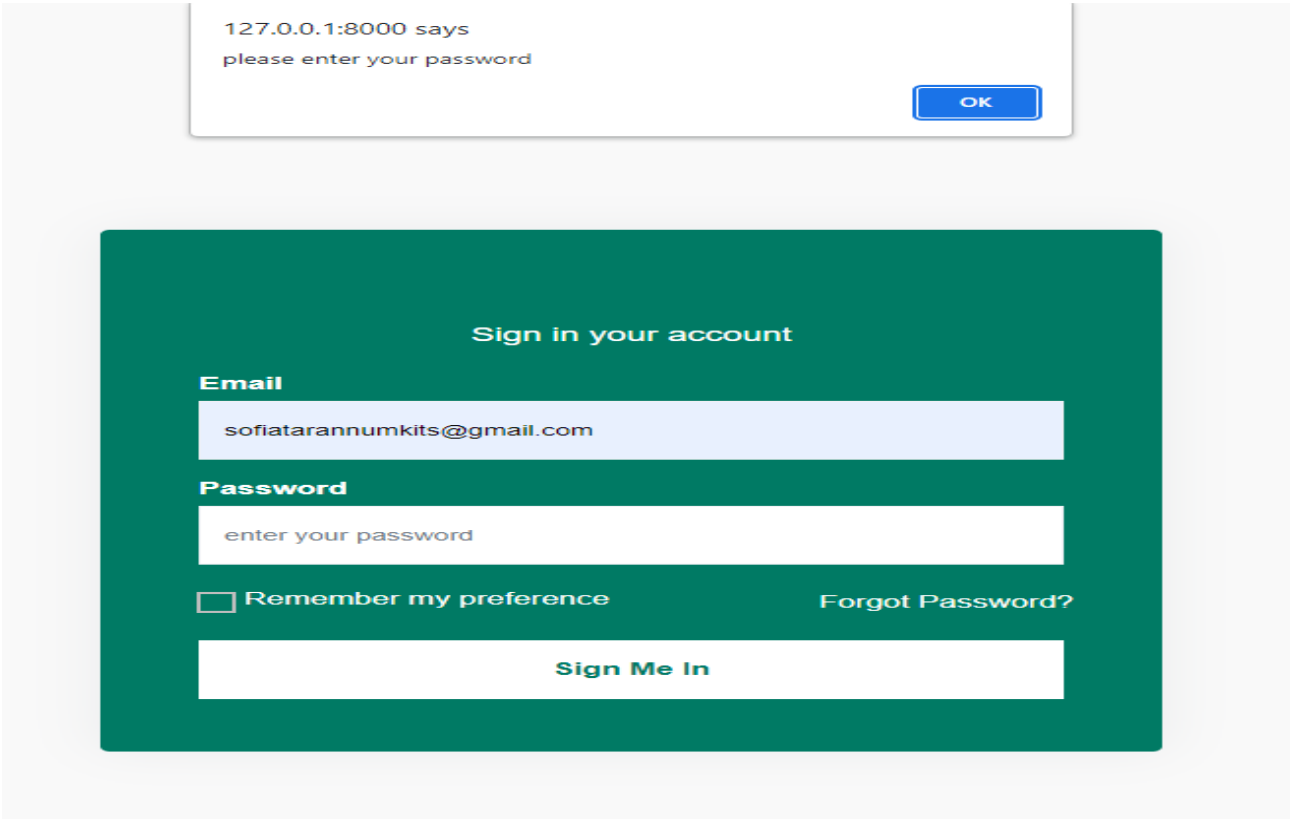


Figure 6.1.2 Password field is empty

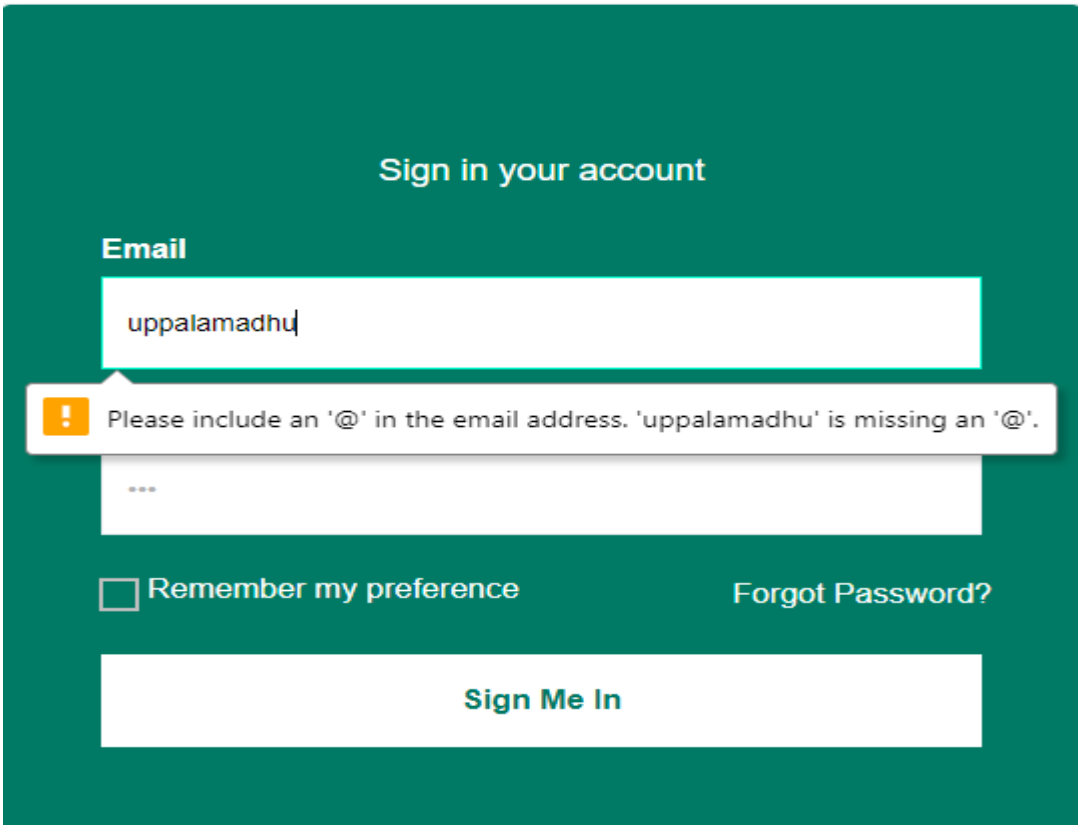


Figure 6.1.3 Invalid email format

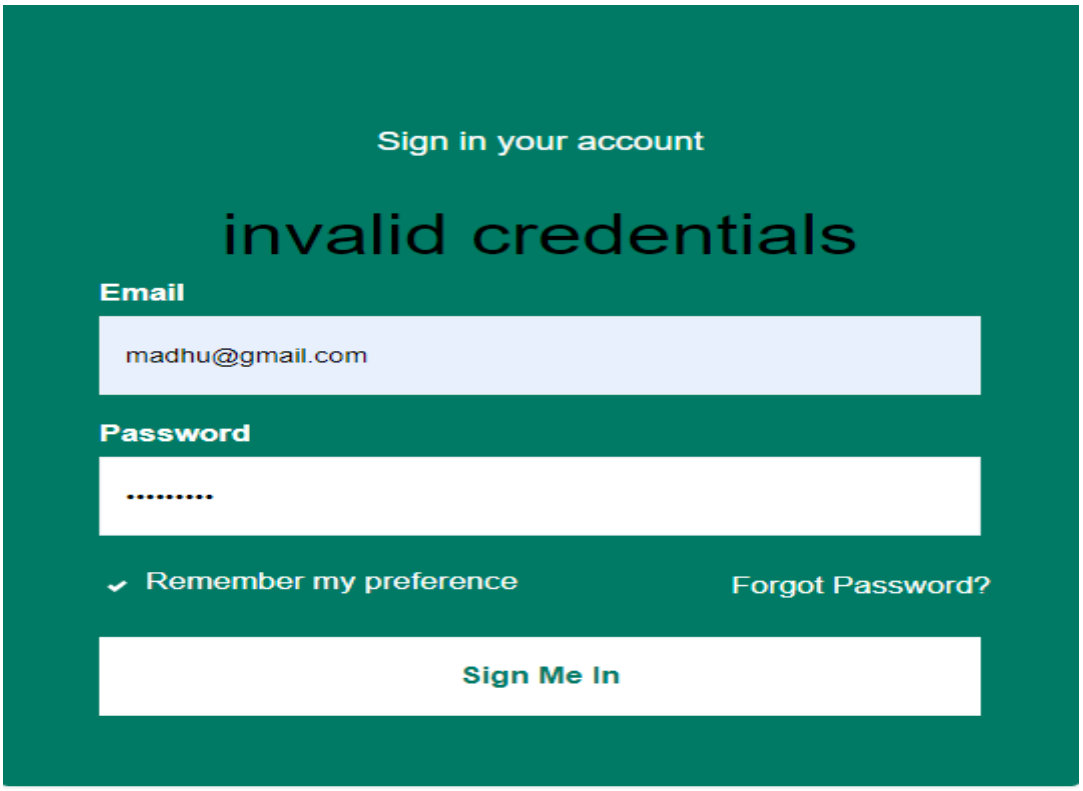


Figure 6.1.4 Invalid credentials

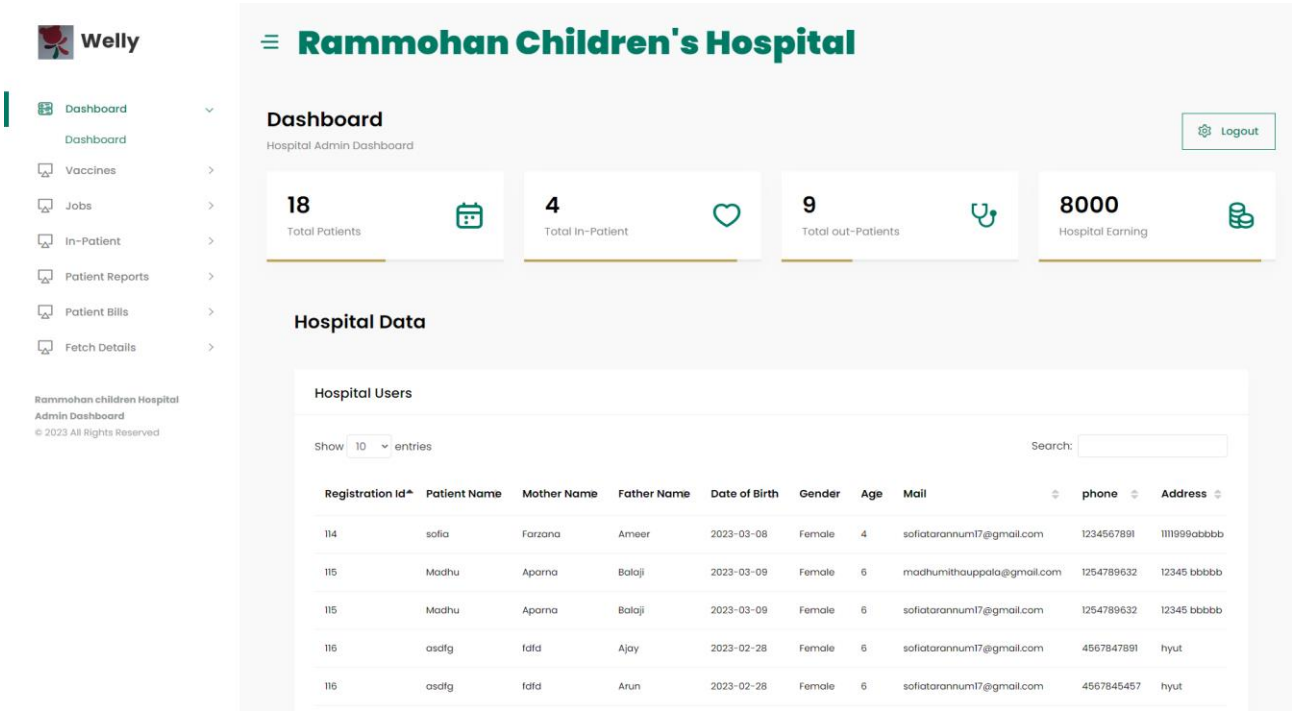


Figure 6.1.5 Successful login

Test no.	Data input	Expected output	Actual output	Pass/Fail
1	Email field is empty	Please enter your email	Please enter your email	Pass
2	Password field is emply	Please enter your password	Please enter your password	Pass
3	invalid email format	Please include an @ in the email address	Please include an @ in the email address	Pass
4	Wrong email or password	Invalid credentials	Invalid credentials	Pass
5	correct username and password	View of Dashboard	View of Dashboard	Pass

Figure: 6.1 Outputs table

6.2 Database Testing

This test involves testing whether the values entered through the form gets stored in the database correctly or not.

Figure 6.2.2 shows that entered details are stored correctly in the database or not.

