# DWIT COLLEGE

# DEERWALK INSTITUTE OF TECHNOLOGY

## Tribhuvan University

## Institute of Science and Technology



# DOCTOR'S TIME

## A PROJECT REPORT

### Submitted to

### Department of Computer Science and Information Technology

### DWIT College

*In partial fulfillment of the requirements for the Bachelor's Degree in Computer Science and Information Technology*

Submitted by

Krishna Chauhan

August, 2016

# DWIT College
# DEERWALK INSTITUTE OF TECHNOLOGY
## Tribhuvan University

# SUPERVISOR'S RECOMMENDATION

I hereby recommend that this project prepared under my supervision by KRISHNA CHAUHAN entitled **"DOCTOR'S TIME"** in partial fulfillment of the requirements for the degree of B.Sc. in Computer Science and Information Technology be processed for the evaluation.

…………………………………………

Sarbin Sayami

Assistant Professor

Institute of Science and Technology

Tribhuvan University

**DWIT College**

**DEERWALK INSTITUTE OF TECHNOLOGY**

**Tribhuvan University**

## LETTER OF APPROVAL

This is to certify that this project prepared by KRISHNA CHAUHAN entitled **"Doctor's Time"** in partial fulfillment of the requirements for the degree of B.Sc. in Computer Science and Information Technology has been well studied. In our opinion it is satisfactory in the scope and quality as a project for the required degree.

| | |
|---|---|
| …………………………………… <br> Sarbin Sayami [Supervisor] <br> Assistant Professor <br> IOST, Tribhuvan University | …………………………………………… <br> Hitesh Karki <br> Chief Academic Officer <br> DWIT College |
| ………………………………………….. <br> Jagdish Bhatta [External Examiner] <br> IOST, Tribhuvan University | ………………………………………….. <br> Rituraj Lamsal [Internal Examiner] <br> Lecturer <br> DWIT College |

# ACKNOWLEDEMENT

It gives me immense pleasure to express my deepest sense of gratitude and sincere thanks to our highly respected and esteemed project instructor Mr. Sarbin Sayami (Asst. Prof., IOST, TU) for his valuable guidance and encouragement for completing this work. His useful suggestions for this whole work and co-operative behavior are sincerely acknowledged.

I would like to express my sincere thanks to Mr. Hitesh Karki, CAO (Chief Academic Officer), for giving me this opportunity to undertake this project. I am grateful to Mr. Sanjeev Budha and Mr. Asim Regmi for their constant support and guidance. At the end, I would like to express my sincere thanks to all friends and others who helped me directly or indirectly during this project work.

Krishna Chauhan

TU Exam Roll no:  1801/069

# STUDENT'S DECLERATION

I hereby declare that I am the only author of this work and that no sources other than the listed here have been used in this work.


... ... ... ... ... ... ... ...


Krishna Chauhan

Date: August, 2016

# ABSTRACT

The project entitled "Doctor's Time" is based on doctor, hospital and user entity. It provide platform for user to view list of hospital and doctor. User can search hospital and their associated doctor with respect to location of hospital. Besides detail description on hospital and doctor, user can request for appointment to doctor.

First come first service (FCFS) scheduling algorithm is implemented for scheduling the user appointment.

Keywords**:** *Doctor's time, first come first service*

# TABLE OF CONTENTS

# LIST OF FIGURES

# LIST OF TABLES

# LIST OF ABBREVIATIONS

SWOT       Strength weakness opportunity threats

API       Application program interface

SDLC       Software development lifecycle

FCFS       First come first service

# CHAPTER 1: INTRODUCTION

## 1.1 Background

The development of medical science emerge the development of hospital system. Lots of research has been carried out in a scientific approach in order to diagnosis the human diseases hence people prefer hospital for treatment. In present context if person want to know about hospital and its associated doctor for treatment then he/she has to physically present over hospital. Every second of patient is precious and hence it is good to have a prior information about hospital and doctor. If all of the process could be done automated then it could be easier then above manual process and lots of time could be saved. Hence the treatment procedure could be efficient.

Doctor's Time is a web application where user can view list of hospital and doctor. Although lots of application exist over the android play store. Doctor's time provide the user friendly platform for searching the doctor according to the location of hospital beside description of hospital and doctor it provide virtual platform for user to request appointment to doctor.

This application ultimately helps people over the valley and hence remove the traditionally approach of going hospital for getting the information about the availability of doctor associated to hospital.

## 1.2 Problem Definition

With the increase in a number of hospital, people often get confused about the best-suited hospital according to their preferences. In addition to that, people face a hard time to find out the best hospital for treatment, especially when they are new to that place. Due to lack of information about the hospital and doctor many patient had to be in dilemma for getting service. Prior information on hospital and doctor is necessary before treatment. Today the era

of modern digital technology, all the work are automated and computed online. No need to present physically over a place to make things happens

(Daniel Lemire, 2013)

## 1.3 Objectives

The main objectives of Doctor's Time are as follows:
   a. To view information on hospital and its associated doctor.
   b. To search hospital and its associated doctor according to hospital location.
   c. To request appointment to doctor.

## 1.4 Scope

Doctor's Time can be used by people interested for prior information on hospital and doctor for treatment and requesting appointment to doctor. Hence this application reduce the effort of going hospital physically for getting information about doctor and hospital for treatment.

## 1.5 Limitation

   a. Appointment request by user may not be considered by the doctor.
   b. Only Selected registered Hospital with in a Kathmandu valley is listed.
   c. Only the registered user will be allowed to use the system

## Outline of Document

The remaining part of the document is organized and represented in project block diagram as shown in Figure-1 below:

Preliminary

Section

- Title Page
- Abstract
- Table of Contents
- List of figures and Tables

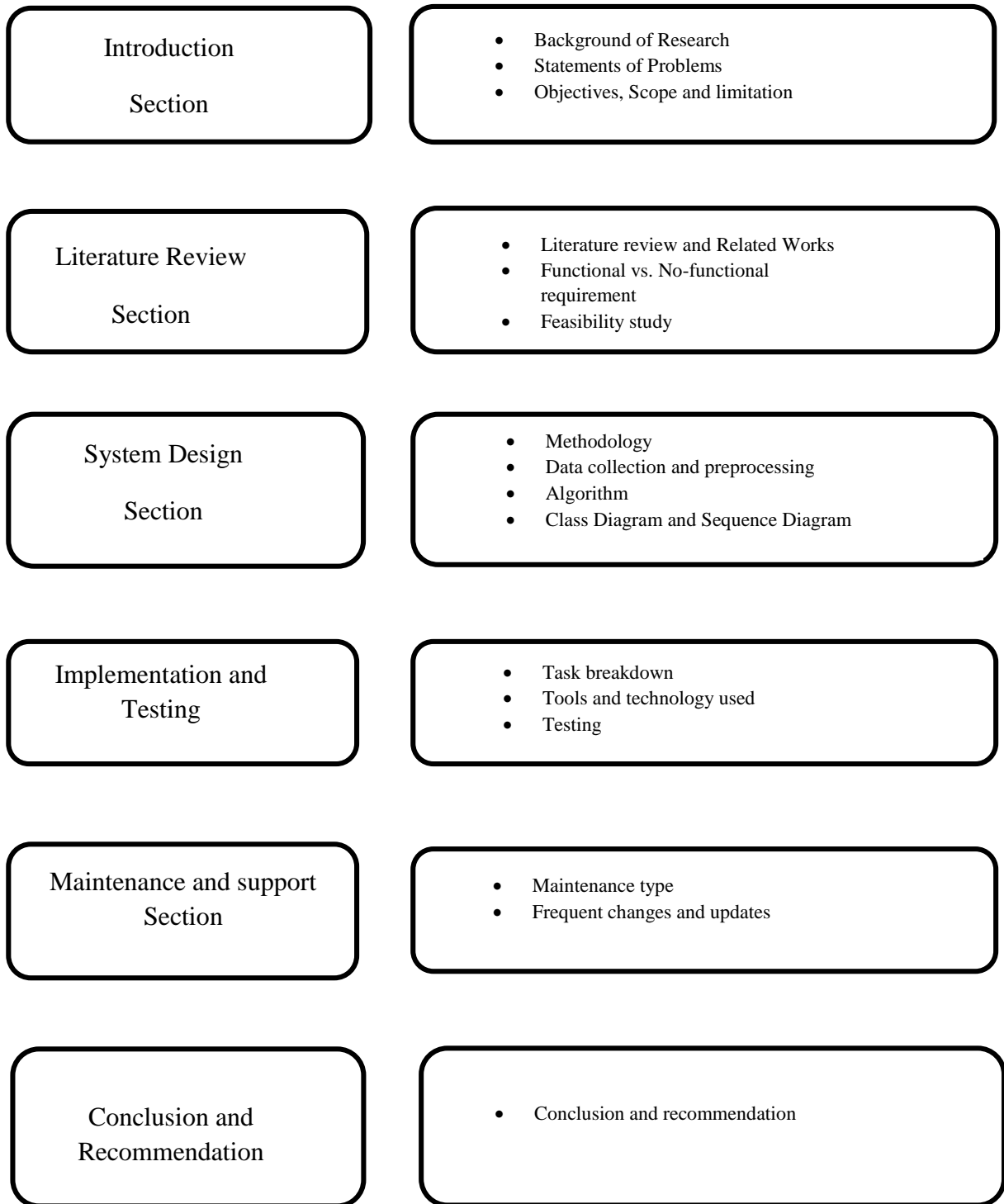| | |
|---|---|
| Introduction<br><br>Section | • Background of Research<br>• Statements of Problems<br>• Objectives, Scope and limitation |
| Literature Review<br><br>Section | • Literature review and Related Works<br>• Functional vs. No-functional requirement<br>• Feasibility study |
| System Design<br><br>Section | • Methodology<br>• Data collection and preprocessing<br>• Algorithm<br>• Class Diagram and Sequence Diagram |
| Implementation and Testing | • Task breakdown<br>• Tools and technology used<br>• Testing |
| Maintenance and support Section | • Maintenance type<br>• Frequent changes and updates |
| Conclusion and Recommendation | • Conclusion and recommendation |

Figure 1- Project outline

# CHAPTER 2: REQUIREMNET ANALYSIS AND FEASIBILITY STUDY

## 2.1 Literature Review

### 2.1.1 Scheduling

Scheduling is one of the most primary and essential part of any operating system. It allow one to decide which threads are given to resource from moment to moment. It prioritizes processes to efficiently execute the user requests and help in choosing the appropriate process for execution. Round Robin (RR) & Priority Scheduling (PS), First Come First Service(FCFS) are one of the most widely used and acceptable CPU scheduling algorithm. CPU scheduling is similar to other types of scheduling, which have been studied over the years. CPU scheduling refers to the decision of allocating a single resource among multiple clients.

(Garg, 2009)

Oulu hospital wants to identify and reduce large waiting times at their outpatient clinic. For the past few years the clinic has used a self-service system whereby patients register on arrival and hospital use a patient call-in system. Appointment scheduling problems have received the attention of operations research (OR) scientists since Bailey's initial work in 1952, but the problem of long waiting times still exists in the outpatient clinic. The no-show behavior of patients has been identified as a key factor resulting in low efficiency in many field study papers. Attempts to cut down on no-shows by reminders, telephones, or emails play an important role in mitigating no-show phenomenon, but it cannot eliminate the negative effects entirely. A revolutionary appointment system named the open-access system has been introduced as an alternative method to reduce no-show behavior by giving some patients appointments on the day they call.

(Strahl, 2015)

In 'Sequential Appointment Scheduling', Chongjun Yan, Jiafu Tang,and Bowen Jiang perform in-depth analysis on  sequential appointment algorithm considering walk-in patients, system capacity and no show patient . The objective is to determine the optimal booking number of patients and the optimal scheduling time for each patient to maximize the revenue of all the arriving patients minus the expenses of waiting time and overtime. Based on the assumption that the service time is exponentially distributed, this paper proves that the objective function is convex.

(Chongjun Yan, 2014)

### 2.1.2 Comparative analysis on different scheduling algorithm

Various process scheduling algorithms exist and focuses on the scheduling algorithms used for scheduling processes in a multiprogramming system namely First-Come-First-Served (FCFS), Round Robin (RR), Shortest Job First (SJF), Shortest Remaining Time First (SRTF) and Lottery scheduling. Each algorithm has been discussed and a comparison was made on the basis of eight parameters significant in processes scheduling. In fact, compared to other papers, this research made use of more parameters for the analysis. These parameters include CPU utilization, throughput, waiting time, response time, fairness, starvation, predictability and preemption.

Table 1- Comparative analysis on scheduling algorithm

| Algorithms | FCFS | RR | SJF | SRTF | LOT |
|---|---|---|---|---|---|
| CPU Utilization | Low | Medium | Higher | Requires very little overhead | CPU time prop to the job number of tickets given to each job |
| Throughput | Traded off for better Response Time | Bad when the chosen quantum is small | Traded off for better Response Time | Gives the highest throughput of all scheduling algorithms. | Probabilistic guarantee of throughput proportional to ticket allocation |

| Waiting Time | The average waiting time is large | The average waiting time is large as compared to others algorithms | The average waiting time is small compared to other algorithms | The average waiting time is large | Low waiting time |
|---|---|---|---|---|---|
| Response | Good | Time Bad when the number of processes is large | Good | Good | Good |
| Fairness | Unfair for long jobs make short jobs wait | Fair | Unfair | Unfair | Fair |
| Predictability | More predictable | Predictable | Impossible to predict amount of CPU time | Impossible to predict the amount of CPU time a job has left | Less predictable |

As shown in Table-1, different CPU scheduling algorithm has been analyzed and comparative differentiation on different scheduling algorithm based on different scheduling criteria is listed above.

