

# **Efficient Doctor-Patient Portal**

*Project report*

*submitted in fulfillment of the requirements for the Degree of*

**BACHELOR OF TECHNOLOGY**

By

Ashutosh Bhati (141243)

Porshiya Mangal (141246)

Varun Garg (141250)

Chirag Ahuja (141302)

Rajat Chauhan (141316)

Arkapravo Kar (141321)

Under the supervision of

**Ms. Anjali B.**



**Department of Computer Science & Engineering**

**JAYPEE UNIVERSITY OF INFORMATION  
TECHNOLOGY, WAKNAGHAT**

**May, 2018**

## Candidate's Declaration

We hereby declare that the work presented in this report entitled "Efficient Doctor-Patient Portal" in partial fulfillment of the requirements for the award of the degree of **Bachelor of Technology in Computer Science and Engineering submitted in the department of Computer Science & Engineering and Information Technology**, Jaypee University of Information Technology Waknaghat is an authentic record of our own work carried out over a period from under the supervision of

The matter embodied in the report has not been submitted for the award of any other degree or diploma.

Ashutosh Bhati (141243)

Porshiya Mangal (141246)

Varun Garg (141250)

Chirag Ahuja (141302)

Rajat Chauhan (141316)

Arkapravo Kar (141321)

This is to certify that the above statement made by the candidate is true to the best of my knowledge.

Supervisor Name: Anjali B.

Department Name: ETA

Dated: 19.5.2018

## **Acknowledgement**

Behind every achievement lies an unfathomable sea of gratitude of those who energized us, without whom it would never have come into existence. It is our immense pleasure to express our gratitude, regards and heartfelt respect to for her endless and extreme support during and beyond the tenure of the project work. Her advices have always lighted our path whenever we got stuck in our work. We would also like to thank all the faculty and staff of CSE department, JUIT Waknaghat, Solan for their support and help in providing the resources which were required. We would like to thank all our other friends who encouraged and supported us in every step of our career and personal life. Last but not least, we would like to express our love, respect and gratitude to our parents, brothers, sisters and all our family members, who have always been with us in every decision we have made, without whom we would have never been able to achieve whatsoever we could have till date.

## Table of Contents

Sr. No.	Topics	Page no.
<b>1</b>	<b>Introduction</b>	
1.1	Introduction	1-2
1.2	Problem Statement	2
1.3	Objectives	2-3
1.4	Methodology	3-9
1.5	Organization	10
<b>2</b>	<b>Literature Survey</b>	11
2.1	HTML	11-12
2.2	CSS	12-13
2.3	Bootstrap	14-15
2.4	Angular JS	16-20
2.5	Java Service Layer	20-21
2.6	Java DAO Layer	21
2.7	Hibernate	21-22
<b>3</b>	<b>System Development And Algorithms</b>	23
3.1	Hardware Requirements	23
3.2	Software Requirements	23
3.3	Software Development model	23-24
3.4	Class diagrams	24-25
3.5	Entity Relationship diagram	26
3.6	Agile Sheet	26-30
3.7	Algorithms	30-38
<b>4</b>	<b>Performance Analysis And Result</b>	
4.1	Performance Analysis And Result	39-52
<b>5</b>	<b>Conclusions</b>	
5.1	Conclusions	53
5.2	Future Scope	53
<b>6</b>	<b>References</b>	
	References	54

## List of Figures

<b>Figure No.</b>	<b>Name</b>	<b>Page No.</b>
1	Agile Model	9
2	Bootstrap Usage	14
3	Module, Scope, Controller Relation	18
4	Scope Hierarchy	20
5	Agile Iterations	24
6	Entity Relationship Diagram	26
7	Register Algorithm	31
8	Login Algorithm	32
9	Book Appointment Algorithm	33
10	Organ Registration Algorithm	34
11	Admin Login Algorithm	35
12	Add Doctor Algorithm	36
13	Add Treatment	38

# CHAPTER 1 - INTRODUCTION

## 1.1 INTRODUCTION

In large hospitals, massive staff members and doctors are present. Efficient management of time and scheduling of tasks becomes a very tough task. At present, a person sick from any disease needs to visit the hospital to book an appointment. All the patients need to wait in a queue while getting an appointment. In case that a doctor calls off an appointment due to some emergency reasons then the patient remains unaware about the cancellation of the appointment till the time he or she visits the hospital. Also whenever the patient visits the doctor he or she is asked to bring their prescriptions. Doctors don't keep a record of when the patient last visited him for an appointment. Doctors are unaware of who all will visit him on a given day and hence are unable to manage the appointments accordingly. Doctors are not able to access the patient details and medical history until and unless he/she checks the medical report.

Sometimes patients are not able to book appointments according to their convenience. The patients don't have the liberty to reschedule their scheduled appointment in case of any emergency. In case the patient requires an organ then he/she doesn't have a record of all the potential organ donors.

We have designed an Efficient doctor patient portal which is able to manage and help doctors along with patients. It consists of the following modules:

1. Admin Login: The system can be supervised by the admin. The admin can add doctors according to their specialty, view patient's medical history and can also view all the organ donors.
2. User Login\Registration: The patients will first need to register themselves and only after that they can login into the system.
3. Medical History: In our portal there is a feature to update and view the medical history and information of the patients.
4. Doctor Search: System allows for doctor search through categories, name and location.
5. Appointment Booking: Patients will need to login into the system. They will then search for the doctors as per their specialty and can book appointment for their required date and time. If that time slot is not free then he/she will need to choose other time slot.
6. Booking cancellation: To cancel the booking, the patient will need to login into the system and then select the option to cancel the appointment.
7. Feedback: The patients can provide feedback of the doctors by whom they were treated. This leads to transparency.
8. Organ Donor Registration: A patient will have to register himself so that he can donate organ.
9. Organ Donor Search: Patient can search an organ according to his/her requirements.

With the help of this system the doctors can very easily manage the appointment slots online. This system helps in managing various doctors based on their available timings and dates. The complete medical history of all the patients is maintained in the database so that every time a patient login into the system he/she is able to access all their medical history. The doctors can refer to the medical history of the patients whenever required. The doctor can view whole of the patient's medical history even before the patient comes to see the doctor.

This system enables to have an automated doctor patient managing system using an online interface. Another module included in this system is organ donor module which provides patients the option of searching for the desired organ. Also it provides an option of registration in order to donate an organ.

## **1.2 PROBLEM STATEMENT**

Proposed doctor patient handling, managing system helps doctors in their work and also patients to book doctor appointments and view medical progress. The system allows doctors to manage their booking slots online. Patients are allowed to book empty slots online and those slots are reserved in their name. The system manages the appointment data for multiple doctors for various date and times. It also allows for organ donation registration as well as organ search. The module is designed to help urgent organ requirements through easy/instant searches.

## **1.3 OBJECTIVES**

1. As an admin I should be able to login through an ID and Password.
2. As an admin I should be able to logout.
3. As an admin I should be able to add Doctors according to their specialty.
4. As an admin I should be able to view Doctors of all the departments.
5. As an admin I should be able to view Patient details along with their medical history.
6. As an admin I should be able view all the organ donors along with their blood group.
7. As an admin I should be able to view all the feedbacks given by the patients.
8. As a doctor I should be able to login through an ID and password provided by the admin.
9. As a doctor I should be able to view my appointments.
10. As a doctor I should be able to view patient details and their medical history.
11. As a doctor I should be able to add treatment for the patients.
12. As a doctor I should be able to view my details.
13. As a doctor I should be able to Logout.
14. As a patient I should be able to register in case I have not yet registered.
15. As a patient I should be able to login though a login ID and Password.
16. As a patient in case I forget the password I should be able to reset it.
17. As a patient I should be able to able to book appointment for a specified time slot for a specific doctor.
18. As a patient I should be able to view my previous appointments.
19. As a patient I should be able to cancel my appointments through my appointment ID.
20. As a patient I should be able to search doctor by their name, specialty or city.

21. As a patient I should be able to view my details along with my medical history and prescriptions by the doctor.
22. As a patient I should be able to register for organ donation.
23. As a patient I should be able to search organ by the organ name.
24. As a patient I should be able to provide feedback.
25. As a patient I should be able to deregister for organ donation.
26. As a user you can you can access the home page of the website along with its services contact us and about the team.

## **1.4 METHODOLOGY**

### **What does 'scrum' mean**

The word is borrowed from 'Scrumming' in Rugby, which is the process of a team trying to get the possession of ball, that is thrown into the tunnel formed between two teams.

Each team member in the 'Scrum' formation has an important role to play

- Has to work as a closely knit formation.
- Has to put all their strength and skill into play and get the ball to a team member outside the formation.

### **Scrum methodology in agile**

Scrum is an iterative and incremental agile software development framework using a team structure which

- Is self-organizing
- Works as a single unit
- Looks for and responds to changes on a daily basis
- Creates deployable working product components as soon as possible

Each iteration delivers one or more working components of the product. The cycle repeats until

All the product components are deployed or budget is depleted or the deadline arrives

### **Advantages of scrum methodology**

- No matter which impetus stops work, scrum ensures that the most valuable work has been completed when the project ends.
- Eliminates delays in communications and decision making.



- Every team member is equally significant and has visibility into the work status over individuals and overall project.
- It ensures effective use of time, money and people.

## **RULES OF SCRUM**

### **Complete Transparency**

- Significant aspects of the process must be visible to all members of the scrum team.

### **Diligent Inspection**

- Since the requirements are usually incomplete and ambiguous to start with, it is very important that the product is diligently inspected and feedback is provided.
- Inspections should not hamper the progress of the work. They have to be time limited and happen only at the respective events.
- Inspections must be performed by skilled inspectors.

We know it's the nature of agile model that changes appear anytime in between. Therefore, scrum artifacts must be inspected by the inspectors for errors keeping in mind these inspections don't get in the way of the work.

### **Adaptation**

If the end user/inspector determines that one or more aspects of a product is deviating outside acceptable limits, team and the processes should be flexible to make adjustments as soon as possible. During agile projects, the changes may come anytime so the teams should be adaptive enough to make the changes in the under construction system, and may be the approach they are following avoiding any deviation.

### **Events**

Scrum methodology prescribes a set of events which are used to

- Create regularity in team collaboration
- Minimize the need for other meetings

All events are time boxed. This means that every event has a preset maximum duration, as agreed at the beginning of the project, and cannot exceed that limit under any circumstance.

All events must be conducted to ensure successful completion of the project.

### **The sprint**

At the heart of scrum methodology is the Sprint – a single iteration of product development cycle.

- It must have a time of one month or less.
- At its end, a sprint must deliver a “done”, usable and potentially releasable part of the product. This is called as the **sprint goal**.
- A new sprint starts immediately after the conclusion of one sprint. Sprints are usually sequenced and grouped as **releases** of the product.

The Sprint acts as a container for all other events that are described further. Each such event within a sprint is a formal opportunity to diligently inspect the chosen aspect of the project, and adapt, as necessary.

### **Sprint planning**

Sprint planning meeting happens at the beginning of each sprint.

- It must have a time box of 8 hours or less
- At its end, each scrum team member must have discussed, analyzed and agreed upon the answers to the following questions
  - What components of the product can be delivered in the current sprint? – The Sprint Goal
  - How will the chosen work get done? – The Sprint Plan

### **Daily scrum**

Daily scrum meetings happen every day of the sprint at the beginning of each sprint.

- It must happen at the same place and same time every day to eliminate confusions.
- It must have a time of 15 minutes or less
- At its end, every team member must have the answers for the following questions
  - What was accomplished/not accomplished since last scrum meeting?
  - What is planned to be accomplished in the next 24 hours?
  - What are the road blocks foreseen for the next 24 hours and to the sprint goal, in general?

### **Sprint review**

Sprint Review meeting happens at the end of each sprint after all the product development work is stopped for that particular sprint.

- It must have a time of 4 hours or less
- It entirely consists of product inspection and feedback elicitation tasks to determine
  - Whether the sprint goal was achieved?
  - What are the deviations, if any, from the scrum goal?

- Will the current status of the delivered product components elicit changes to the planned components of later sprints?

### **Sprint retrospective**

Sprint retrospective meeting happens at the end of each sprint after the Sprint review meeting. With the Sprint review, we inspect the product, the sprint retrospective inspects all the non-product aspects of the concluded sprint – people, relationships, processes and tools

- It must have a time box of 3 hours or less
- It is an exercise conducted to elicit feedback from the team on
  - Which aspects of the project operated well?
  - Which aspects of the project need improvements?
  - How to implement the improvements for the next sprint?