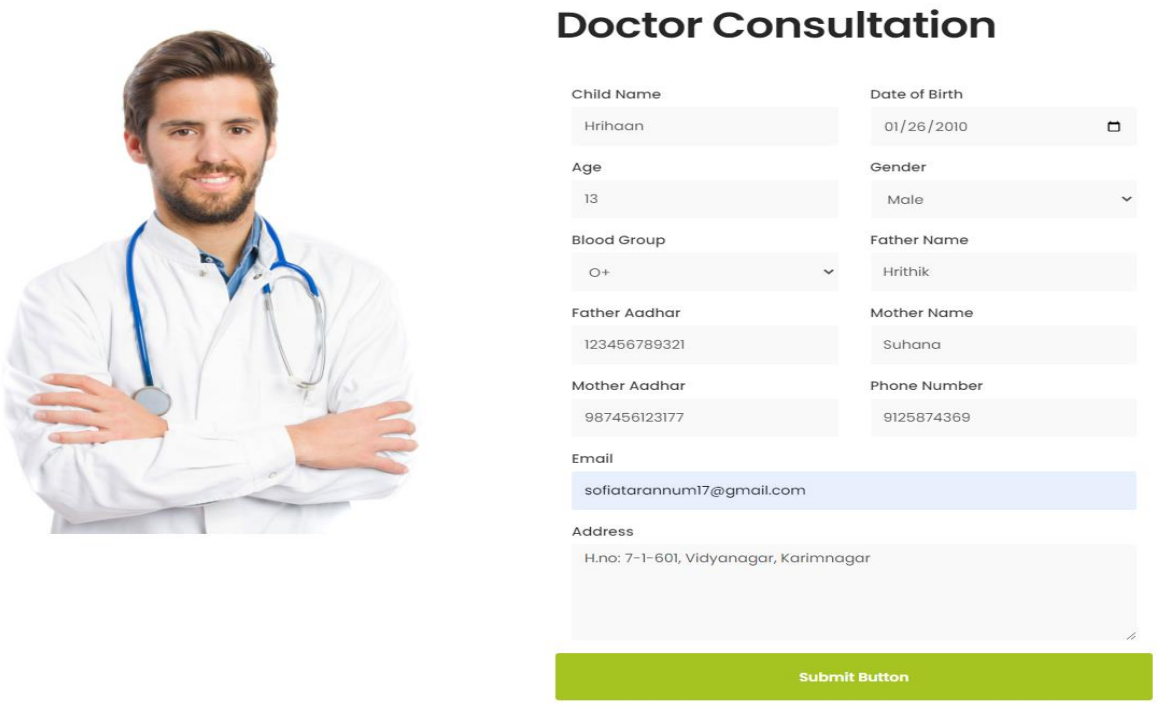


Figure: 6.2.1 Doctor Consultation form

	id	cname	dob	age	gender	cbloodgrp	fname	faadhno	mname	maadhno	phone	mail	address
Edit Copy Delete	10		0000-00-00	0				0		0	0		
Edit Copy Delete	74	sofia	2023-03-08	4	Female	O+	Ameer	145623789123	Farzana	123456456456	1234567891	sofiatarannum17@gmail.com	1111999abbbb
Edit Copy Delete	75	Madhu	2023-03-09	6	Female	O+	Balaji	123456789012	Aparna	98765432112	1254789632	madhumithauppala@gmail.com	12345 bbbbb
Edit Copy Delete	76	Madhu	2023-03-09	6	Female	O+	Balaji	123456789012	Aparna	98765432112	1254789632	sofiatarannum17@gmail.com	12345 bbbbb
Edit Copy Delete	77	asdfg	2023-02-28	6	Female	O+	Ajay	145678903214	fdfd	345121212121	4567847891	sofiatarannum17@gmail.com	hyut
Edit Copy Delete	78	asdfg	2023-02-28	6	Female	O+	Arun	145678903214	fdfd	121454545454	4567845457	sofiatarannum17@gmail.com	hyut
Edit Copy Delete	79	sofia	2023-03-08	4	Female	O+	Ameer	145623789123	Farzana	123456456456	1234567891	sofiatarannum17@gmail.com	1111999abbbb
Edit Copy Delete	80	Saba	2023-03-28	7	Female	O+	Akbar	123456789987	Naziya	147899632547	1236547892	sofiatarannum17@gmail.com	bbbb
Edit Copy Delete	82	madhu	2023-05-10	13	Female	O+	raju	132347878787	rani	224412121212	3656541456	sofiatarannum17@gmail.com	4444
Edit Copy Delete	83	sofia	2023-05-09	12	Male	O+	rtyu	132347878787	fdfd	224412478978	3656547878	sofiatarannum17@gmail.com	555
Edit Copy Delete	84	Hrihaan	2010-01-26	13	Male	O+	Hrithik	123456789321	Suhana	987456123177	9125874369	sofiatarannum17@gmail.com	H.no: 7-1-601, Vidyanagar, Karimnagar

Figure: 6.2.2 Database Messages details

6.3 Acceptance Testing

Testing to verify a product meets customer specific requirements. The acceptance test suite is run against supplied input data. Then the results obtained are compared with the expected results of the client. A correct match was obtained.

Test cases

Test case1: Admin login(successful)

Test case: Admin logged in successfully.

Test description: Provide Admin login by checking the details.

Pre Condition: Database connectivity.

Action Performed: Entered valid login details without leaving any field.

Expected results: Successful login.

Condition verified: Yes

Result: Success.

Test case2: Admin login(unsuccessful)

Test case: Admin logged (unsuccessful)

Test description:report error message

Pre Condition: Database connectivity.

Action Performed: Entered invalid login details

Expected results: display message as invalid login

Result: Success.

6.4 Test Results

Test cases are done on every page on this website. All the test cases are passed successfully. No defects were encountered.

CHAPTER 7

RESULTS

After performing testing we get certain results. The results obtained from testing as shown below. Results of the system can be expressed and evaluated in terms of output screen. The output screen can be used to show the objectives set at the beginning are achieved at the end.

The below figure shows the admin dashboard, which shows the hospital user details.

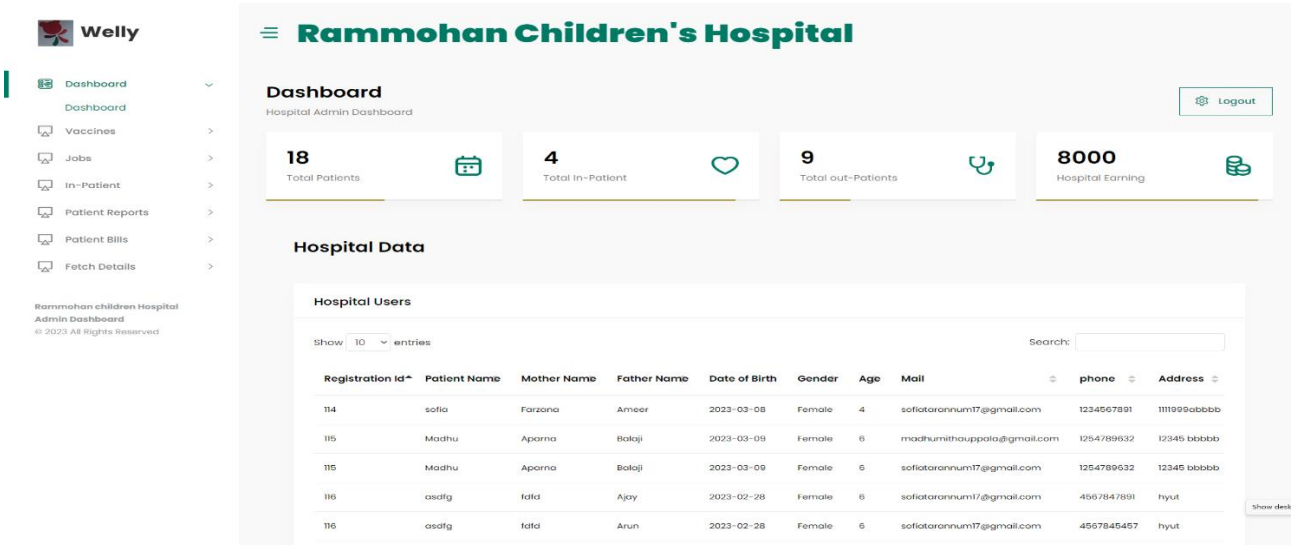


Figure 7.1 Screenshot of Admin dashboard

The below figure shows how to edit the status from out-patient to in-patients.