(Adekunle1, 2014)

Many service systems provide appointments to customers in advance of their arrival. However, because service times are uncertain, the amount of time to allocate between customer arrivals is a challenging decision. Short inter-arrival times can lead to high service system utilization, but at the expense of long customer wait times. Long inter-arrival times, on the other hand,

tend to reduce customer waiting, but at the expense of lower resource utilization. When the number of customers to be scheduled is known in advance, schedules can be designed using stochastic optimization models. However, in many service systems appointment scheduling is complicated by the fact that the exact number of customers to be scheduled is not known in advance. Instead, customers request appointments sequentially over time, and appointments are quoted on-line, i.e. sequentially at the time of each appointment request. Since rescheduling of appointments is uncommon in most service industries it is necessary to make these on-line scheduling decisions in such a way that schedules are adaptable to variation in customer demand.

(Gupta, 2003)

Since much of the literature is in the context of health care so I use the terms patient and customer interchangeably. Most of the previous work on dynamic appointment scheduling has assumed a fixed first-come-first-served (FCFS) queue discipline. Figure 1 illustrates the evolution of an on-line appointment schedule over time for a specific example in which up to 5 customers are scheduled. Figure 1(A) illustrates the FCFS policy in a dynamic scheduling environment. Figure 1(B) illustrates the more general case, which we consider in this article, in which the sequence is not fixed a priori. Note that customers are scheduled in order of their appointment requests but their appointment times on the day of service do not necessarily follow that order.
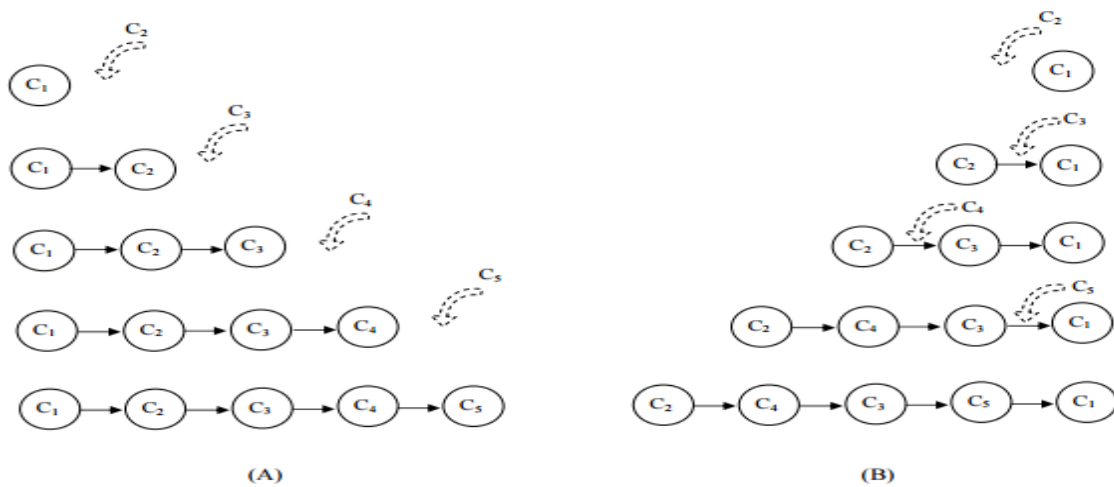


(A)　　　　　　　　　　　　　　(B)

Figure 2- First come first service

Figure-2, Illustration of the on-line scheduling problem for scheduling of up to 5 customers. Figure (A) illustrates the case in which the sequence of appointments is FCFS; Figure (B) illustrates the general case in which the FCFS sequence is relaxed.

(Alexander Gose, 2012)

## 2.2 Related Work

Some of the related project/work similar with **"Doctor's Time"** are listed below:

### Omhospitalnepal

It is a web based system that provide platform for end user to view the detail description of doctor and schedule appointment to doctor associated with om hospital. User requested appointment is schedule according to the date of appointment. Doctor manually response the appointment request.

### Hospital Nepal

The application provides health related news and articles at the fingertips of the user. Added to the features, it has database of doctors, hospitals, clinics, ambulance and blood donor's profile that can be a valuable asset for anyone in need or emergency

(java & Himalayan, 2015)

### Doct Schedule

It is web based doc scheduling system– streamlines the scheduling process by improving dock productivity, expanding visibility on scheduled appointments and measuring vendor compliance.

## 2.3. Requirement Analysis

### 2.3.1. Functional requirement vs. non –functional requirement

A requirement analysis was done on the product and the following data was obtained:

Table 2- Functional requirement vs. Non –functional requirement

| SN | Role | Task-Functional requirement | Description-Non Functional requirement |
|----|------|-----------------------------|----------------------------------------|
| 1 | Admin | Login | Admin should provide the valid credential in order to use the system. |
| 2 | User | Sign up/login | User Should sign up first in order to create the account and login for using the system. |
| 3 | User | Search | User can search for hospital and their associate doctor on the basis of hospital location. |
| 6 | User | View details | User can view the details information about hospital and doctor. |
| 7 | User | Set appointment | User can set the appointment to the doctor through the system. |
| 8 | Doctor | Login | Doctor should provide a valid credential in order to use the system. |
| 9 | Doctor | View appointment | Doctor can view the appointment request by the user and set the date according to their preference. |