### **Artifacts**

The number and content of artifacts defined by scrum methodology are specifically designed to

- Maximize transparency of key information so that everybody has the same understanding of the artifact
- Eliminate the confusion and delays due to excess documentation

The only scrum artifacts are -

### **Product backlog**

- It is an ordered list of everything that might be needed in the product - features, functions, requirements, enhancements and even fixes that are required in future.
- Individual items are ordered based on clarity of requirements. Higher ordered product backlog items are usually clearer and more detailed than lower ordered ones.
- It is used by all the members of the scrum team and considered as the only source of requirements.
- Each item must have sequence number, description, effort estimate and implementation cost assigned.
- It is a living document. The product backlog exists as long as the product exists.
- Apart from regular updates at the end of sprints, the product backlog keeps on getting updated as the product is used, gains value, gets market feedback, business environments change, etc.
- At any time in the project, the total work remaining in product backlog can be summed.

### **Sprint backlog**

- The sprint backlog is the set of product backlog items selected for a particular Sprint.

- Sprint backlog is also used document the tasks required to achieve each product backlog and who is supposed to execute that task.
- It is also a dynamic document which can change during daily scrum meeting as required. However, it exists only till the end of the corresponding sprint.
- It is visible to the entire scrum team. However, it solely belongs only to the team members who are directly involved in building the product. Only they can make the updates.
- At any point of time in a sprint, the total work remaining in the sprint backlog can be summed.

### **Increment**

- The increment is the sum of all the product backlog items completed in every sprint.
- It is not a document, but the product in itself.
- At the end of a sprint, the new increment must be “Done,” which means it must be in usable condition.
- The increment keeps increasing in size after every sprint.
- It exists only till the scrum project exists.

### **TEAM STRUCTURE**

#### **Characteristics of a scrum team**

**Self-organizing** – The team chooses how best to accomplish their work rather than being directed by others outside the team.

**Cross functional** – Teams have all competencies needed to accomplish the work without depending on others not part of the team.

#### **Roles in a scrum team**

1. Product owner
2. Development team
3. Scrum master
4. Scrum mentor (This is an optional member, who is an expert in Scrum methodology, included when the entire team is new to working in a scrum project.)

The product owner is the representative of the end users or client who need the product that is being built by the scrum team. To the non-scrum team members, he/she is sole contact of the scrum team if they want to communicate any updates to the product backlog or the scrum team in general.

## **Responsibilities of a product owner**

The product owner is responsible for maximizing the value of the product and the work of the development team by -

- Creating the product backlog by documenting the set of end user requirements, called User Stories, and their priorities clearly.
- Ensuring that the product backlog is accessible, transparent, clear and shows what the Scrum Team will work on next.
- Controlling (authorizing/rejecting) the updates to the product backlog and its contents from both Inside and outside the scrum team.
- Deciding the scope of each sprint.
- Inspecting the product during scrum events, recognize deviations to sprint goals and provide immediate feedback.
- Keeping a track of the overall work remaining at all times.

The Product Owner is the only authorized person who can cancel a sprint, if the sprint goal becomes obsolete.

The Development Team consists of scrum team members who do the work of delivering a potentially releasable increment of “Done” product at the end of each sprint. Only they possess the technical skill set required to create the increment. Individual members may have specialized skills and areas of focus, but accountability of the increment belongs to the development team as a whole.

## **Responsibilities of the development team**

- Only authority to provide effort estimates of the items in the product backlog.
- Negotiate the sprint goal with the product owner.
- Create and maintain the sprint backlog.
- Designing, coding and testing the increment on a daily basis.

The Product owner and scrum master roles are not considered as development team unless they are also executing the work of the sprint backlog.

## **Generic responsibilities of a scrum master**

- Scrum Master is the moderator/facilitator/time keeper for all Scrum Events. Whenever the stipulated time box ends, it is his/her responsibility to make sure that the event is closed immaterial of its completion status.
- It is also Scrum Masters' job to encourage interactions that add value and restrict the ones that are unproductive in a scrum event.
- Scrum Master acts as subject matter expert on scrum. Hence he is the point of contact for the scrum team when they need clarification on processes.

- Scrum Master trains the new members of the team on the Scrum methodology.

### Services Offered by scrum master to product owner

- Provide assistance and suggestions on effective product backlog management
- Ensure the development team do not deviate from product backlog without consent from the Product owner.

### Services offered by scrum master to the development team

- Moderate negotiations or changes to Scrum Goal, with the product owner, when development constraint demands such a change.
- Inspect sprint backlog to check its effective usage.

### Stages

Now let's look at how a Scrum project gets planned and executed. While doing so we will also look at what you, as a tester, will be responsible to do in the stages that require tester's involvement.

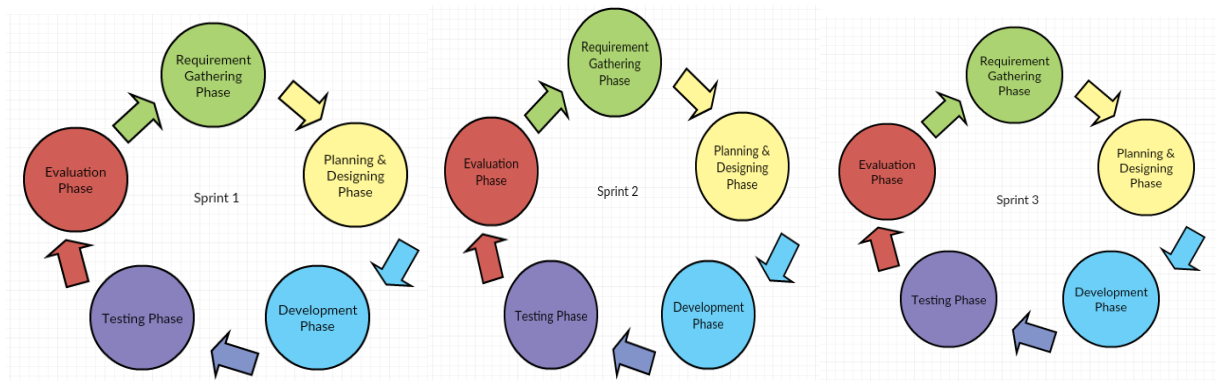


Figure 1 Agile Model

## 1.5 ORGANISATION

**Chapter 1** Highlights the importance of the project in creating an efficient doctor-patient portal system. It clearly states the problem statement and objective of the project. Then it defines the methodology used to build this project. So this project helps to build a smooth relation between doctors and patients.

**Chapter 2** Explains about the skills required to make the project.

**Chapter 3** Describes about the system development which is one of the prime aspect of the project. It, in detail, explains about the development model, class diagrams, use case diagrams, hardware and software requirements, algorithms required to explain the flow of actions.

**Chapter 4** explains performance analysis along with methods as well as results which are displayed for the complete analysis.

Finally, **chapter 5** concludes everything we did in the project as well as any further scope if it has.

# CHAPTER 2 - LITERATURE SURVEY

## 2.1 HTML

- The text which is displayed on a computer and it referring to other image or text which the user can use is hypertext
- The set of tags that helps the web browser in showing text in a document is markup language.

HTML can help in:

- Publishing documents with text, headlines, images etc.
- Creating forms in order to collect user data
- Including videos, audio clips, flash movies etc. in an HTML document
- Accessing online information using hyperlinks

Structure of a HTML page:

```
<!DOCTYPE html>  
<html>  
  <head>  
  
  </head>  
  <body>  
  
  </body>  
</html>
```

<! DOCTYPE html>	DTD (Document type definition) Instructing web browser about the version in which the page has been written
<html>	It comprises of various html elements
<head>	It consists of head elements and also the metadata.
<body>	It consists of all the data of an HTML



	document
--	----------

HTML element consists of both the start tag (<body>) and end tag(</body>). The html element where end tag is missing is said to be an empty HTML element. HTML elements present in body can be broken down into two parts “block-level” and “inline”.

Inline element: It can be specified anywhere in the line. It can contain data and also other inline elements. For example: <span>

Block element: It can only be specified at the beginning of a new line. It can contain block level elements and inline element. For example: <div>

## 2.2 CSS

CSS (Cascading Style Sheets) is a design language which helps in simplifying the process of making presentable web pages. It helps in controlling the text color, font style, background images, background colors. It also has to ability in controlling the way in which an HTML document is presented.

Key Features:

- CSS helps in saving time by reusing the same sheet for various HTML pages. The style for every HTML element needs to be defined once and can be applied to a number of web pages.
- While using CSS the rule for a tag is written once and then applied for all the occurrences for that tag. This helps in reducing the size of the code which leads to faster loading of pages.
- Whenever a universal change is to be made only the style needs to be changed and elements present in all of the web pages will be automatically updated.
- CSS stores the web applications locally using an offline cache. This helps in viewing the websites offline.

3 ways of adding CSS to html:

- In Inline CSS the style is defined and applied using style attribute.

```
<head>
</head>
<body>
<element style="Enter CSS property here">This is Inline CSS</element>
</body>
```

- In Embedded CSS styles are specified in the style element and then embedded in the head element.

```
<head>
<style>
<!-- Style definition will go here -->
</style>
</head>
<body>
</body>
```

- In External CSS the style is specified in an external file using the extension .css.

```
<head>
<link rel="stylesheet" href="path_to_the_css_file" type="text/css" />
</head>
<body>
</body>
```

## 2.3 Bootstrap

Responsive Web Design aims for the creation of web sites that adjust themselves automatically according to the device to look good. Bootstrap helps in creating RWD. This way the appearance of the website remains uniform on different browsers. It helps in making web page styling and designing easier.

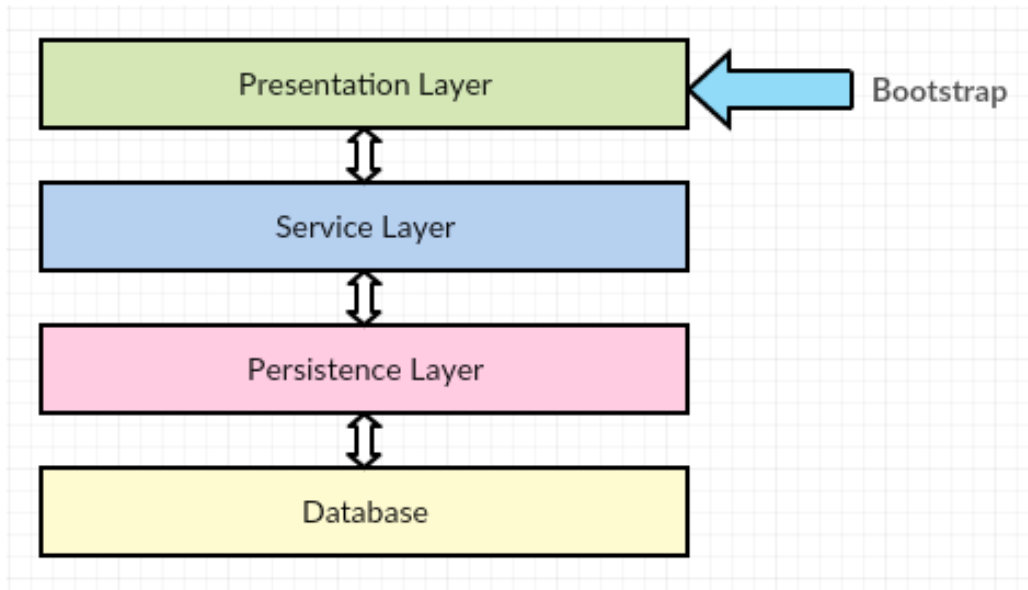


Figure 2 Bootstrap Usage

Bootstrap.css file is added in the HTML page just the way external css files are added.

The first step is to add container classes. A container element is required by bootstrap in order to hold the page contents for proper alignment.

**.container** : fixed width responsive container

**.container-fluid** : full width responsive container, utilizing entire width of the browser

Bootstrap provides the below set of classes to style forms:

1. Form Group class (**form-group**)

This class is applied to UI components of the form.

2. Form Layouts

Form layout classes are applied to <form> tag that holds the UI components.

In order to have consistent look and feel, and optimum spacing between the form elements, across various resolutions, we can make use of **form-group**. Every <label> and its corresponding form element should be wrapped in a <div> having **form-group**. Every form input element (except button) should have class form-control. Bootstrap further simplifies the

process of aligning the form elements (labels, input fields, buttons etc.) by providing **three types of form layouts**.