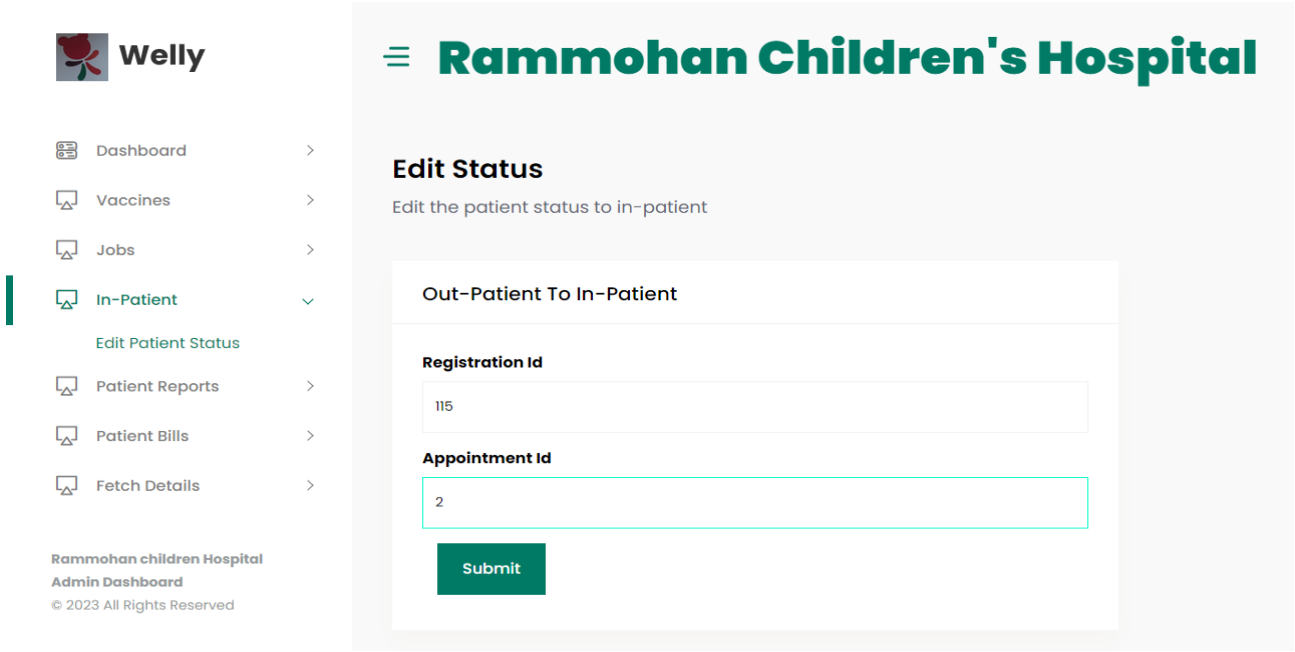


Figure 7.2 Screenshot of Edit Status

The below figure shows how to add vaccines which are visible to the users.

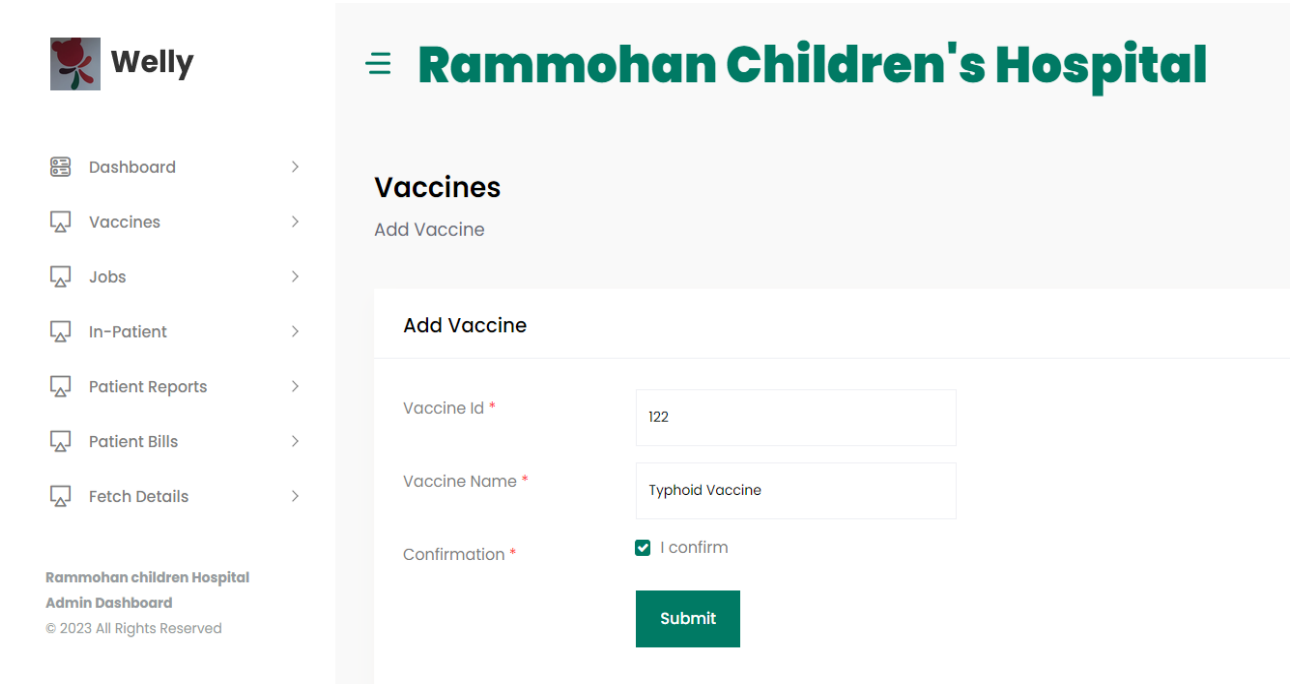


Figure 7.3 Screenshot of Add Vaccine

The below figure shows the vaccine available which can be edited and deleted.

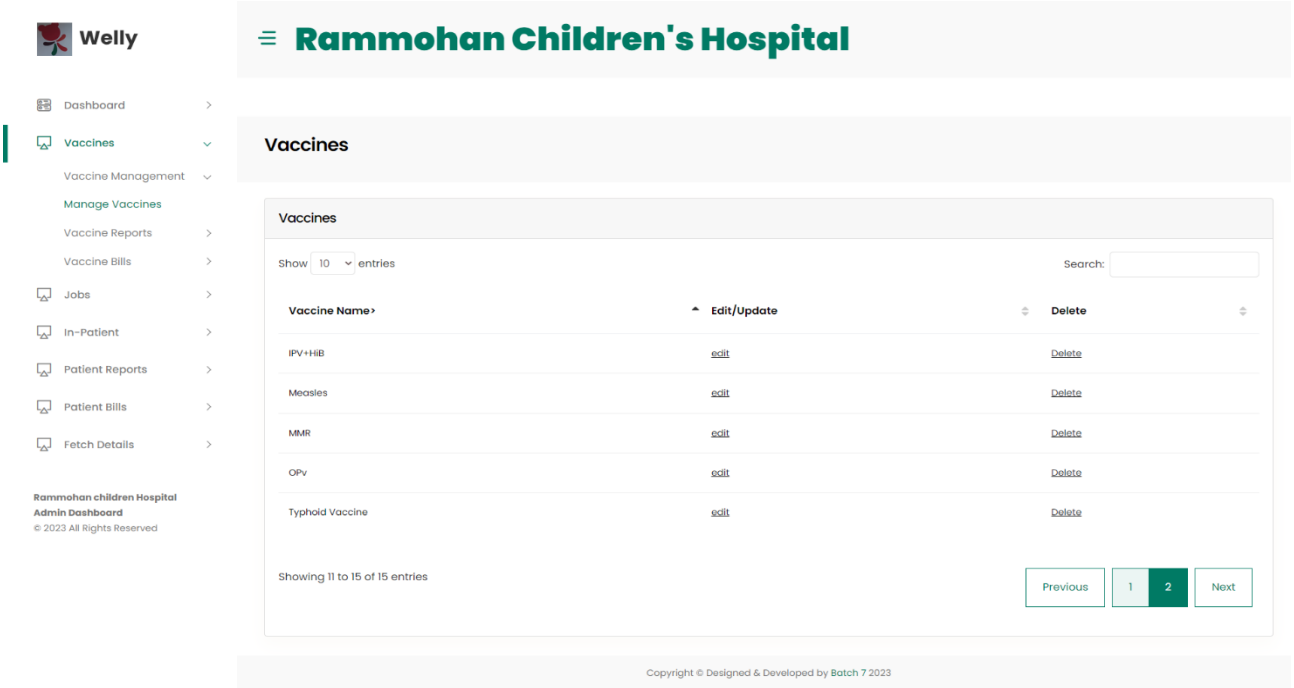


Figure 7.4 Screenshot of Vaccines

The below figure shows how to generate vaccine reports.