As shown in Table-2, System provide interface to admin for constant information updating through the admin authentication. User should login in to the system followed by the sign up process. Information about the doctor, hospital and appointment are viewed on their respective dashboard of user. User can search the hospital according to the hospital location. The list of hospital and associated doctor are generated. User can request for appointment to doctor and doctor can view the request appointment by the user and set the date to user according to preference.
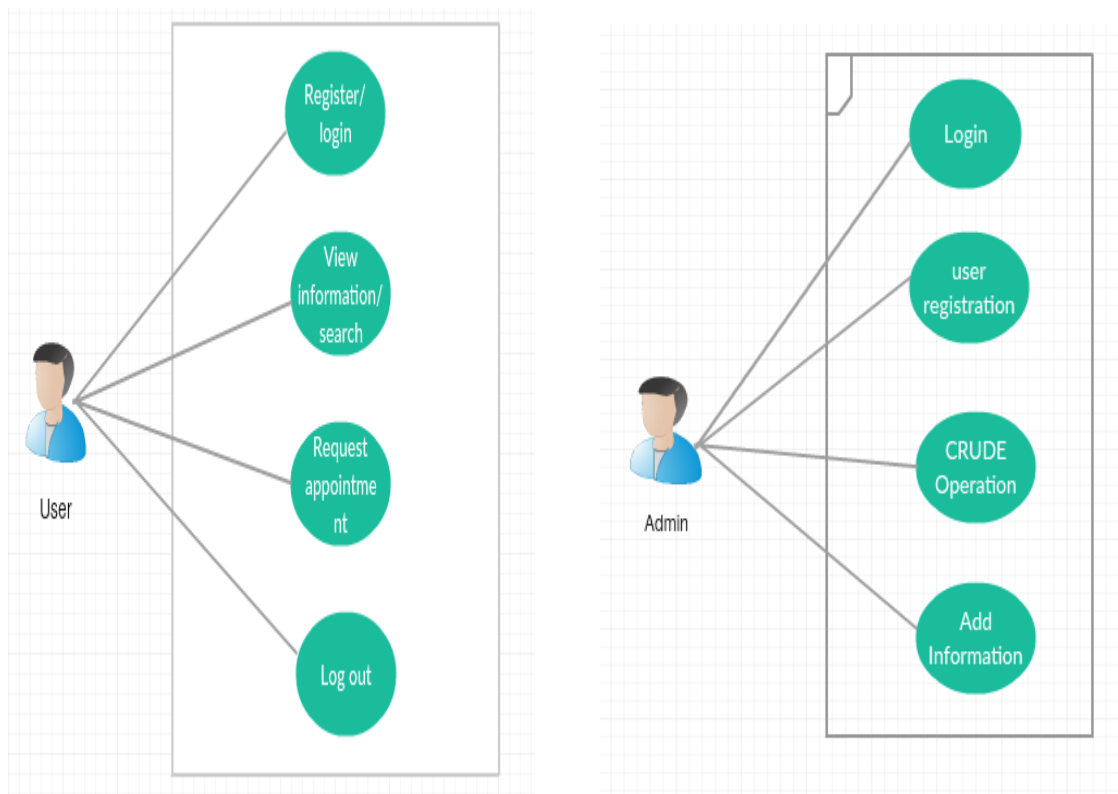
## 2.3.2. Use case



Figure 3- Use case diagram

Figure-3, shown above describe the activity of user and admin .Admin and user are the major role in this system. Admin is responsible for constant uploading and updating the information through valid credential and on the other side user can view the information through login followed by the registration/sign up process. User can even request for appointment to doctor.
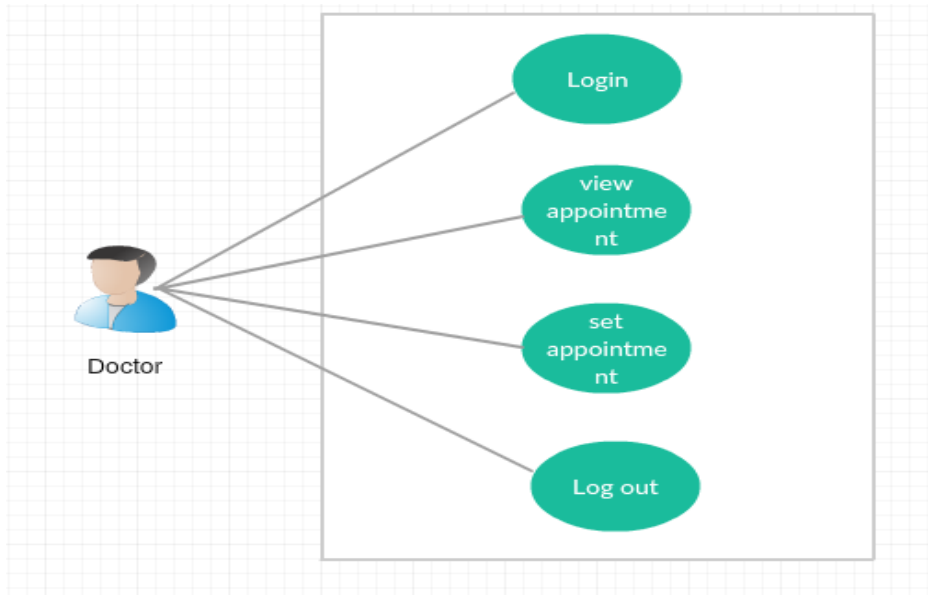
Figure 4- Use case diagram

In figure-4, doctor activity are visualized via use case diagram shown above. Doctor plays a vital role in this system. Doctor and user are interconnected. Doctor is responsible for constant appointment response to the user. Doctor has to login for using the system and is created by admin.

## 2.4. Feasibility Analysis

The following result was obtained while performing a feasibility analysis:
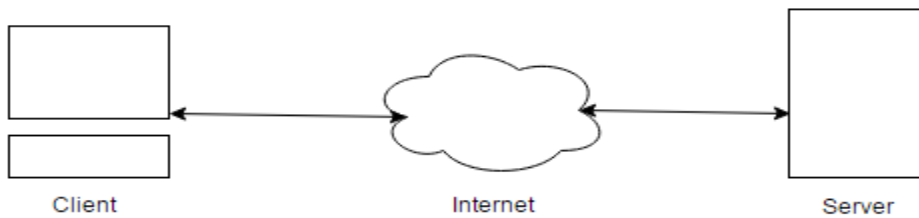
### 2.4.1 Operational feasibility



Figure 5- Architecture of operational feasibility

Doctor's Time is based on client-server architecture where server store and serve the information added by admin, user and doctor. The application can be accessed from anywhere

at any time with an internet connection and hence it is easy to use. Thus, it is operationally feasible.

### 2.4.2. Technical feasibility

System can be supported on both windows and Linux platform for its operation. It is a web application that uses HTML/CSS and JavaScript at the front end and PHP and MySQL at the back end. All of the technology required for developing application are available and can accessed openly and freely, hence it was technically feasible.

### 2.4.3. Schedule feasibility

Sprint is planned based on the application requirement. The task covering in each sprint is figure out and the deadline for the sprint is estimated.

Table 3- Schedule feasibility

| Legend | Activity | Duration | Precedence |
|--------|----------|----------|------------|
| A | Requirement analysis | 2 week | A |
| B | Feasibility study and data crawling | 1week | A,B |
| C | Mentor help | 2week | C |
| D | Coding | 2 week | C,D |
| E | Testing | 1 week | A,B,C,,D |
| F | Documentation | 3 week | A,B,C,D,E |

The total time for the development of the application is 3 months. From the Table-3, above we can see that Doctor's Time is completed in 11 week.

# CHAPTER 3: SYSTEM DESIGN

## 3.1 Methodology

To illustrate the problem shown in problem statement I choose to develop a web-based application using waterfall model. SQL tokenization is used for retrieving information from database and displaying information to user.

 User can request for appointment to doctor. First come first service (FCFS) scheduling algorithm is implemented for scheduling the user appointment.