They are as follows:

- Vertical Form Layout
- Horizontal Form Layout
- Inline Form Layout

Some of the bootstrap components:

Navigation bar (**navbar**) helps user in browsing through various webpages of the website. HTML5 contains a <nav> element for the creation of the navigation bar. Bootstrap consists of various classes that help in enhancing the look of the navbar.

Some classes which can be used to enhance the navigation bar

- . navbar (creates a navigation bar)
- . navbar-default (creates light grey navigation bar)
- . navbar-toggle (styles toggle button which opens navbar on small screens)
- . navbar-collapse (collapses the navbar)

Glyphicons consists of monochromatic symbols and icons which are created with the emphasis to simplicity. This enhances the user experience on the website. The corresponding class of the glyphicon needs to be added in the <span>. Example: <span class=" glyphicon glyphicon-lock blue"> </span>.

Carousel is a slideshow which helps in cycling through the series of content. It can be used for text, images or custom markup. It consists of support for the indicators and next/previous controls.

Some of the carousel attributes:

- .carousel (helps in creating a carousel)
- .slide (helps in adding animation effect for sliding from one item to another)
- .item (specifies content of each slide)
- .carousel-indicators (helps in adding indicators for the carousel)

## 2.4 AngularJS

AngularJS helps in providing the structural framework in order to build dynamic web applications. It gives the liberty to make use of HTML as a template language and extend HTML's syntax to showcase the application's components precisely. It provides efficient binding of data.

Some of the AngularJS directives:

ng-app	Helps in linking and defining the AngularJS application to HTML
ng-model	Helps in binding the values of AngularJS application data to HTML input controls
ng-bind	Helps in binding AngularJS application data to the tags of HTML
ng-init	Helps in initializing the data of the application
ng-repeat	Helps in repeating html elements for every item in a collection

### ng-app

This directive sets off the AngularJS application. It helps in specifying the root element. It helps in initializing or bootstrapping the application when the webpage consisting of the AngularJS application is loaded. It also helps in loading the various modules in the application. In the below example a default AngularJS application is defined using ng-app attribute of a div element.

```
<div ng-app = "">  
...  
</div>
```

### ng-init

This directive aids in initializing the data for the application. It helps in putting values to the variables which are being used in the application. In the given example an array of countries has been initialized using this directive.

```
<div ng-app = "" ng-init = "countries =
[{{locale:'en-US',name:'United States'},
{locale:'en-GB',name:'United
Kingdom'}, {locale:'en-
FR',name:'France'}}">
.....
</div>
```

## ng-bind

This directive helps in binding the values of AngularJS application data to the HTML input controls. In the given example, a model named “name” is defined.

```
<div ng-app = "">
...
<p>Enter your Name: <input type =
"text" ng-model = "name"></p>
</div>
```

## ng-repeat

This directive helps in repeating html elements for every item in collection. In the example given below iteration has been done on the array of countries.

```
<div ng-app = "">
...
<p>List of Countries with locale:</p>
<ol>
  <li ng-repeat = "country in countries">
    {{ 'Country: ' + country.name + ',
Locale: ' + country.locale }}
  </li>
</ol>
</div>
```

## 2.4.1 Modules, scopes and controllers

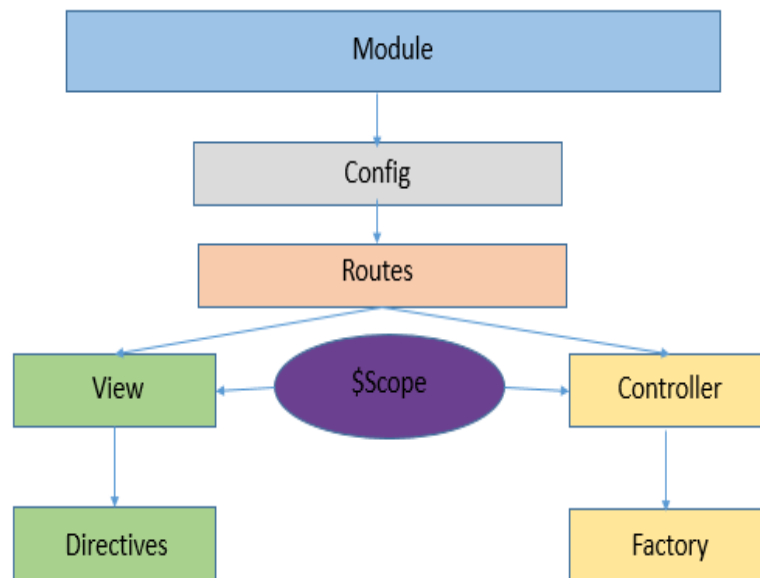


Figure 3 Module, Scope, Controller Relation

### Modules

Module act as a container for different parts of the web app-services, filters, directives, controllers, etc. Applications contains a main method that instantiates and binds the different parts of the application together. Unlike other applications Angular JS doesn't have a main method. Rather modules are specified in such a way that how an application should be bootstrapped. There are enormous benefits to this approach:

- Declarative processes are easier to comprehend.
- Code can be packaged as reusable modules.
- Since, Modules delay execution order of loading module is random.
- Unit tests only loads relevant modules, which keep them fast.
- To override configuration, end to end tests use modules.

## Scopes

Scope is an object that refers to the application model. Scopes copies the DOM structure of the application. Scope propagate events. Scope are able to watch expressions. Few characteristics of scope are as follows:

- \$watch- Scope provide APIs to scrutinize model mutations.
- \$apply- Scope provide APIs to disseminate any model changes through the system.
- Scopes supply context against which expressions are calculated.
- Scopes are used to limit access to the properties of web app components while supplying admission to shared properties.

## Controllers

In AngularJS, a controller is defined by a JS constructor function that is utilized to augment the Angular JS scope. Controller are joined to the document object model in several ways. For every way, AngularJS will start a new object, using the defined controller's constructor junction.

- ng Controller- A new child scope is generated and is made available as an injectable parameter to controller.
- Route controller- It is used in \$route definition.
- The controller of component or regular directive.

### 2.4.2 Built in filters in AngularJS

There are various in built filters in AngularJS such as:

- Filter- Chooses a subset of things from exhibit and returns it as another cluster.
- Currency- Arranges a number as a money (i.e. \$2,897.68). At the point when no cash image is given, default image for current district is utilized.
- Number- Configures a number as content.
- Date- Configures date to a string in light of the asked for organize.
- JSON- Enables you to change over a JavaScript question into JSON string.
- Lowercase- It converts any string to lowercase.



- Uppercase- It converts any string to uppercase.
- Order By- Returns an exhibit that contain things from the predefined gathering, requested by a comparator work in light of the qualities registered utilizing the articulation predicate.

### 2.4.3 Directives and their Corresponding Scopes

Directives follow the scope hierarchy. At the application level, ng-app directive created a scope called as \$rootScope. But some directives in angular like ng-repeat, ng-if, ng-view will create a new scope, which will be added as a child to their parent's scope.

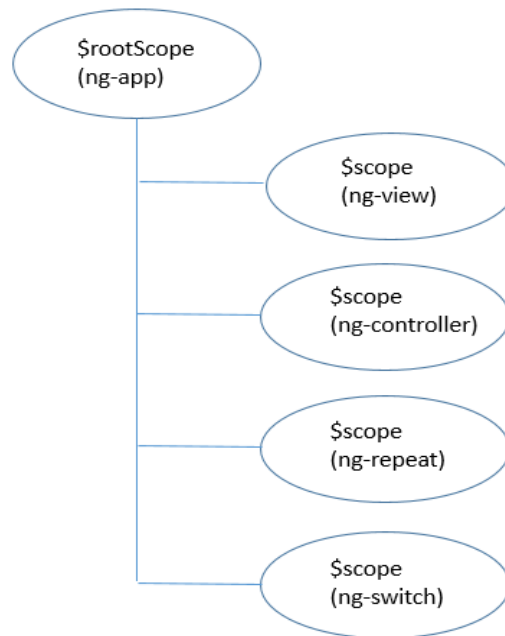


Figure 4 Scope Hierarchy

### 2.5 Java Service Layer:

The above discussions complete our front end technologies used for the application but, if we consider the backend, then we need to check for java code that we have used for the development. The html, CSS, Boot Strap make the front end better while JSPs and Angular make the dynamic content available for us. But, this dynamic content is generated from the back end only i.e. the database. The front end validations are mandatory and have been included but we

also need to perform such operations in the back end. Also, we need to have an interface between the database and the front end. It is the service layer of the Java Enterprise Application and DAO layer of Java Enterprise Application that are responsible for actual interfacing between both of them.

## **2.6 Java DAO Layer**

Once the object of DAO layer is used to call the function in the DAO implementation class, the function needs to be defined here. It is the DAO layer where the sessions are created and data is saved to or retrieved from the databases directly. But, the data received here was in the form of objects of bean class but we cannot insert the bean class object directly to the tables present in our database. So, we construct Entity classes that have been mapped with the tables that we are using to store our data. So, in persistence tier, i.e. the DAO layer we basically create entity class objects, store the whole data into their data fields and it is these objects that are used to write data to the tables, thereby making our input and output possible. In the DAO layer, we can use Hibernate and create queries and basically perform all the operations that are needed to be performed for the project.

## **2.7 Hibernate**

Using Hibernate within a java application, we can store and retrieve data from/to a database. Such activities where a program needs to associate with outer situations like records, database, I/O gadgets, and so forth are called as Input/output (I/O) tasks. In an I/O task source and goal can be anything that holds, creates or devours information, e.g. document framework, application program, fringe gadgets like console, comfort, and so forth. Hibernate is an open source ORM execution for Java engineers which takes after JPA measures.

The mapping of the Entity class with the tables is the part of Hibernate only. The primary key column is mapped with the corresponding instance variable and the other instance variables are mapped with the corresponding columns in the table.

Once the mapping is done and the properties configured all we need to do is, follow some of the steps to perform the persist operation. Initially, we have to create the instance variables of session and session factory and then define the context of the transaction.

After this is done then, we can add the data to the data fields of the entity objects, but this should be done in an open transactions and then the session and session factory can be closed. Transaction interface provides a variety of method that can be used like begin()-to begin a transaction, commit()-to save the changes and rollback()-to undo the changes.

Similarly, if the data has to be retrieved from the database, it could be done in the same way. But, the data can only be retrieved within a session and that too only in entity object. Information can be recovered from the database table as far as comparing substance example by conjuring the get() strategy for Session interface. If we consider the get() method, It restores the occasion of the given substance class with the given identifier esteem. It returns invalid if there is no such record exist in the database. There is another method called the load() method. If we need to delete the details or any record from the database then we have the delete() function. For erasing the client subtle elements of the recovered client (stage 4), you have to summon delete() technique for the Session interface. It acknowledges the protest which should be erased (recovered substance question). The delete() technique ought to be summoned inside a dynamic transaction .The delete() strategy causes the execution of SQL DELETE when you commit() the Hibernate Transaction.

# CHAPTER 3 - SYSTEM DEVELOPMENT AND ALGORITHMS

## 3.1 Hardware Requirements

The minimum requirements needed to perform operations are:

- Intel Pentium Processor at 2Ghz or higher.
- RAM 256MB or more.
- Hard disk capacity 10GB or more.

## 3.2 Software Requirements

The software required to perform the implementations are:

- Windows or Linux Operating system (Ubuntu, Fedora).
- JDK 8 and above.
- Eclipse/ NetBeans IDE
- Visual Paradigm

## 3.3 Software Development model

Agile development model is used for software development. It is a group of software development methods in which the SRS and its solutions evolve through collaboration between cross-functional and self-organizing teams. It encourages robust planning, mutative development, before/on time delivery, continuous improvement scope, and promotes quick and flexible response to change.

Figure 5 describes the methodology of this agile development model whereby each iterations delivers some features.