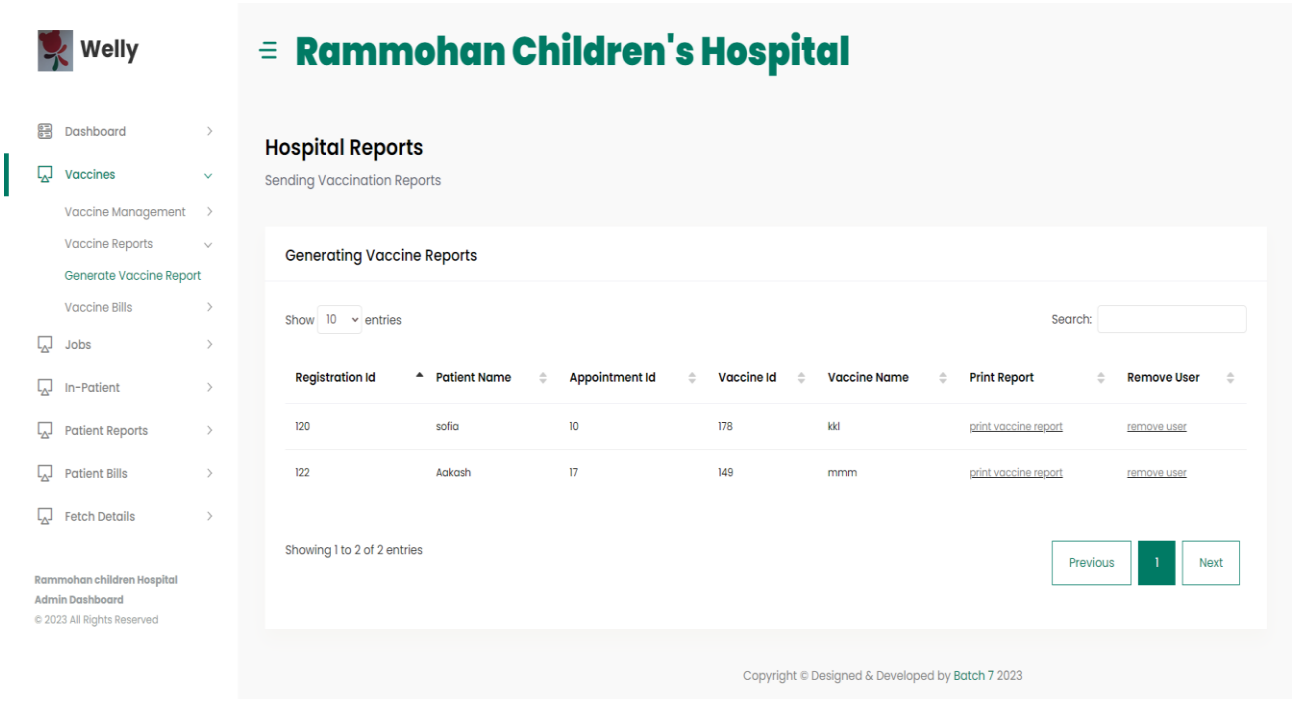


Figure 7.5 Screenshot of Generating vaccine reports

The below figure shows the sample of a vaccine report.

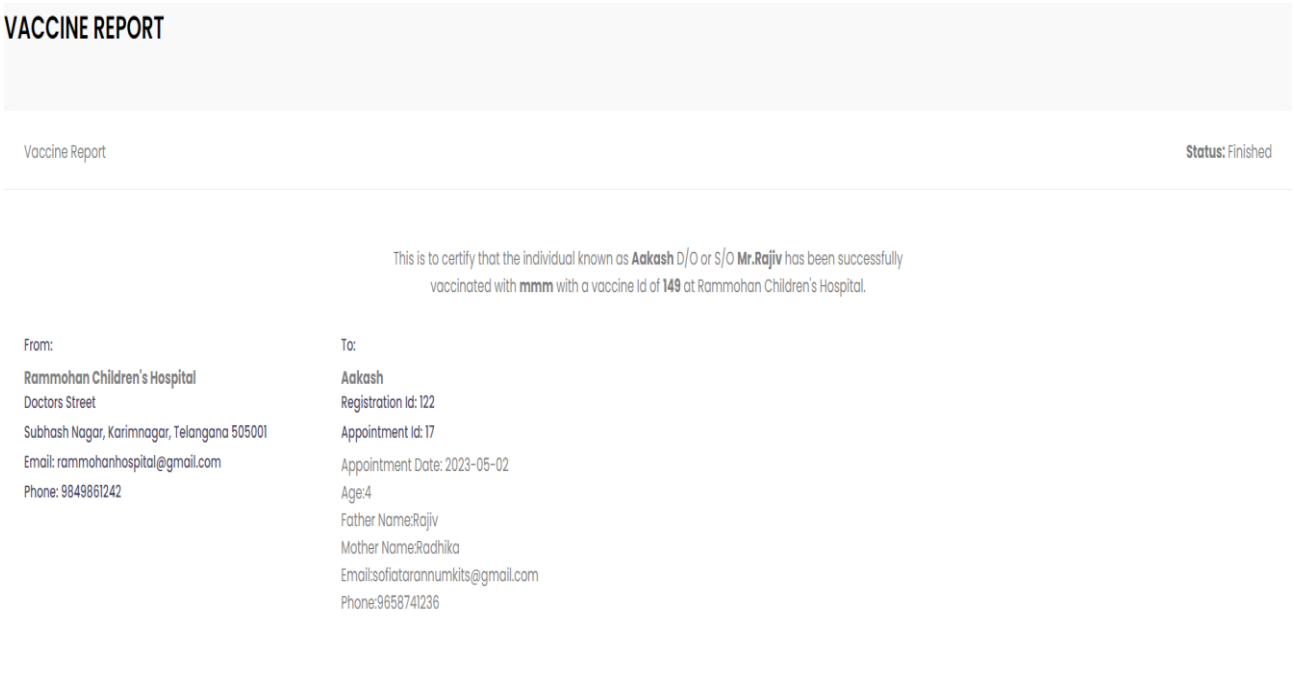


Figure 7.6 Screenshot of sample vaccine report

The below figure shows how to upload the vaccine report of a patient .