<div align="right">(Manya, 2015)</div>

### 3.1.1 Data Collection

Data/information required for this application is collected manually from the two well recognized hospital named Medicare national hospital and OM hospital and research center. Other information on hospital and doctor is collected from internet visiting their respective website. The collected data are the raw data.  Table-4 shown below define the sample format of data.

Table 4- Data format

| Hospital Name | Hospital Location | Doctor Name | Specialist |
|---|---|---|---|
| **Medicare Hospital** | Chabhil | Dr. Arun Shrestha | Nephrology |
| | | Dr. Arunima Shrestha | Neurology |
| | | Dr. Abhani Upadhaya | Cardiologist |

### 3.1.2 Data preprocessing

The collected data is stored and maintained via MySQL database management system using a relational model.

For appointment sequencing and scheduling, First come first service scheduling algorithm is implemented in which first come first service scheduling algorithm is implemented for scheduling the user appointment.

### 3.1.3 Algorithm

CRUD Operation

CRUD refers to the four major functions implemented in this applications. The CRUD functions are the user interfaces to databases, as they permit users to create, view, modify and alter data. CRUD works on entities in databases and manipulates these entities.

Scheduling algorithm

For scheduling the user appointment FCFS algorithm is implemented.

a) Once doctor is fetched, appointment capacity for the respective doctor is allocated by admin.
b) User request for appointment to doctor.
c) If number of appointment request by user equal to doctor appointment capacity then set the appointment on the next consecutive day on FCFS basis.
d) If number of appointment request by user exceed the doctor appointment capacity then set the remaining appointment after the first priority user is served.

### 3.1.4 Retrieving the information

SQL query is used for the retrieving the necessary information and displaying to the end user. FCFS is implemented for scheduling the user appointment.

### 3.1.5 Output variable

The output variable will be:
a. List of doctor
b. List of Hospital
c. Search result
d. Appointment response

# 3.2 System Design

## 3.2.1 System requirement – hardware and software platform

The hardware requirement for the application is:

a. Device Type: Personal Computers (laptop or desktop).
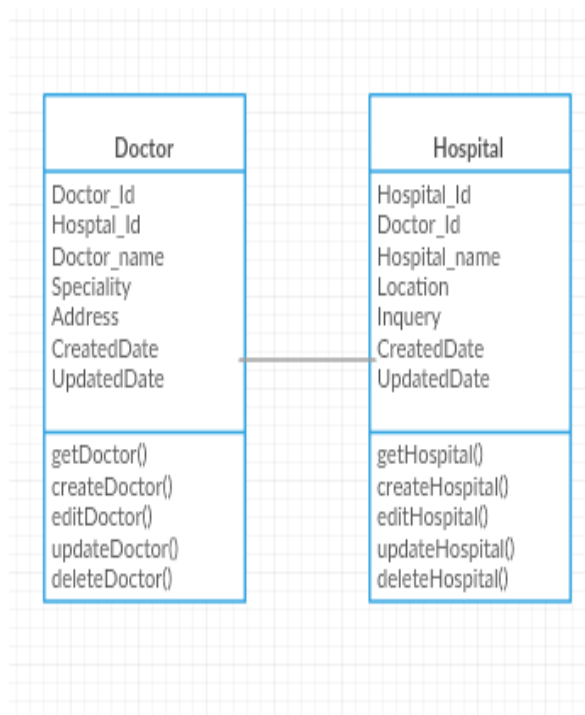b. Operating System: Windows OS, Linux, MAC

## 3.2.2 Class diagram



Figure 6- Class diagram

Doctor and hospital are the main class in this project. As shown in figure-5, doctor id and hospital id both are used in each class that uniquely represent the doctor and hospital. Each doctor is associated with hospital and each hospital has a doctor. Individual doctor can be associated with the multiple hospital so the hospital id and doctor id both attribute is used in doctor and hospital entity. The function shown in figure-5, depict the different operation used in the class.
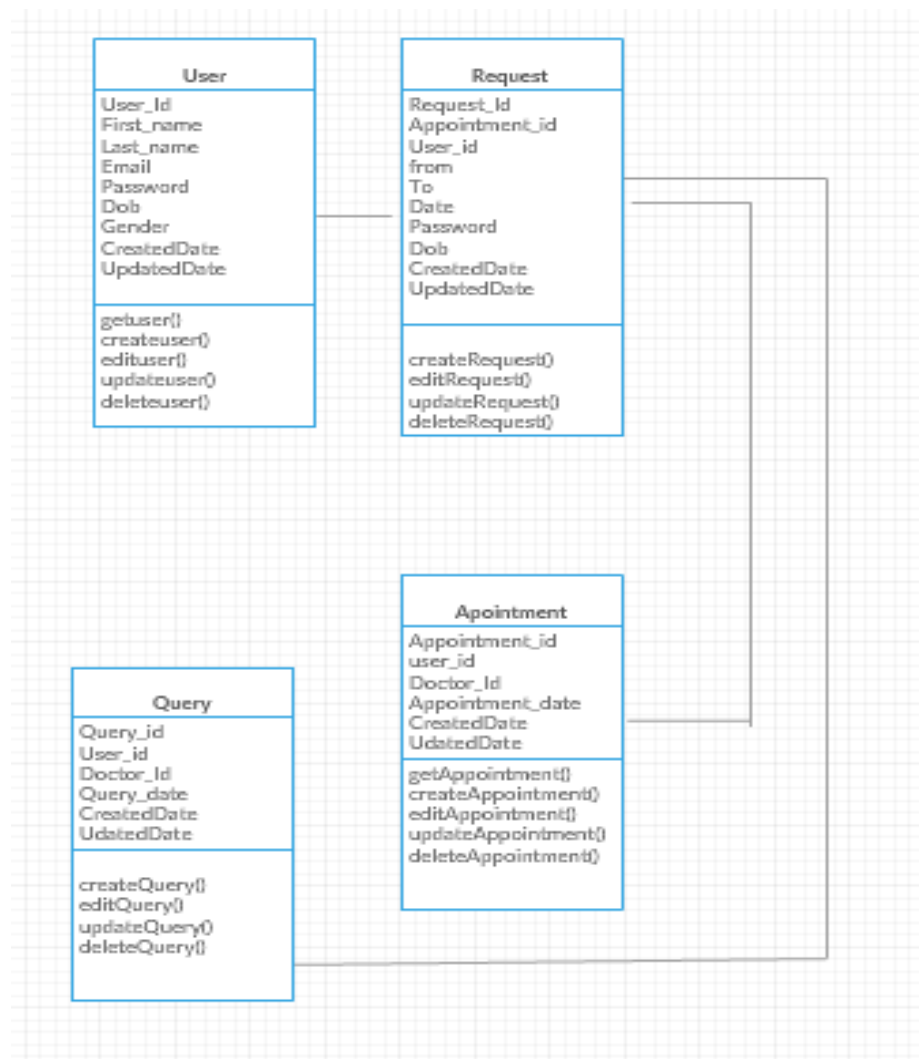
Figure 7- Class diagram

As shown in figure-6, user and appointment class is depict. User has a privilege to sign up and create an account. Individual user has a user id that uniquely represent the user. User has a direct relation with appointment, query and doctor class. User request for appointment via request class and appointment class create a unique appointment id for user. Different function in the class illustrated above depict the operation in each class.