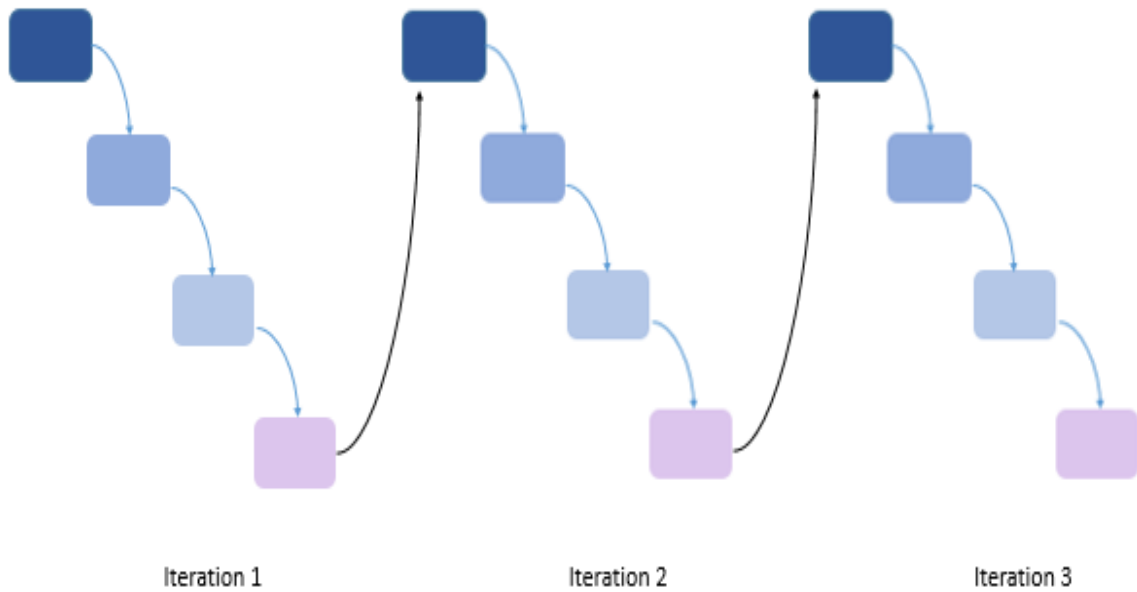


Figure 5 Agile Iterations

### 3.4 Class diagrams

Organdonor	Feedback
private Integer patientId private String organname private Integer orgId	private Integer feedbackId private Integer docId private String feedback private String docName private String message
Getters Setters	Getters Setters

Doctor	Appointment
<pre>private Integer docId private String name private String docCity private String address private String speciality private Long mobileNo private String emailId private String password private String message</pre>	<pre>private Integer appointmentId private String city private Integer patientId private Integer docId private String patientName private String docName private String speciality private Calendar dateOfAppointment private String timeOfAppointment private String message</pre>
<p>Getters Setters</p>	<p>Getters Setters</p>
Patient	Treatment
<pre>private Integer patientId private String name private String address private Long mobileNo private String bloodGroup private String emailId private String password private Calendar dateOfBirth private String message</pre>	<pre>private Integer treatmentId private Integer patientId private String patientName private String problem private String treatmentNote private Calendar dateOfTreatment private String treatment private Integer docId private String docName private String message</pre>
<p>Getters Setters</p>	<p>Getters Setters</p>

### 3.5 Entity Relationship diagram

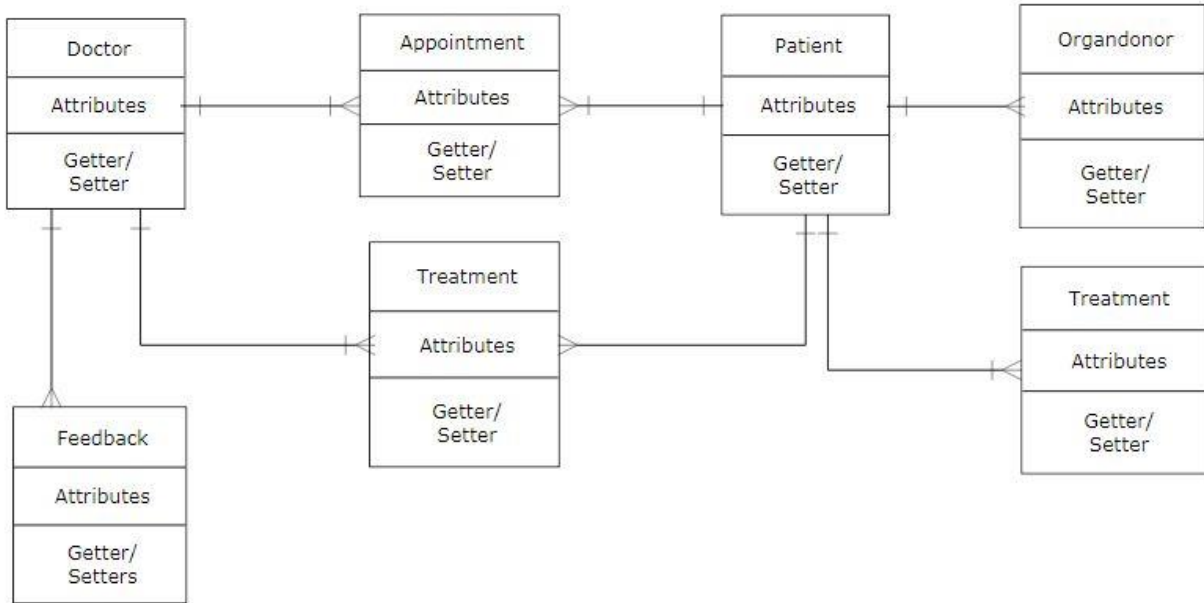
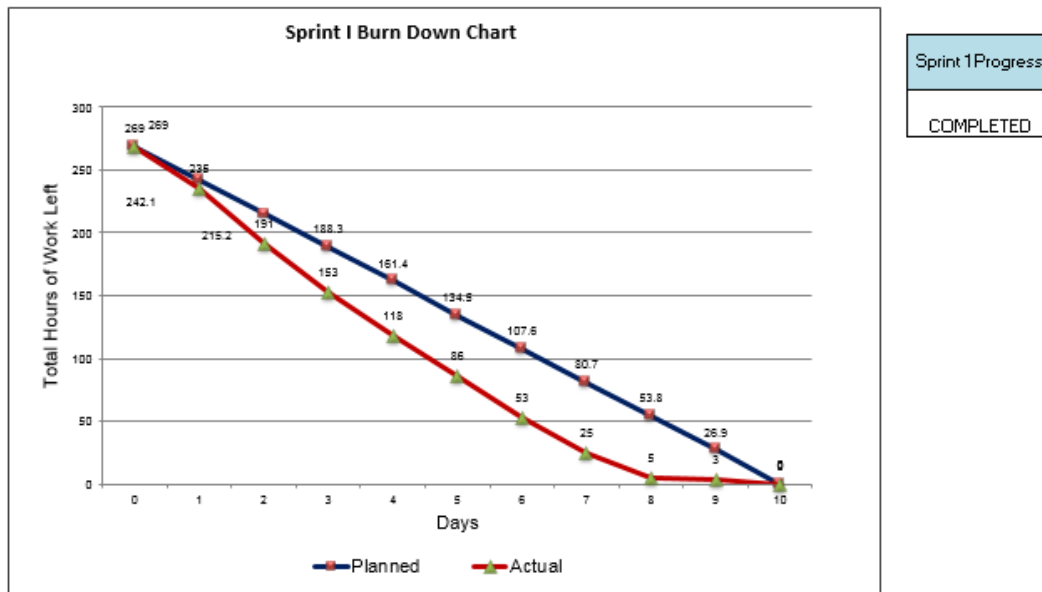


Figure 6 Entity Relationship Diagram

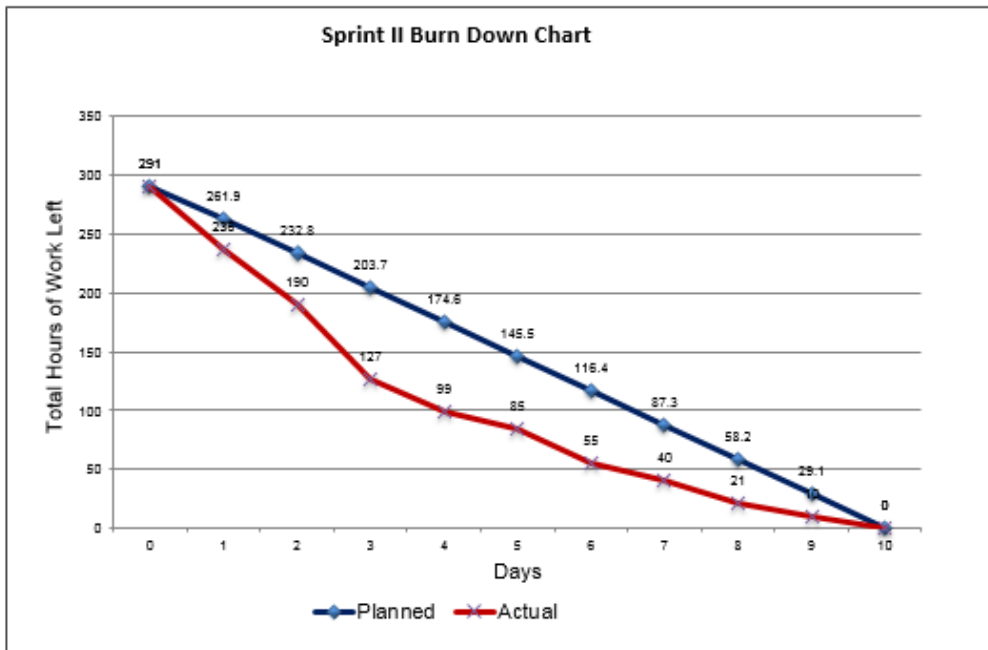
### 3.6 Agile Sheet

#### Burn Down Charts

##### Sprint 1

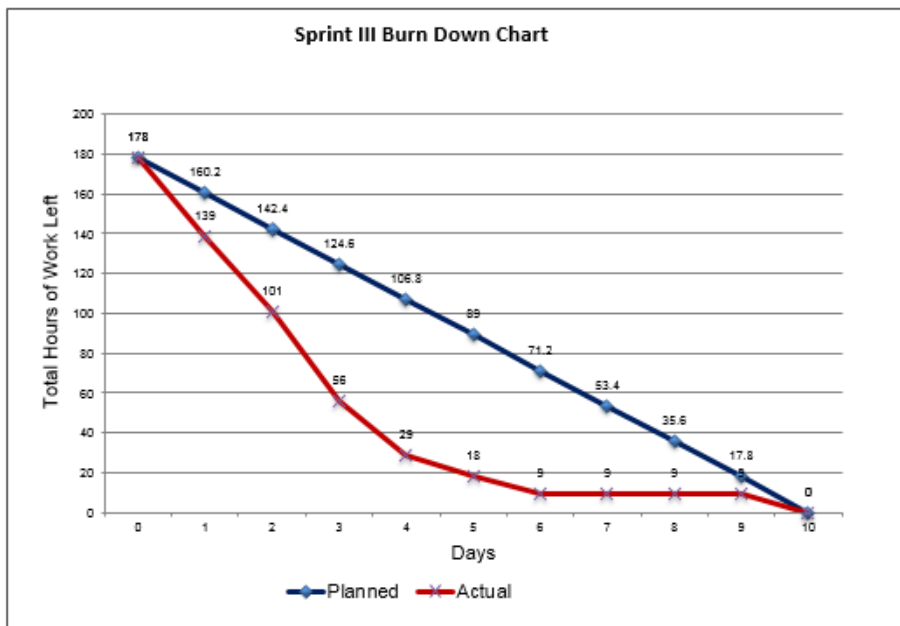


## Sprint 2



Sprint 2 Progress  
COMPLETED

## Sprint 3



Sprint 3 Progress  
COMPLETED



## Product Backlog

	Planned Sprint	Actual Sprint	US ID	User Story Description	MOSCOV	Dependency	Assignee	Status
1								
2	Sprint 1	Sprint 1	US-01	As an admin I should be able to login through an ID and Password	MUST HAVE	ND	Everyone	3- Completed
3	Sprint 1	Sprint 1	US-02	As an admin I should be able to logout	MUST HAVE	US-01	Everyone	3- Completed
4	Sprint 1	Sprint 1	US-03	As an admin I should be able to add Doctors according to their speciality	MUST HAVE	US-01	Everyone	3- Completed
5	Sprint 1	Sprint 1	US-04	As an admin I should be able to view Doctors of all the departments	MUST HAVE	US-01,US-03,US-17	Everyone	3- Completed
6	Sprint 2	Sprint 2	US-05	As an admin I should be able to view Patient details along with their medical history	MUST HAVE	US-01,US-11,US-14	Everyone	3- Completed
7	Sprint 3	Sprint 3	US-06	As an admin I should be able view all the organ donors along with their blood group	SHOULD HAVE	US-01,US-22,US-25	Everyone	3- Completed
8	Sprint 3	Sprint 2	US-07	As an admin I should be able to view all the feedbacks given by the patients	COULD HAVE	US-01,US-24	Everyone	3- Completed
9	Sprint 1	Sprint 1	US-08	As a doctor I should be able to login through an ID and password provided by the admin	MUST HAVE	US-03	Everyone	3- Completed
10	Sprint 2	Sprint 2	US-09	As a doctor I should be able to view my appointments	MUST HAVE	US-08,US-17	Everyone	3- Completed
11	Sprint 2	Sprint 2	US-10	As a doctor I should be able to view patient details and their medical history	MUST HAVE	US-08,US-11	Everyone	3- Completed
12	Sprint 2	Sprint 2	US-11	As a doctor I should be able to add treatment for the patients	MUST HAVE	US-08,US-17	Everyone	3- Completed
13	Sprint 2	Sprint 1	US-12	As a doctor I should be able to view my details	SHOULD HAVE	US-08	Everyone	3- Completed