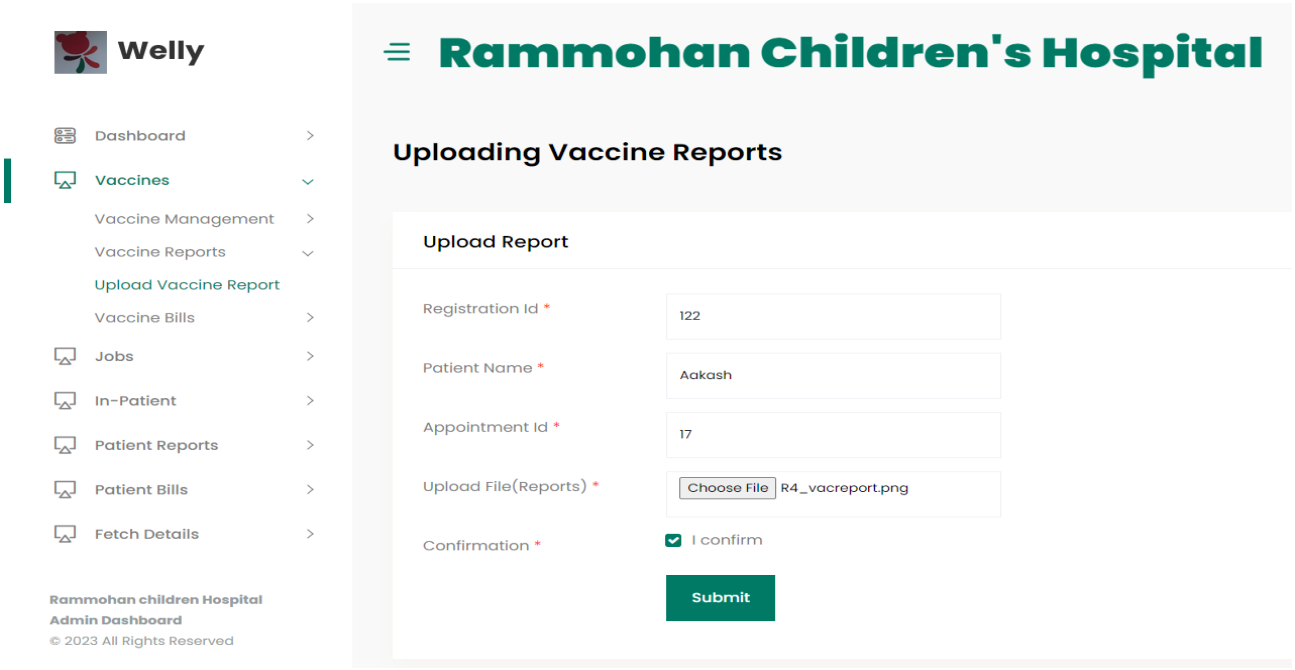


Figure 7.7 Screenshot of uploading vaccine reports

The below figure shows the reports of patients available, when clicked on send reports a mail with an attachment of vaccine report is sent to the respective user.

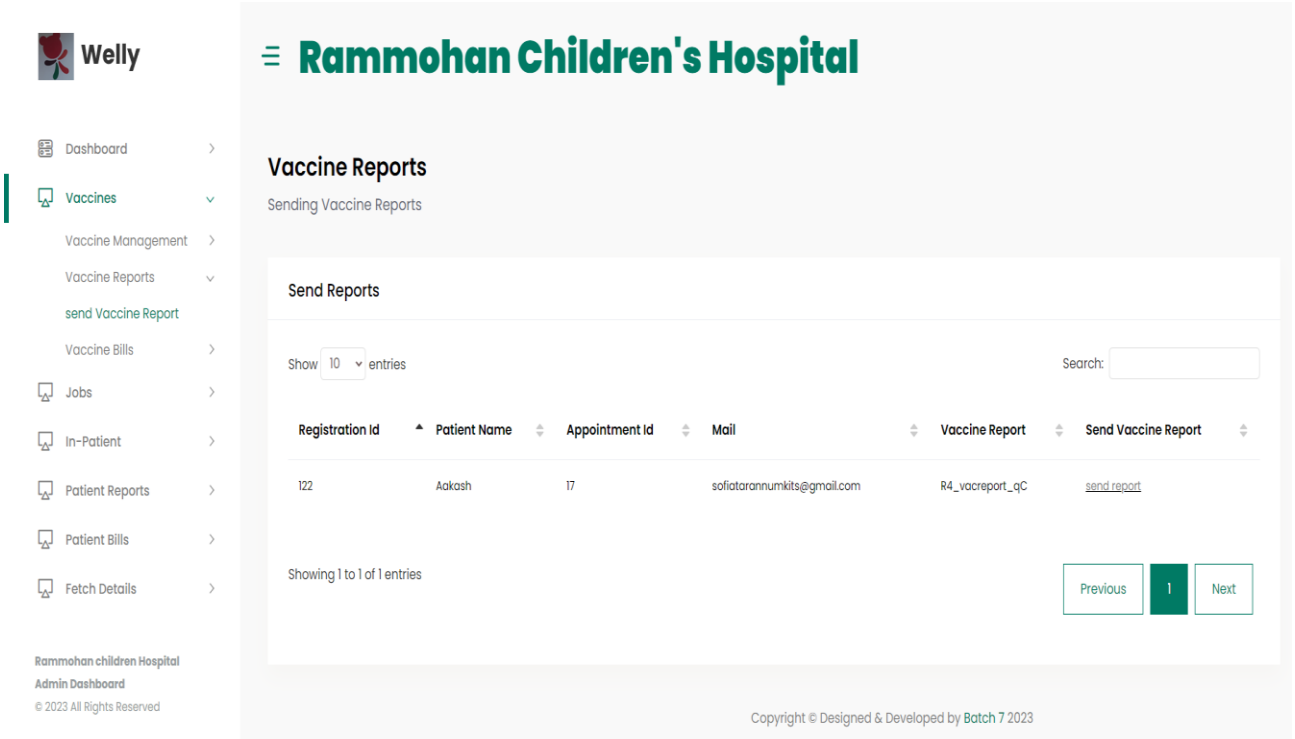


Figure 7.8 Screenshot of sending vaccine reports

The below figure shows that the vaccine reports has been successfully sent to the user.

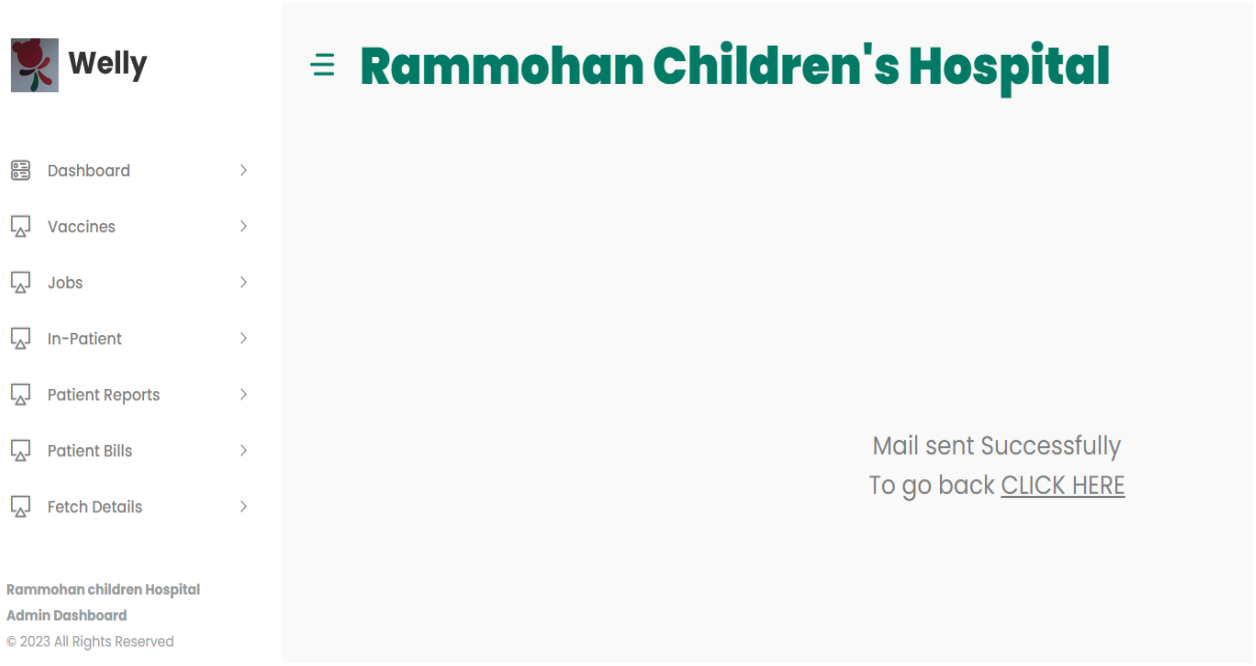


Figure 7.9 Screenshot of mail sent successfully

The below figure shows the patients to whom the bills are to be generated.