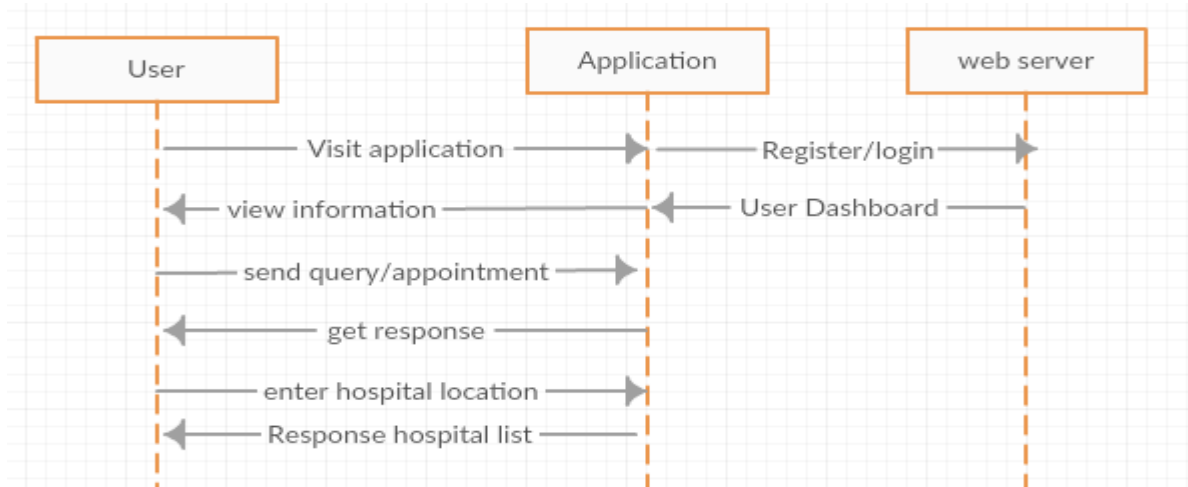**3.2.3. Sequence diagram**



Figure 8 - Sequence Diagram (User)

As shown in figure-7 user has to first login followed by registration in order use the system. User request the system for viewing and searching a related information on the other side admin manage and update the information through web app and ultimately user can view their required information. User can send the appointment and query request and hence application redirect the response.
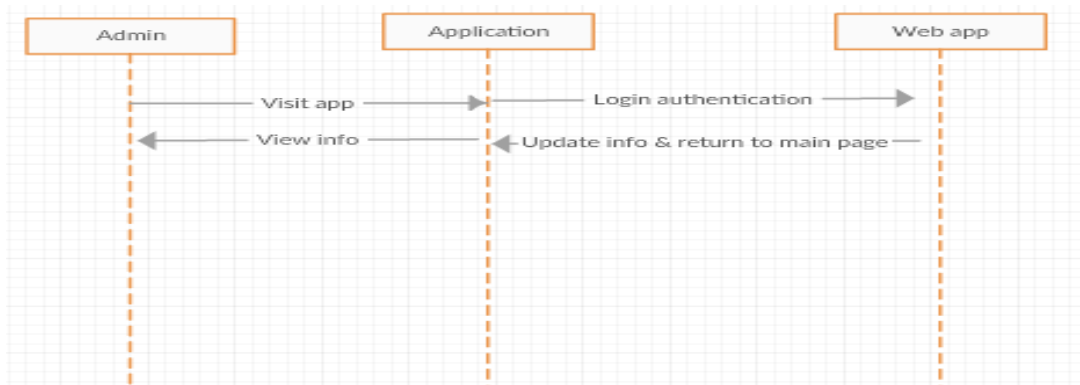


Figure 9- Sequence Diagram (Admin)

Admin manages hospital and doctor department and all the information regarding the hospital and doctor and are updated instant as shown in Fig-8. User management is controlled and managed by admin.
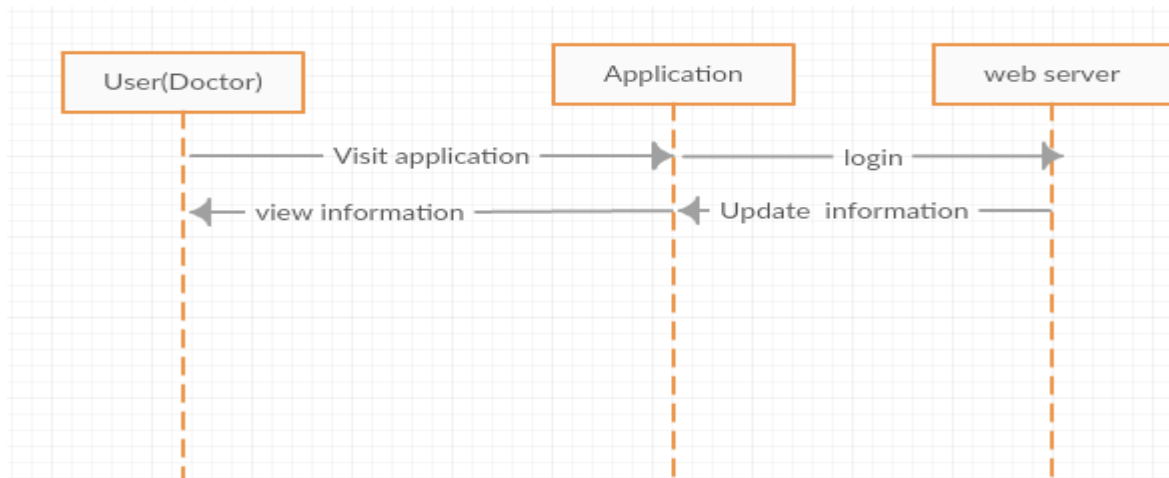
Figure 10- Sequence Diagram (Doctor)

Doctor are the special entity and are responsible for managing the user appointment. As shown in figure-9, doctor has to login for using the system and is managed and created by admin. User appointment request are handle by the doctor and give response according to FCFS scheduling algorithm.

# CHAPTER 4: IMPLEMENTATION AND TESTING

## 4.1 Implementation

**Task Breakdown**

Sprint is planned based on the application requirement. The task covering in each sprint is figure out and the deadline for the sprint is estimated.

This section describes about the technologies used in Doctor's Time. Doctor's Time is a web application. PHP framework is used for developing the application. MySQL is a Popular and most common choice of database for developing web application which is used for database management system. WAMP server is used for hosting the system locally. For client side, HTML is used for presentation of the content Similarly, CSS was used as a style sheet language to describe the look and Formatting of a document written in HTML.