## Sprint Backlog

US ID	Task Description	Task Start Date	Task Completion Date	Team Member	Activity	Status	Estimate Effort (In Hours)	Day 1	Day 2	Day 3	Day 4	Day 5	Day 6	Day 7	Day 8	Day 9	Day 10
							269	235	191	153	118	86	53	25	5	3	0
<b>SPRINT 1 BACKLOG</b>																	
US-01	Design Of UI	16/04/2018	18/04/2018	Chirag	Design	3- Completed	8	7	4	0	0	0	0	0	0	0	0
	Back-End Design	16/04/2018	18/04/2018	Varun	Design	3- Completed	8	4	4	0	0	0	0	0	0	0	0
	Business Logic Implementation	16/04/2018	21/04/2018	Varun	Coding	3- Completed	10	9	8	7	6	4	2	0	0	0	0
	UI Implementation	16/04/2018	19/04/2018	Arkapravo	Coding	3- Completed	8	6	4	2	0	0	0	0	0	0	0
	Back-End Implementation	16/04/2018	21/04/2018	Rajat	Coding	3- Completed	10	9	8	7	6	5	1	0	0	0	0
	Review	21/04/2018	24/04/2018	Porshiya	Integration Test	3- Completed	4	4	4	4	4	4	4	2	0	0	0
	Unit Testing	21/04/2018	24/04/2018	Ashutosh	Unit Test	3- Completed	2	2	2	2	2	2	2	2	0	0	0
	Documentation	21/04/2018	24/04/2018	Ashutosh	Design	2- In Progress	3	3	3	3	3	3	2	1	0	0	0
	UI Implementation	16/04/2018	19/04/2018	Chirag	Coding	3- Completed	8	3	3	2	1	0	0	0	0	0	0



## Burn Down Chart data

SPRINT I BURN-DOWN CHART DATA				SPRINT II BURN-DOWN CHART DATA				SPRINT III BURN-DOWN CHART DATA			
Sprint	Day	Planned	Actual	Sprint	Day	Planned	Actual	Sprint	Day	Planned	Actual
1	0	269	269	2	0	291	291	3	0	178	178
	1	242.1	235		1	261.9	236		1	160.2	139
	2	215.2	191		2	232.8	190		2	142.4	101
	3	188.3	153		3	203.7	127		3	124.6	56
	4	161.4	118		4	174.6	99		4	106.8	29
	5	134.5	86		5	145.5	85		5	89	18
	6	107.6	53		6	116.4	55		6	71.2	9
	7	80.7	25		7	87.3	40		7	53.4	9
	8	53.8	5		8	58.2	21		8	35.6	9
	9	26.9	3		9	29.1	10		9	17.8	9
	10	0	0		10	0	0		10	0	0

## Product Backlog Grooming

Sprint	US ID	Points Discussed
1	US-01	To add front end validations in password mobile number
1	US-27	To stop redirecting the login page back to the home page after login
1	US-17	While making an appointment only show doctors for the selected city
1	US-27	To add some text or image in both doctor and patient pages after they login to make it more interactive
2	US-24	To add feedback on the same page where patient looks his medical history
2	US-12	To add change password separately as a button while viewing doctor details
2	US-12	Doctors can update their time slots i.e. for which time slots they are available

## 3.7 Algorithms

### 3.7.1 Algorithm for registration of patient

#### Steps:-

- User clicks on the Login Here/Register link and then on Register now link
- A form appears which prompts the user to fill his/her:
  1. Name
  2. Email Address
  3. Password
  4. Hint for password in case the user forgets
- After filling the details click on register which submits the form
- The details go the LoginController accordingly
- The Post method redirects the details to the corresponding method of DoctorPatientAPI

- From API, the service layer method for registration is called Service method calls the PatientDAOImpl class which hits the database and returns a success message which is bind in DoctorPatientAPI

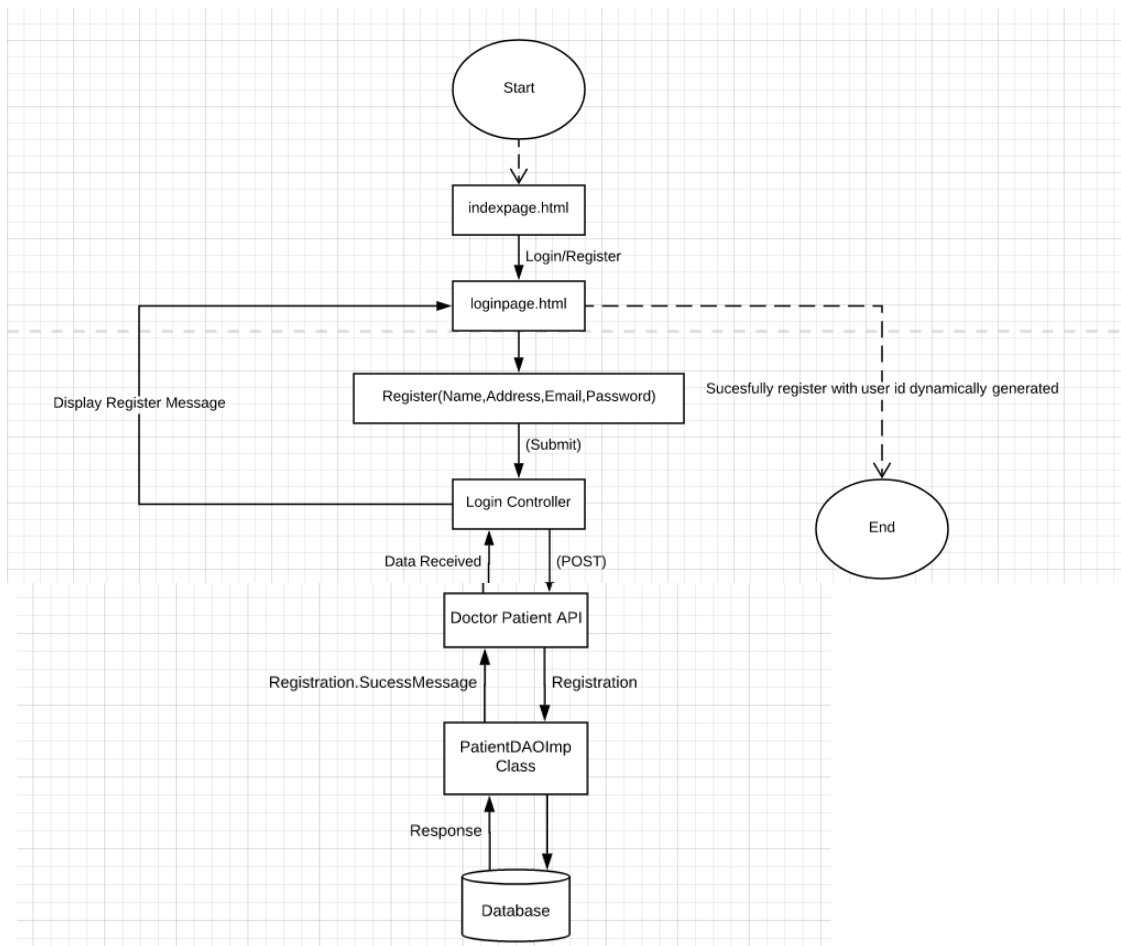


Figure 7 Register Algorithm

### 3.7.2 Algorithm for Login of patient/doctor

#### Steps:-

- User clicks on the Login Here/Register link and then on Register now link
- A form appears which prompts the user to fill his/her:
  1. Email Address
  2. Password
  - After filling the details click on sign in which submits the form
  - The details go the LoginController accordingly
  - The Post method redirects the details to the corresponding method of DoctorPatientAPI
  - From API, the service layer method for login is called Service method calls the PatientDAOImpl class which hits the database and returns a success message which is bind in DoctorPatientAPI

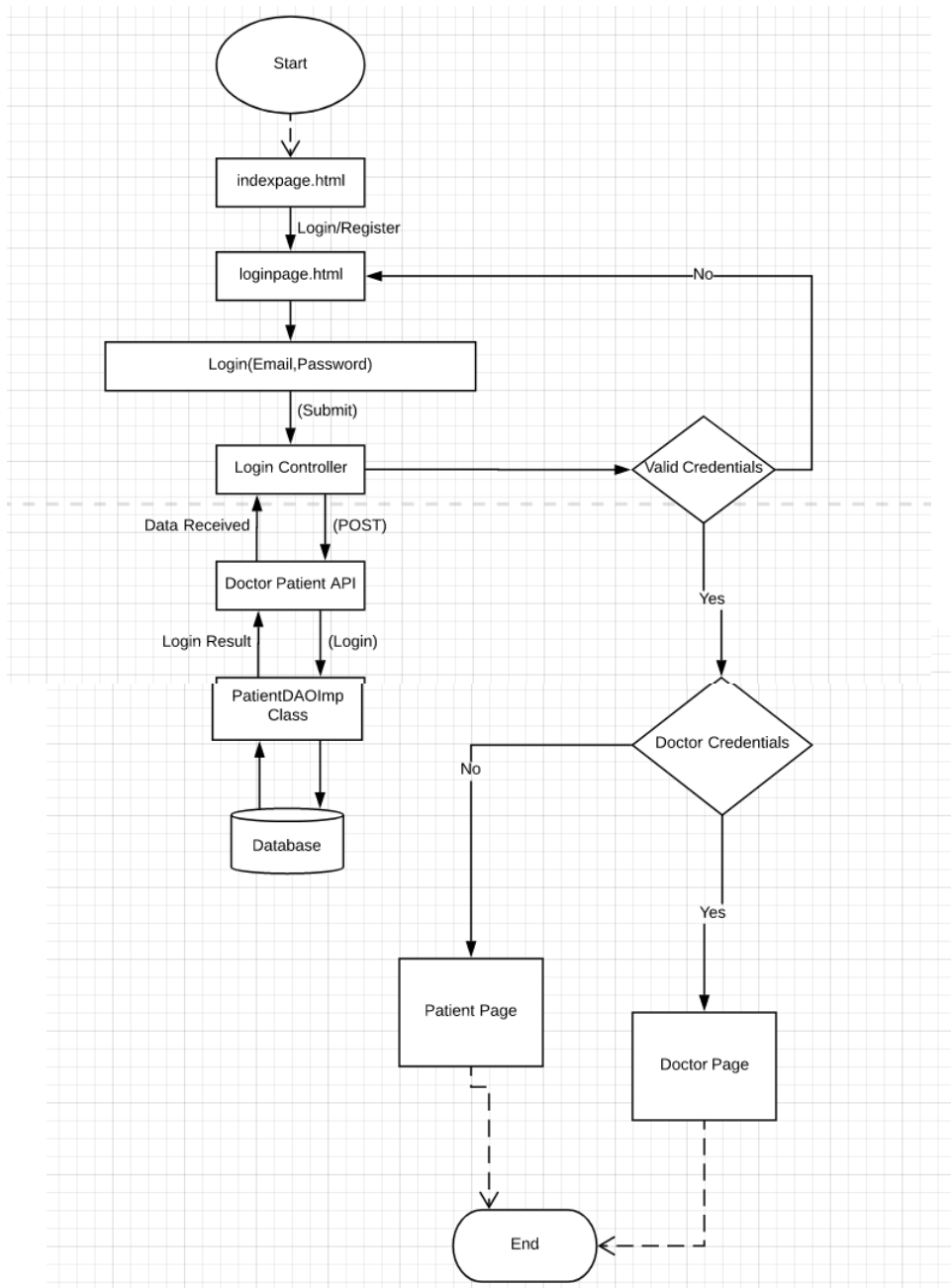


Figure 8 Login Algorithm

### 3.7.3 Algorithm for Booking an Appointment with a doctor

#### Steps:-

- User clicks on the Login Here/Register link and then on Register now link
- A form appears which prompts the user to fill:
  1. Doctor City
  2. Doctor Speciality