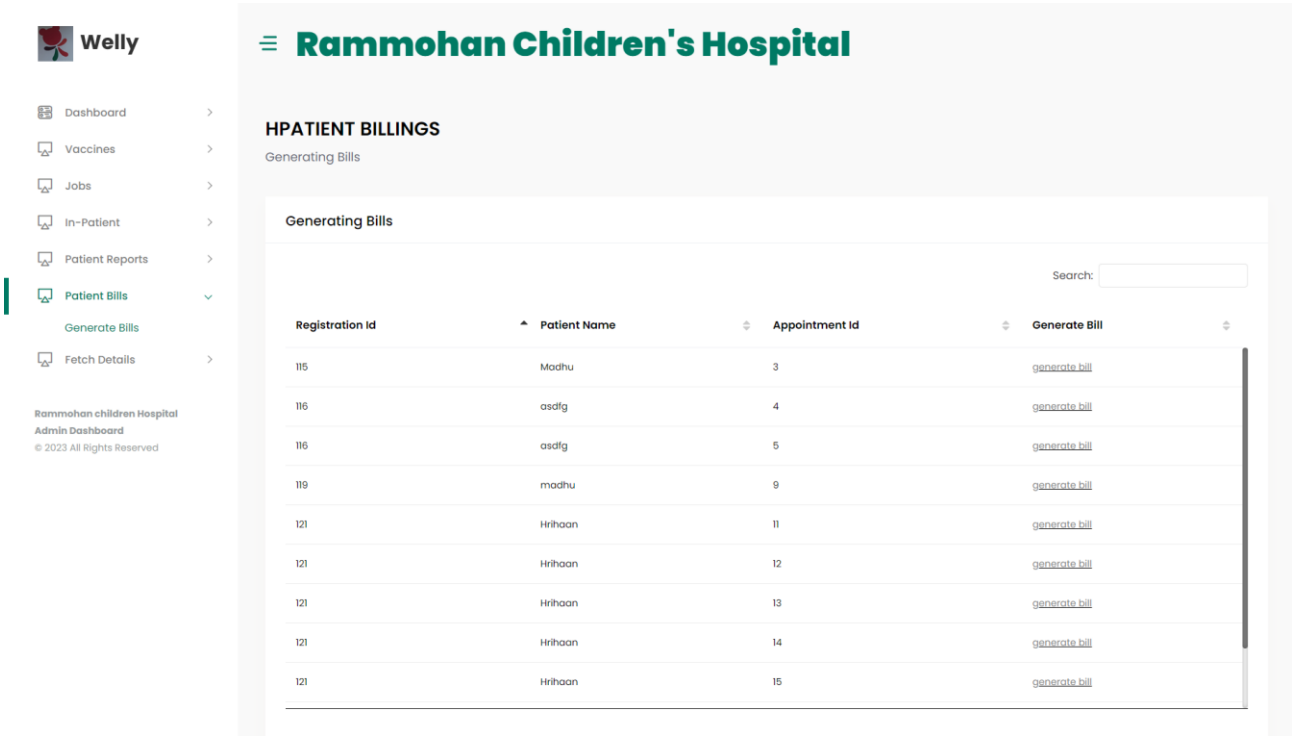


Figure 7.10 Screenshot of generating bills

The below figure shows the shows the hospital bill form .

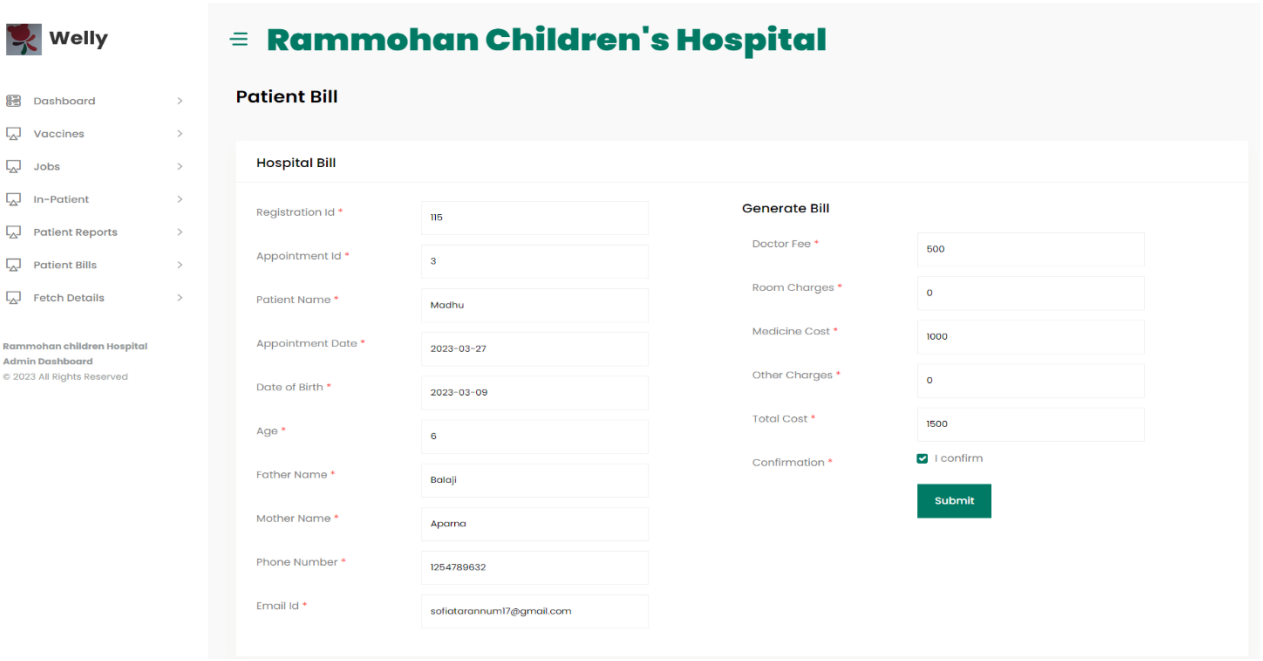


Figure 7.11 Screenshot of hospital bill form

The below figure shows the patients whose bills are to be printed.