Furthermore, Bootstrap is used as a CSS library. JavaScript and JQuery (library of JavaScript) is used as a client side scripting language for client side validation. Ajax, combo of JavaScript and XML used for asynchronous communication with server and client.

### 4.1.1 Tools Used

**PHP**

PHP is used as a programming language for developing the system and PHP storm is used as an IDE platform for developing the web application.

**Creately**

Creately was used to build use case diagram, class diagram, sequence diagram.

### 4.1.2 Listing of major classes / methods

```php
require ('../config/databaseConnection.php');
require('../common/Common.php');


$objCommon = new Common();
if(isset($_REQUEST['action']) && !empty($_REQUEST['action']))
{

    $rs = array();


    $method = $_REQUEST['action'];
    $result = $objCommon->$method($_REQUEST);


    $rs["data"] = $result;
    echo  json_encode($rs);


}
```

 Description of handler class

Handler are intermediary function which acts as link between web pages and common class.

The handler for this application are: -

**doctorHandler**

```php
$objCommon = new Common();

if(isset($_POST['mode'])){

  if($_POST["mode"]=="add"){

      $doctor_name = $_POST['doctorName'];
      $specialist = $_POST['specialist'];
      $associated_hospital = implode(",",$_POST['associated_hospital']);
```

```php
    $address = $_POST['address'];
    $mobile_number = $_POST['mobileNumber'];
    $image = $_FILES['image']['name'];
    move_uploaded_file($image_tmp,"../images/$image");



    $result = $objCommon-
>createDoctor($doctor_name,$specialist,$associated_hospital,$address,$mobile_number,$image);
```

**HospitalHandler**

```php
$objCommon = new Common();

if(isset($_POST['mode'])){

  if($_POST["mode"]=="add"){

    $hospital_name = $_POST['hospitalName'];
    $location = $_POST['hospitalLocation'];
    $associated_doctor = implode(",",$_POST['associatedDoctor']);
    $inquery = $_POST['inqueryDetail'];

    $hospital_image = $_FILES['image']['name'];
    $image_tmp = $_FILES['image']['tmp_name'];

    move_uploaded_file($image_tmp,"../images/$hospital_image");

    $result = $objCommon-
>createHospital($hospital_name,$hospital_image,$location,$associated_doctor,$inquery);
```

**SerachHandler**

```php
session_start();
require('../common/Common.php');

$objCommon = new Common();
```

```
if(isset($_POST["search"])){

    $location = $_POST['location'];

    $result = array();
    $result = $objCommon->searchHospital($location);

    $_SESSION['hospitalList']= $result;
    header("Location:../views/searchList.php");

}
```

## 4.2 Testing

During testing following functional requirement of the system are tested.

### 4.2.1 Test Case 1

Test Case Id: 1

Test Case Description: Admin Login

| Precondition | Input Data | Steps | Expected Outcome | Actual Result | Result |
|---|---|---|---|---|---|
| Admin should be logged in. | Account should be registered | 1. Enter valid email address and valid password. 2. Press login button. | Redirect to admin dashboard. | Successfully login | Pass |

**4.2.2 Test Case 2**

Test Case Id: 2

Test Case Description: user registration

| Precondition | Input Data | Steps | Expected Outcome | Actual Result | Result |
|---|---|---|---|---|---|
| For new user | Enter valid information | 1. Enter first name 2. Enter last name 3. Enter email address 4. Enter password 5. select gender 6. click submit | User successfully registered and redirect to login page | Successfully registered | Pass |

**4.2.3 Test Case 3**

Test Case Id: 3

Test Case Description: user login

| Precondition | Input Data | Steps | Expected Outcome | Actual Result | Result |
|---|---|---|---|---|---|
| User should be logged in. | Valid user email and password | 1. Enter valid email address 2. Enter valid password 3. Click login | Redirect to user dashboard | Successfully login | Pass |

## 4.2.4 Test Case 4

Test Case Id: 4

Test Case Description: Add doctor

| Precondition | Input Data | Steps | Expected Outcome | Actual Result | Result |
|---|---|---|---|---|---|
| Admin should be logged in. | Enter valid information related to doctor | 1. Enter doctor name, associated hospital name, specialty <br> 2. Click add doctor | Doctor created and displayed on user dashboard | Doctor successfully created | <mark>Pass</mark> |

## 4.2.5 Test Case 5

Test Case Id: 5

Test Case Description: Add Hospital

| Precondition | Input Data | Steps | Expected Outcome | Actual Result | Result |
|---|---|---|---|---|---|
| Admin should be logged in. | Enter valid information related to hospital | 1. Enter hospital name, associated doctor name, inquiry detail, location | Hospital created and displayed on user dashboard | Hospital successfully created | <mark>Pass</mark> |

**4.2.6 Test Case 6**

Test Case Id: 6

Test Case Description: Search hospital

| Precondition | Input Data | Steps | Expected Outcome | Actual Result | Result |
|---|---|---|---|---|---|
| User should be logged in | Enter valid location of hospital | 1. Enter hospital location<br>2. Click search button | List of hospital is generated with their associated doctor | Search result successfully displayed | Pass |

**4.2.7 Test Case 7**

Test Case Id: 7

Test Case Description: Appointment scheduling request

| Precondition | Input Data | Steps | Expected Outcome | Actual Result | Result |
|---|---|---|---|---|---|
| User should be logged in | Click appointment button | 1. Login using user credential.<br>2. Click appointme nt button | Patient name with description is registered for appointment in respective doctor dashboard | Appointment request details schedule to respective doctor dashboard | Pass |

**4.2.8 Test Case 8**

Test Case Id: 8

Test Case Description: Appointment response

| Precondition | Input Data | Steps | Expected Outcome | Actual Result | Result |
|---|---|---|---|---|---|
| Doctor should be logged in | Valid email address and password | 1. Login using doctor credential.<br>2. View the appointment list of user | Schedule the user appointment on FCFS basis with respect to appointment capacity per doctor defined by admin | User appointment is scheduled on FCFS basis, if number of appointment request exceed the appointment capacity then appointment schedule on next consecutive day | Pass |

# CHAPTER 5: MAINTENANCE AND SUPPORT

## 5.1 Maintenance Plan

The application will be maintained and updated over the period of time and necessary support will be provided to adapt the system to the future needs. Doctor's Time will implement corrective maintenance, perfective maintenance and preventive maintenance. Different bugs and error that may occur when the project is hosted will be maintain and resolved. For increasing efficiency various implementation will be changed. To make sure from un-ethical attack, security mechanism will be added.

# CHAPTER 6: CONCLUSION AND RECOMMENDATION

## 6.1 Conclusion

Doctor's Time provide a platform to the end user for searching doctor associated with hospital accordance to hospital location. Besides searching user can manually view the information on hospital and doctor from the respective tab in the dashboard of the application.