3. Doctor Name
  4. Date of appointment
  5. Time for which user need appointment
- After filling the details click on submit button which submits the form
  - The details go the PatientHomeController accordingly
  - The Post method redirects the details to the corresponding method of DoctorPatientAPI
  - From API, the service layer method for booking an appointment is called  
Service method calls the PatientDAOImpl class which hits the database and returns a success message which is bind in DoctorPatientAPI

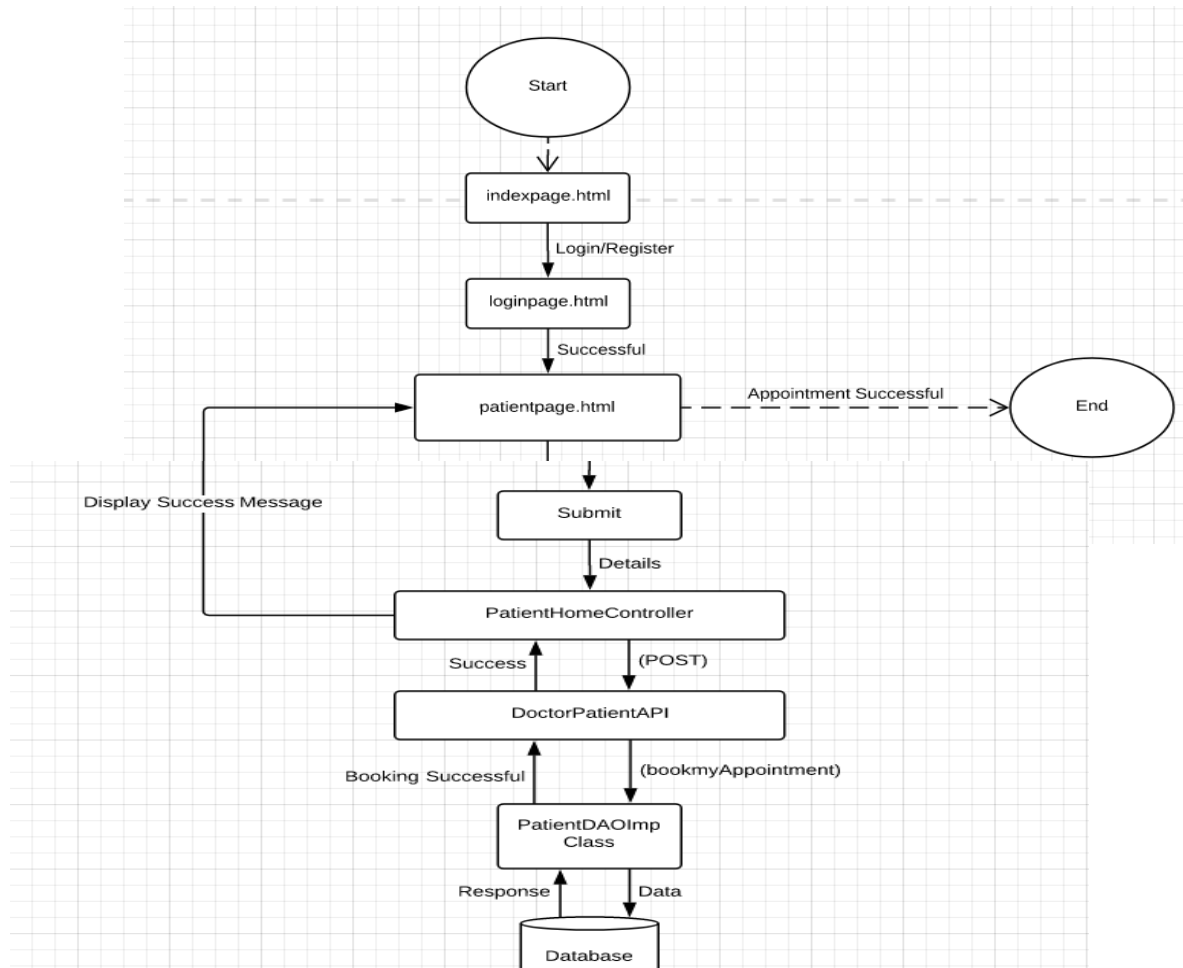


Figure 9 Book Appointment Algorithm

### 3.7.4 Algorithm of Registration for Organ Donation

#### Steps:-

- User clicks on the Login Here/Register link and then on Register now link

- A form appears which prompts the user to select the organ name from the dropdown organ list
- After filling the details click on submit button which submits the form
- The details go the Patient Home Controller accordingly
- The Post method redirects the details to the corresponding method of Doctor Patient API
- From API, the service layer method for organ Registration is called Service method calls the Patient DAOImpl class which hits the database and returns a success message which is bind in Doctor Patient API

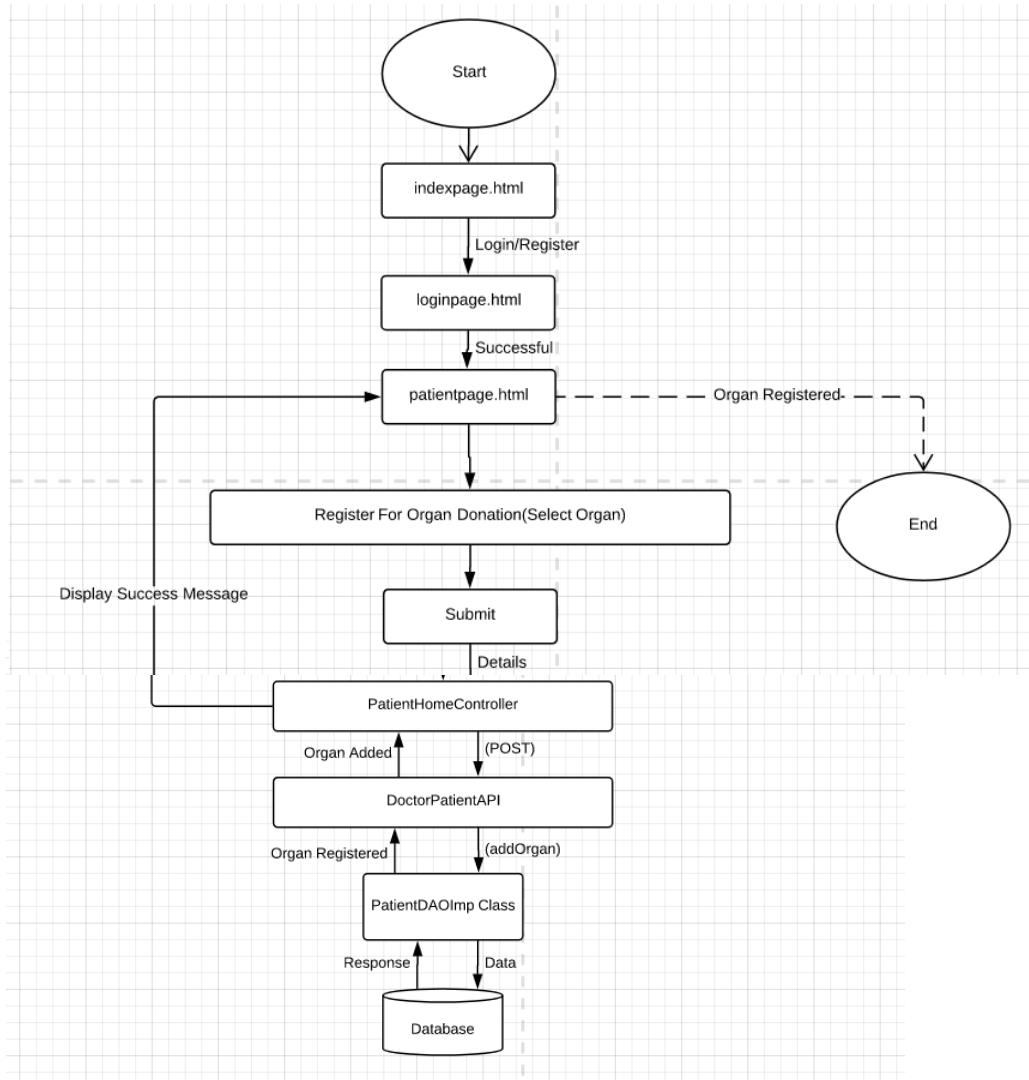


Figure 10 Organ Registration

### 3.7.5 Algorithm for Login of Admin

#### Steps:-

- User clicks on the Login Here/Register link and then on Admin Login link

- A form appears which requires two dummy details
  1. User name
  2. Passsword
- Here we have to use dummy authentication which has username as “test@test” and password is “test” after filling click on submit button which submits the form
- The details go the AdminHomeController accordingly
- The Post method redirects the details to the corresponding method of DoctorPatientAPI
- From API, the service layer method for login is called Service method calls the PatientDAOImpl class which hits the database and returns a success message which is bind in DoctorPatientAPI

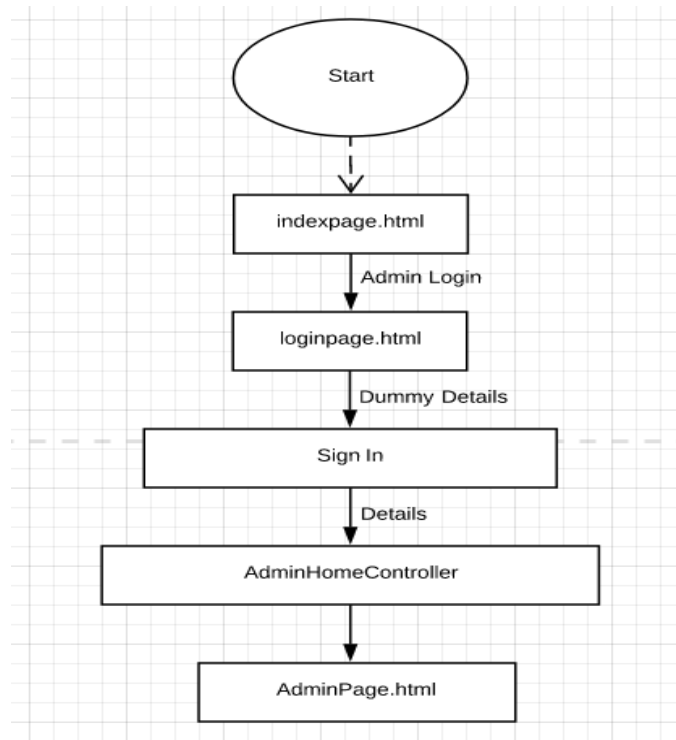


Figure 11 Admin Login

### 3.7.6 Algorithm for Adding a Doctor in database

#### Steps:-

- After following algorithm (4.1.5)
- User clicks on the Add Doctor link
- A form appears which prompts the admin to fill:
  1. Doctor Name
  2. Doctor Address
  3. Doctor City



4. Speciality
  5. MobileNo
  6. Email
  7. Password
- After filling the details click on submit button which submits the form
  - The details go the AdminHomeController accordingly
  - The Post method redirects the details to the corresponding method of DoctorPatientAPI

From API, the service layer method for addDoctor is called

Service method calls the DoctorPatientDAOImpl class which hits the database and returns a success message which is bind in DoctorPatientAPI.