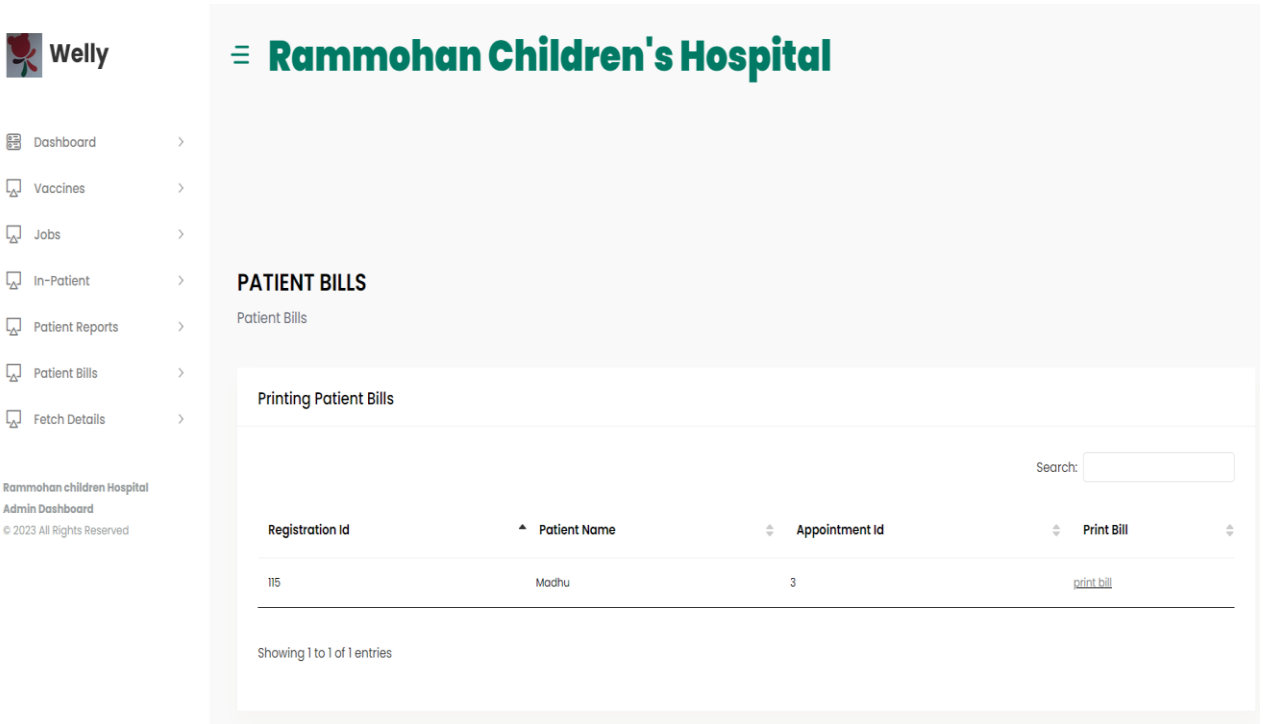
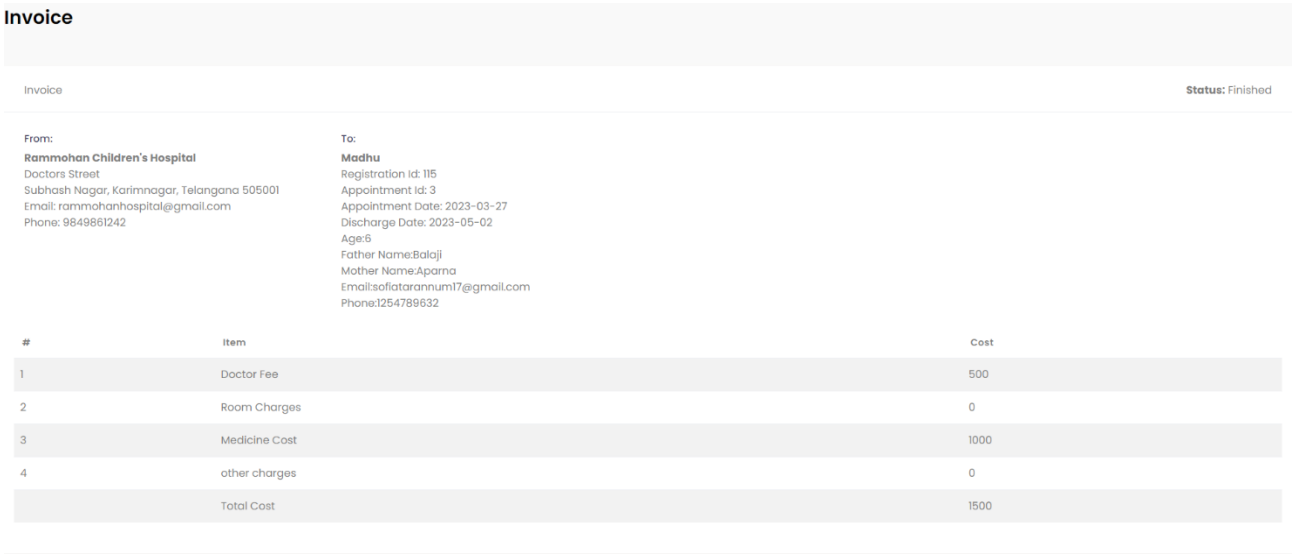


Figure 7.12 Screenshot of printing bills

The below figure shows the sample hospital bill of a patient.



Invoice

Invoice — **Status:** Finished

From:
Rammohan Children's Hospital
Doctors Street
Subhash Nagar, Karimnagar, Telangana 505001
Email: rammohanhospital@gmail.com
Phone: 9849861242

To:
Madhu
Registration Id: 115
Appointment Id: 3
Appointment Date: 2023-03-27
Discharge Date: 2023-05-02
Age:6
Father Name:Balaji
Mother Name:Aparna
Email:sofiatarannum17@gmail.com
Phone:1254789632

#	Item	Cost
1	Doctor Fee	500
2	Room Charges	0
3	Medicine Cost	1000
4	other charges	0
	Total Cost	1500

Figure 7.13 Screenshot of sample hospital bill

CHAPTER-8

CONCLUSION

The project we developed is “WEB APPLICATION FOR RAMMOHAN CHILDREN’S HOSPITAL”, which has become a powerful tool to help the healthcare organization to improve their processes and streamline their operations. It provides an integrated, comprehensive solution to managing patient records and billings. Throughout the project, we have identified and addressed various challenges, including designing a user-friendly interface, implementing efficient data management and ensuring platform security. We implemented the designs and developed the code using the Django framework in python. We also designed the database tables using SQL. We incorporated several features such as making appointments for check-ups and vaccines, donating blood, applying for jobs, generating online reports, bills and sending them to the respective users and many more.

Our project helps the admin to easily fetch the in-patient and out-patient data and also helps the users to get bills, reports through emails which saves a lot of time and effort standing in queue. We are confident that our platform will provide significant value to both hospital administrators and users allowing them to seamlessly connect to the hospital facilities. Overall, the children’s hospital project is vast and there are numerous opportunities for growth and development. The key is to stay current with the latest trends in technology and be responsive to the needs of patients and their families.

CHAPTER-9

FUTURE SCOPE& ENHANCEMENTS

The project has a very vast scope and contains several potential areas of improvement and enhancement. Here are some possible future enhancements to the project:

1. **Use of Artificial Intelligence and Machine Learning:**

   Artificial intelligence and machine learning could be used to automate routine tasks, such as patient triage and appointment scheduling. This could free up staff time and reduce errors.

2. **Implementation of Telemedicine:**

   Telemedicine could allow patients to access healthcare remotely, reducing the need for in-person appointments and improving patient convenience.

3. **Online Payment:**

   As our current project does not include any online payments, therefore this project can be enhanced with online payments to provide better convenience to both users and hospital administrators.

4. **Donations and Funds:**

   Donations and funds can be incredibly helpful to the hospital in a number of ways, such as purchasing equipment and supplies, supporting research, improving facilities, and providing financial support to patients.

5. **Hospital staff management:**

   As the current project is only reserved to patients, therefore the project can further be enhanced to staff management which will include all their details such as joining details, personal details, salaries etc.

BIBLIOGRAPHY

1. HTML Black Book by Steven Holzner.

2. The Unified Modeling Language User Guide, by Grady Booch, James Rumbaugh;2nd Edition; Addison-Wesley Professional; ISBN-10: 0321267974, ISBN-13: 978 0321267979

3. Fundamentals of Python Programming by D. Prashanth Kumar ,Dr.K.Praveen Kumar Rao; ISBN-13: 978-8195376551, ISBN-10: 819537655X

4. Software Testing concepts and Tools, by NageshwarRao Pusuluri; 2nd Edition; Dream tech press; ISBN-10: 8177227122, ISBN-13:978-8177227123.

5. Database Modeling and Design, by Toby J.Teorey.Sam S.Lightstone, Tom Nadeau;5th Edition; Morgan Kaufmann;ISBN-10:0123820200,ISBN-13:978-0123820204.

6. Database Design using Entity-Relationship Diagrams by Richard Earp 2nd Edition, April 16,2007;Auerbach Publications; ISBN-10:1439861765, ISBN-13:978-1439861769.