User can request for appointment to doctor. FCFS algorithm is implemented for scheduling the user appointment to doctor.

## 6.2 Recommendation

This application can be further enhanced to visualize the real time updates on the doctor and hospital. Interface will be created for doctors for constant uploading of dynamic data. The real time schedule of doctor associated with hospital like schedule and time will be updated in version 2.0.

# APPENDIX

Snapshot of Application
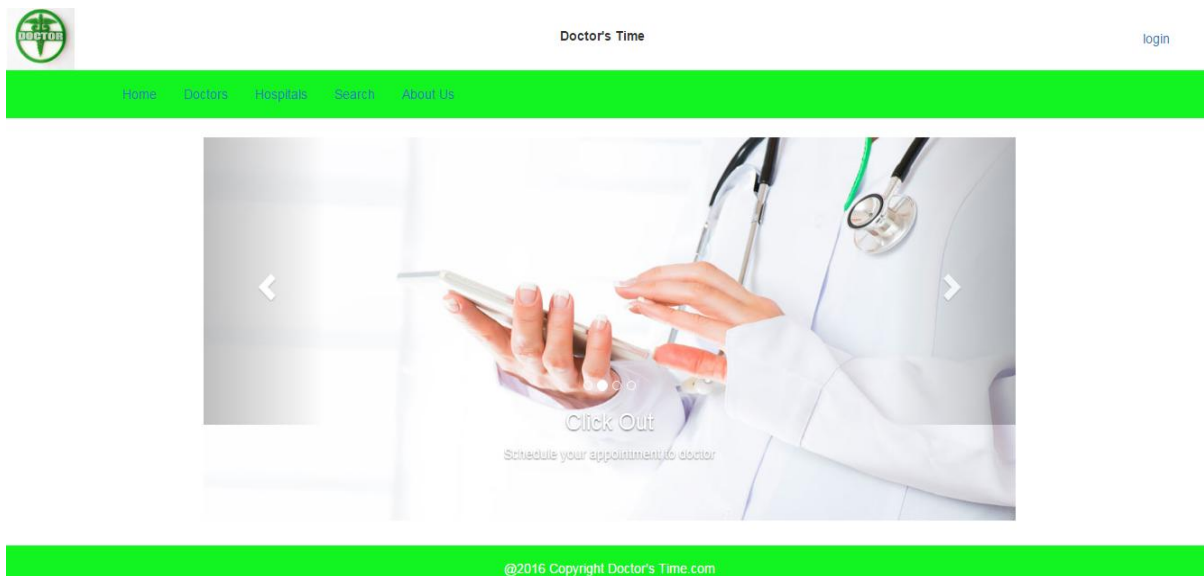
## Doctor ×

| | |
|---|---|
| **Doctor Name** | [                    ] |
| **Specialist** | Neurologist ▾ |
| **Associated Hospitals** | ☐ Bir Hospital<br>☐ Norvic Hospital<br>☐ Janata Hospital |
| **Address** | [                    ] |
| **Mobile Number** | Enter mobile |
| **image** | Choose File  No file chosen |
| | Add |

## Hospital ×

| | |
|---|---|
| **Hospital Name** | [                    ] |
| **Location** | Chabhil ▾ |
| **Associated Doctor** | ☐ Dr Arun Tamang<br>☐ Dr Krishna Chauhan<br>☐ Dr Anil Parajuli<br>☐ Dr Sameer Koirala<br>☐ Dr Sandesh Sharma |
| **Inquery Details** | Enter Number |
| **image** | Choose File  No file chosen |
| | Add |

| | Home | Doctors | Hospitals | Search | About Us |

| Photo | Full Name | Specialist | Address | Associate Hospitals | Mobile |
|---|---|---|---|---|---|
| | Dr Bhola Rijal | GASTROENTEROLOGIST | Chahabil | OM Hospital | 014482482 |
| | Dr Puspha Sharma | NEUROLOGIST | chabhil | OM Hospital | 9851142424 |
| | Dr Achyut Rajbhandari | OrthopaedicSurgery | Maharajung | OM Hospital | 9851123412 |
| | Dr Anand Shrestha | NEUROLOGIST | Sukhedhara | OM Hospital | 9843723188 |

| | Home | Doctors | Hospitals | Search | About Us |

| Photo | Hospital Name | Address | Associated Doctors | Inquery Details |
|---|---|---|---|---|
| | OM Hospital | Chabhil | • Dr Bhola Rijal | 01-4482484 |
| | TU Teaching Hospital | Maharajung | • Dr Puspha Sharma | · 01-4410911 |
| | Helping Hands | Chabhil | • Dr Achyut Rajbhandari | 9827727272 |
| | Medicare National Hospital | Chabhil | • Dr Anand Shrestha<br>• Dr Anil Shrestha | 01423456 |

**Location** [                              ]        [ Search ]

**OM Hospital**

### Associated Doctors

| Photo | Name |
|---|---|
| | Dr Bhola Rijal |
| | Dr Puspha Sharma |
| | Dr Achyut Rajbhandari |
| | Dr Anand Shrestha |

31

Show 2 ▾ entries                                        Search: [                    ]

| Patient Name | Address | email | Contact Number | Actions |
|---|---|---|---|---|
| kris chauhan | chabhil | krish@gmail.com | 9818764497 | ☑ |
| suman Thapa | Sifal | suman@gmail.com | 9842095123 | ☑ |

Showing 1 to 2 of 5 entries                    Previous  1  2  3  Next

Show 2 ▾ entries                                        Search: [                    ]

| Doctor Name | Appointment Date |
|---|---|
| Dr Bhola Rijal | 2016-09-06 |

Showing 1 to 1 of 1 entries                    Previous  1  Next

# REFERENCES

Adekunle1, Y. Z. (2014, Nov 01). A Comparative Study of Scheduling Algorithms for Multiprogramming in Real-Time. International Journal of Innovation and Scientific Research, 1, 5-12.

Alexander Gose, B. T. (2012, oct). On-line Appointment Sequencing and Scheduling. 1, 2-4.

Chongjun Yan, J. T. (2014). Sequential Appoinment Scheduling Considering Walk in Patients. Mathematical Problems in Engineering, 2-5.

Garg, D. R. (2009). Performance Analysis Of CPU Scheduling Algorithm. 3-7.

Gupta, D. a. (2003). On-line Appointment Sequencing and Scheduling.

java, H., & Himalayan, J. (2015). Hospital Nepal. Kathmandu: Hospital Nepal.

Manya. (2015, JANUARY). Php. VOLUME 4(ISSUE 01), 3-7.

Mnoj, J. (n.d.). Trust . 1-5.

Strahl, J. P. (2015, May 27). Previous studies on reducing waiting times. Patient appointment scheduling system:, 4-6.