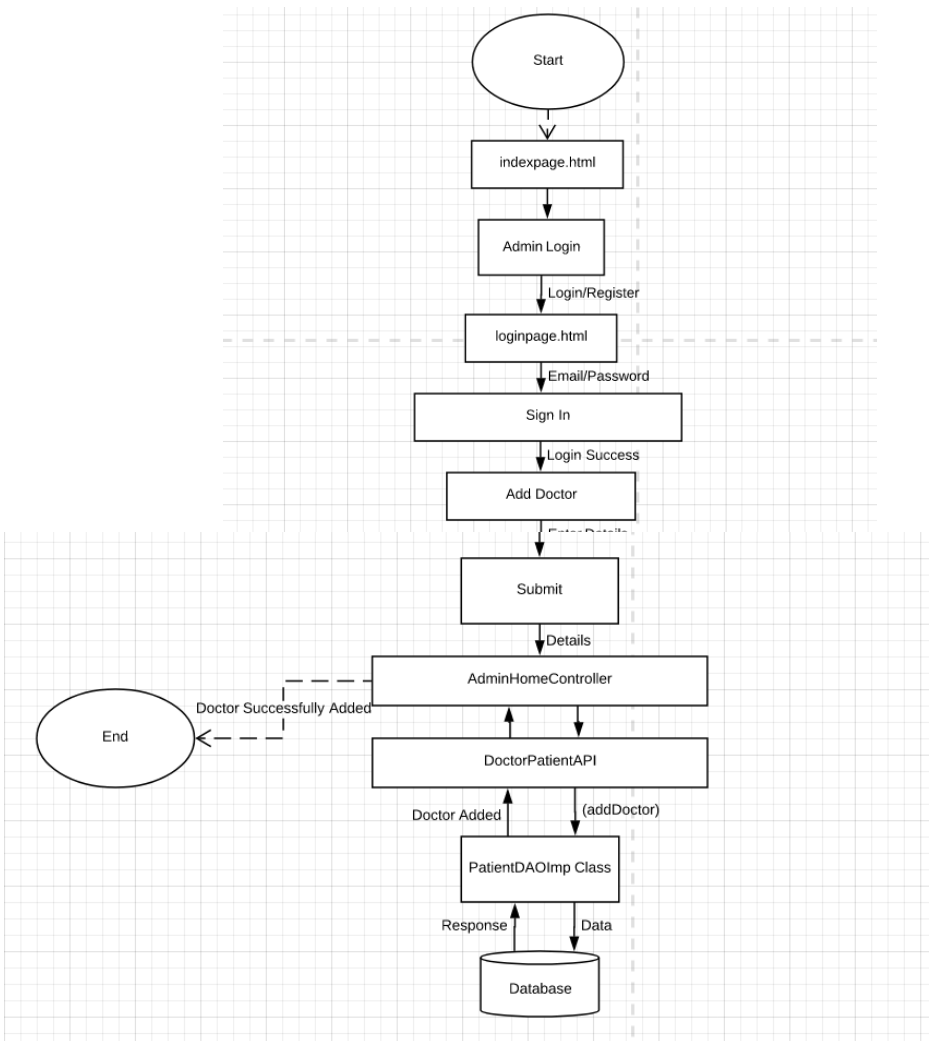
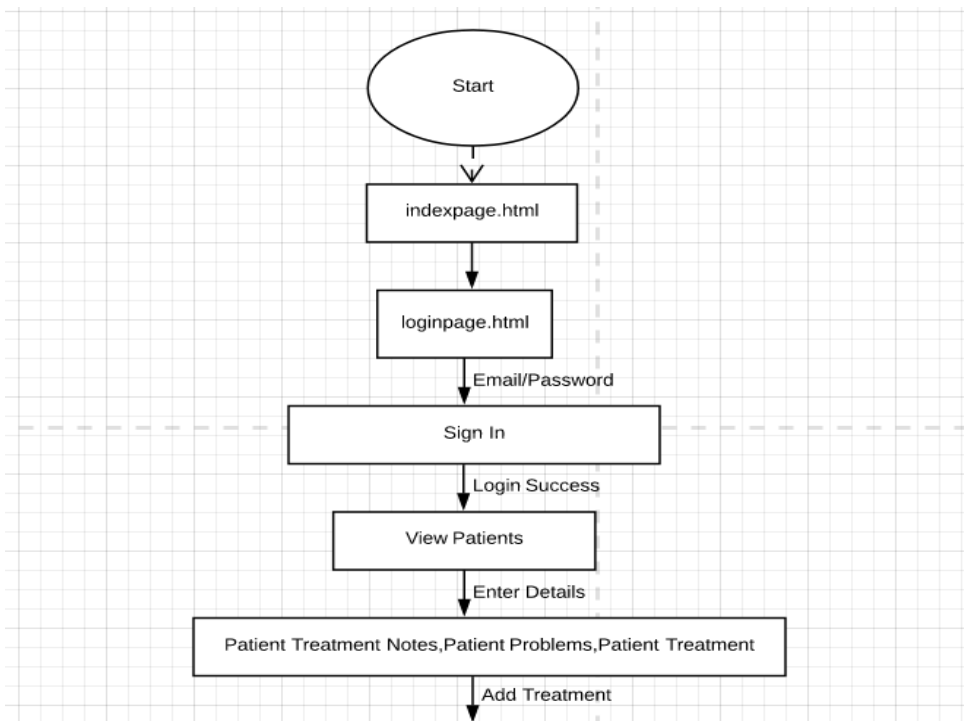


Figure 12 Add Doctor Algorithm

### 3.7.7 Algorithm for Adding a treatment for a patient

#### Steps:-

- After following algorithm (4.1.2)
- User clicks on the View Patient link in which he can view patient details their medical history and can add treatment for the same
- A form appears which prompts the admin to fill:
  1. Patient Treatment Notes
  2. Patient Problem
  3. Patient Treatment
- After filling the details click on Add Treatment button which submits the form
- The details go to the DoctorHomeController accordingly
- The Post method redirects the details to the corresponding method of DoctorPatientAPI  
From API, the service layer method for addTreatment is called  
Service method calls the DoctorDAOImpl class which hits the database and returns a success message which is bind in DoctorPatientAPI



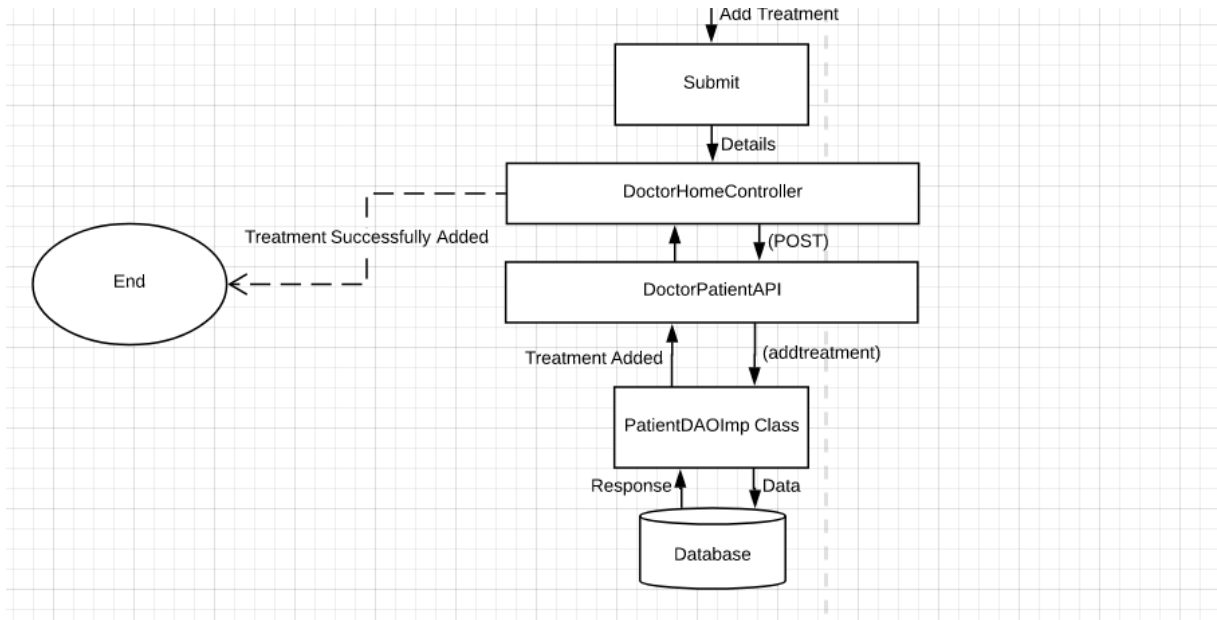


Figure 13 Add Treatment

## CHAPTER 4 - PERFORMANCE ANALYSIS AND RESULT

In this project we used the Scrum flavor of agile methodology that helps us to analyze the project at continuous intervals. There is an analysis at the end of each sprint. We have been successful at achieving our goals at the end of each sprint. This helped to us to rectify our mistakes for the next sprint.

At the end of the first analysis we found the following mistakes:

1. To add front end validations in password mobile number
2. To stop redirecting the login page back to the home page after login
3. While making an appointment only show doctors for the selected city
4. To add some text or image in both doctor and patient pages after they login to make it more interactive

### **Rectifications:**

1. Front end validations added to the registration process using regex.
2. Rectified the links so that the user stays logged in.
3. Added some new functions to back end to classify the doctors based no city and specialty.
4. Made changes in the front end to make the project more interactive.

At the end of the second analysis we found the following mistakes:

1. Make changing password more feasible for users.
2. Two patients should not be able to book an appointment with the doctor for same time.
3. Giving feedback and adding treatment more interactive for the users.

### **Rectifications:**

1. Made changes in back end and front end to increase feasibility for the users.
2. Added unique constraints for time and doctor.
3. Made changes in the front end to make the project more interactive.

This is how our project works:

### **Homepage**



## ABOUT US

We here propose a doctor patient handling, managing system that helps doctors in their work and also patients to book doctor appointments and view medical

## ABOUT US

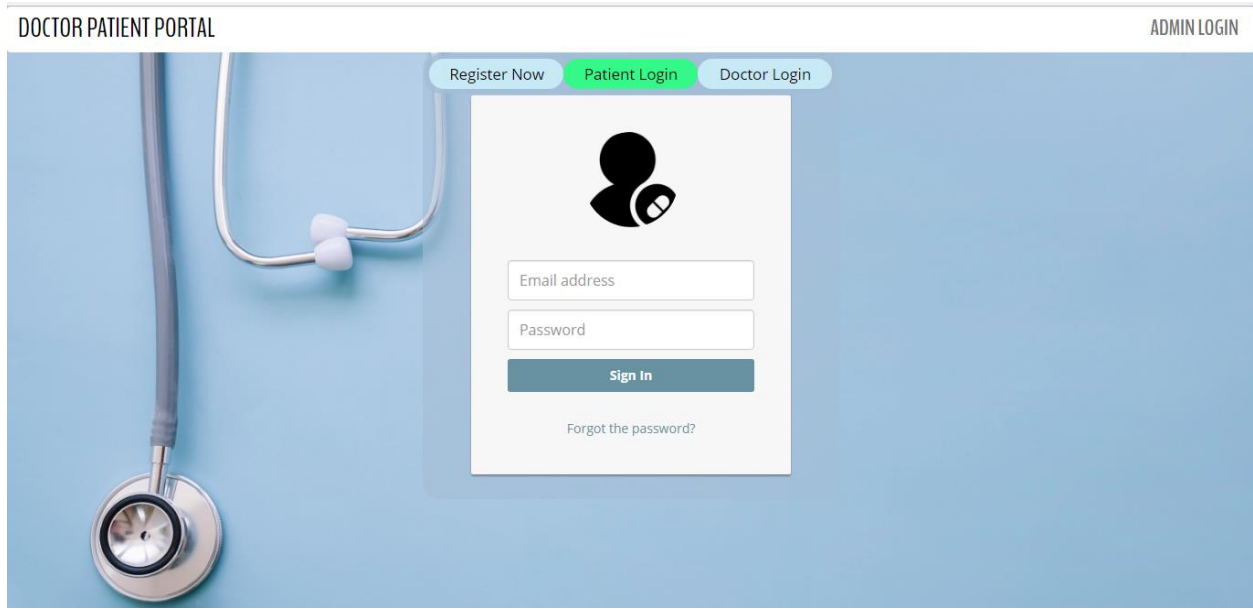
We here propose a doctor patient handling, managing system that helps doctors in their work and also patients to book doctor appointments and view medical progress. The system allows doctors to manage their booking slots online. Patients are allowed to book empty slots online and those slots are reserved in their name. The system manages the appointment data for multiple doctors for various date and times. Each time a user visits a doctor his/her medical entry is stored in the database by doctor. Next time a user logs in he may view his/her entire medical history as and when needed. At the same time a doctor may view patients medical history even bore the patient visits him. This allows for an automated patient doctor handling system through an online interface. Our system also consists of organ donor module. This module allows for organ donation registration as well as organ search. The module is designed to help urgent organ requirements through easy/instant searches.

## OUR SERVICES

We offer patients to make appointments with their doctors and allows them to view their medical history and progress and moreover provide help in urgent organ

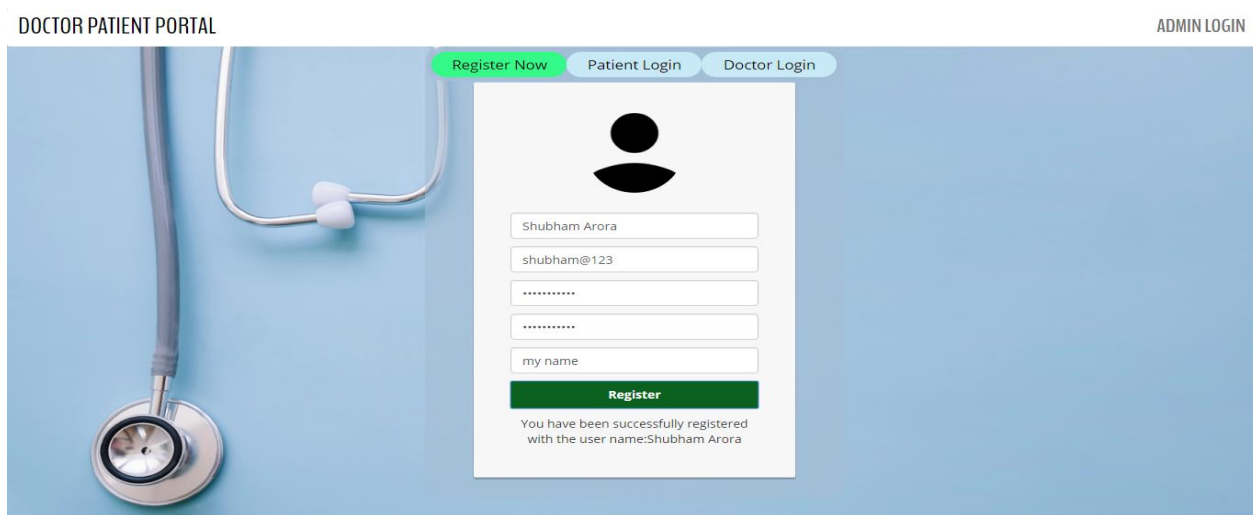
By clicking on login button on the navigation bar user will be directed to the Login page where he can login as a patient, doctor and admin. If he is not registered as a patient, he can register as a patient.

### Login page



Now a user can register by clicking on register now button.

### **Patient registration page**



On successful registration user will get a success message. We have front end validations here. User will get an error message in following conditions:

1. Email does not match the required conditions.
2. Password does not match the required conditions.
3. There is a password mismatch between both password fields.
4. If any required field is not left empty.

After successful registration as a patient user can login with email and password.

After Login a patient can search doctors using the city and specialty filters.

### Search doctors

The screenshot shows a web interface titled "Search Doctors". It contains three dropdown menus for filtering search results:

- Doctor City:** Options include Any City, Mysore, Bangalore, Mangalore, Ooty, Chennai, Hyderabad, and Trivendrum. "Trivendrum" is selected.
- Doctor Speciality:** Options include Any Speciality, Orthopaedic, Psychiatry, Neurology, General Practice, Urology, ENT, and Ophthalmology. "ENT" is selected.
- Doctor Name:** The name "Arkapravo" is displayed and highlighted.

Below the filters, a table displays the details of the selected doctor:

Name:	Arkapravo
Address:	New Market
City:	Trivendrum
Speciality:	ENT
Email:	arka123@arka
Mobile No:	9765476897

Now a patient can also book appointment with the doctors according to their availability.

## Book appointment

### Book An Appointment

Trivendrum

ENT

Arkapravo

05/22/2018

11:00

The slot is currently booked, please select another day/time.

**Submit**

### Book An Appointment

Trivendrum

ENT

Arkapravo

05/22/2018

11:00

You have successfully booked your appointment with Dr.Arkapravo

**Submit**

Patient can also give feedback to the doctors. He can give feedback to only those doctors he has been given treatment by.



## View and cancel appointment

### My Appointments

Appointment Id	Patient Name	Doctor Name	Speciality	Date	Time	Cancellation
205	Shubham Arora	Arkapravo	ENT	Tue/05/2018	1100	<a href="#" style="background-color: #007bff; color: white; padding: 5px 10px; text-decoration: none;">Cancel</a>

## Give Feedback

### My History

Treatment Id	Name	Problem	Treatment Note	Doctor Id	Doctor Name	Date	Feedback
1210	Shubham Arora	continuous bleeding from ear	bed rest required	1210	Arkapravo	Wed 16/05/2018	<a href="#" style="background-color: #007bff; color: white; padding: 5px 10px; text-decoration: none;">Give Feedback</a>

[Submit](#)

Feedback has been added successfully with the id: 21

A patient can update his details and can also change his password.

### My Details

Arkapravo

New Market

ENT

9765476897

arka123@arka

Enter Old password

Enter New Password

Enter New Password Again

Patient is also given the feature to register and deregister for organ donation.

### Organ Registration

### Register For Organ Donation

121

Heart

Organ has succesfully been added.

### Organ Deregistration

### De-register For organ donation

Organ Id	Organ Name	Cancellation
510	Heart	<input type="button" value="Cancel Donation"/>

Now if one is in need of organ he can also search for count of organs available.

### **Search Organs**

Organ List	
Organ Name	Organ Count
Liver	1
Heart	1
Lungs	1

Doctors can login with email and password provided by the admin.

Doctor can use the following functionalities:

### **View Appointments**

My Appointments				
Appointment Id	Patient Id	Patient Name	Appointment Date	Appointment Time
205	121	Shubham Arora	Tue-22/05/2018	1100

## Update Profile and Password

### My Details

## View Patients and their History

### My Patients

Patient Id	Name	Mobile No	Blood Group	Email	Date Of Birth	Add Treatment	View History
121	Shubham Arora	9877664532	O-	shubham@123	16/05/2018	<input type="button" value="Add Treatment"/>	<input type="button" value="View History"/>

Patient Name	Doctor Name	Problem	Treatment	Treatment Notes
Shubham Arora	Arkapravo	continuous bleeding from ear	pills and ointment	bed rest required

By clicking on Add Treatment Button Doctor can add treatment to patient's history.

## Add Treatment

### My Patients

Patient Id	Name	Mobile No	Blood Group	Email	Date Of Birth	Add Treatment	View History
121	Shubham Arora	9877664532	O-	shubham@123	16/05/2018	<a href="#">Add Treatment</a>	<a href="#">View History</a>

Shubham Arora

bed rest required

continuous bleeding from ear

pills and ointment

[Add Treatment](#)

Treatment has been added successfully with the id:1210

Doctor can also view his feedbacks

## View feedback

### View Feedback

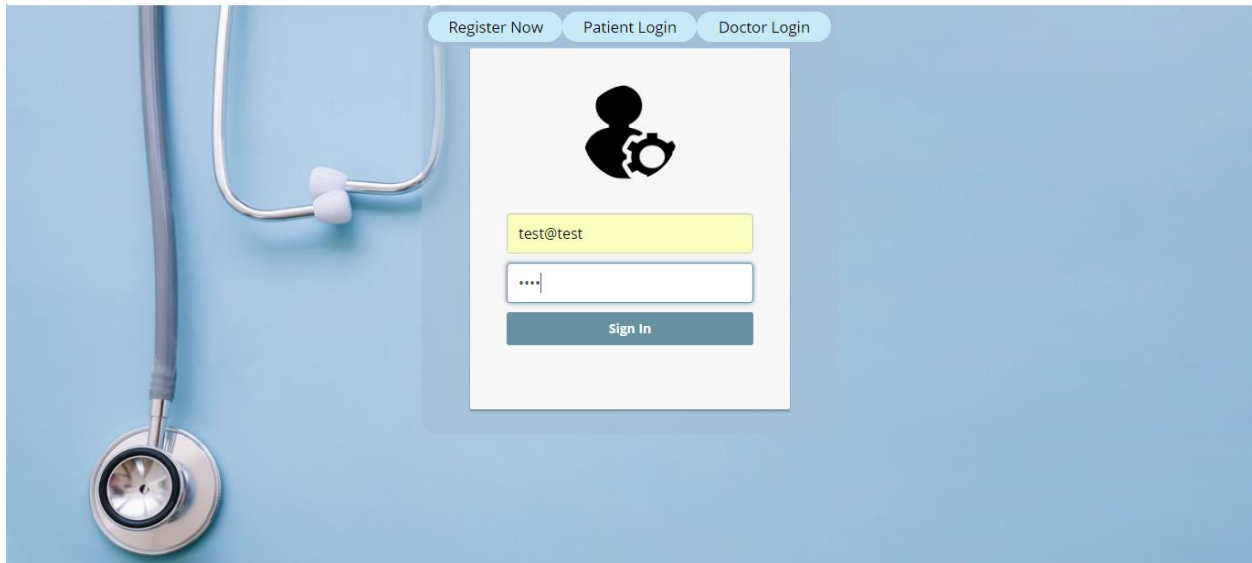
Feedback Id	Feedback
21	very polite with patients

We also have an admin login. Admin is the only one who can add doctors into the system.

## Admin Login

DOCTOR PATIENT PORTAL

ADMIN LOGIN



## Admin front page

DOCTOR PATIENT PORTAL

WELCOME TEST@TEST LOGOUT



## Add Doctor

### Add Doctor

### Add Doctor

Doctor has been added successfully with name:Arkapravo

## View Doctors

Doctor Id	Name	Speciality	Mobile No	Email	Address	City
1010	Rajat Chauhan	Orthopaedic	9805047099	rajat@cc	Solan	Mysore
1020	Kuldeep Gupta	Urology	9805056312	kuldeep@cc	Delhi	Mysore
1030	Anjali Goel	Psychiatry	9805056311	anjali@cc	P-22 Road, Rohini	Bangalore
1040	Gagandeep Dillon	General Practice	9805054321	gagandeep@cc	P-22 Road, Dwarka	Bangalore

## View patients

Patient Id	Name	Mobile No	Blood Group	Email	Date Of Birth	History
101	Rohan Sharma	7899999078	O+	rohan@pp	05/15/2018, 10:55:36 AM	<a href="#">View History</a>

Patient Name	Doctor Name	Problem	Treatment	Treatment Notes
Rohan Sharma	Rajat Chauhan	Immediate Attention	Painkiller	Fracture in arm



## View Feedback

View Feedback			
Feedback Id	Doctor Id	Doctor Name	Feedback
1	1010	Rajat Chauhan	Okay doctor

## **CHAPTER 5 - CONCLUSION**

### **5.1 Conclusions**

The main motive of this project is to make an Efficient Doctor-Patient portal system, so that this helps patients to efficiently view doctors, book appointments, and also view medical progress. The doctors will be able to view their appointments, their patients profile and medical history. This system has an add-on which allows user to register for organ donation as well as search for organs.

We have been successful in adding all the above requirements. A patient can successfully register himself with the system. After registration he can login at any time of the day and can search doctors in different cities according to their specialty. According to availability of the doctors a patient can book appointment with the doctor. One can also view their medical history. A few add-ons that we have added here are that a patient can successfully change his password and update his/his profile, one can even login if he forgets his password through the hint. Other add-on is that patient can give feedback to the doctors who examined him.

We have successfully created an admin login. Admin is one who can add doctors to this portal. Admin can also view other details of doctors and patients.

A doctor can successfully login into the portal with the credentials given by the admin. He can view his appointments and work accordingly. He can also view his patients and their details. He will be able to add treatment to his patients. The add-on feature here is that he can update his profile and change password.

A patient can successfully register and deregister for organ donation. He can also search for the organs were he will find the count of the organs that have been registered. This feature will help many needful people with organ they need.

### **5.2 Future Scopes**

This application can be converted to the mobile app, so that it becomes easier for patients to book appointment with doctors. It will also be easier for doctors to view there appointments any time they want. It can be extended further by adding additional functionalities like:

- Tracking the device location of the users.
- Tracking the nearest clinic according the current location of the users.
- Doctors could be given the feature to select the time slots they will be available to give service to patients.
- This project does not have the location of the doctor's clinic

## REFERENCES

[1]. Infosys Learning Platform.

[2]. [www.w3schools.com](http://www.w3schools.com)

[3]. Research on HTML5 in Web Development

Ch Rajesh, K S V Krishna Srikanth, Department of IT, ANITS, Visakhapatnam (2014).

[4]. Angular JS

Sneha Ambulkar, Student, Department of CSE, GCOE, Amravati, India,

Asst.Prof. Pushpanjali M.Chaouragade, Department of CSE, GCOE, Amravati, India(2016).

[5]. HIBERNATE TECHNOLOGY FOR AN EFFICIENT BUSINESS APPLICATION  
EXTENSION

B.Vasavi, Y.V.Sreevani

Associate Professor, Department of Computer Science and Engineering

Hyderabad Institute of Technology and Management [HITAM], Hyderabad, A.P, India.

Associate Professor, Department of Computer Science and Engineering

Student, B.Tech Final Year, Department of Computer Science and Engineering(2011).