

BSHC4 – Cloud Computing Cloud Application Development Project

Project Name: SHC-PMS
(Smart Health Care-Patient Management System)

Project Description: SHC-PMS is an information system developed using Rails and Cloud 9 and deployed on Heroku that is intended for use in various clinics/ centres in Ireland, other functionalities of the system are mentioned further in the report.

Project by: Navjot Singh Virk
Student ID: x13112406
Stream: Software Development
Course: BSc. Hons in Computing (Year 4)

Project Code Available (Github) at: <https://github.com/virksaabnavjot/shc>

Deployed at: <https://shcnavvirk.herokuapp.com/>

DECLARATION

I hereby declare that the project entitled “**SHC-PMS**” submitted for the "**Cloud Application Development Module**" is my original work and all the external sources used during the development of the project are referenced with care wherever they are used.

Signature: Navjot Singh
Date: 25/July/2017

Table of Contents

DECLARATION	1
1 Introduction.....	2
2 Development Strategy	2
2.1 Database Design	2
2.2 Implementation.....	5
2.3 Design Patterns	12
3 Deployment.....	14
4 Appendix A :Project Setup, Problems faced and Details	14
5 Appendix B :Understanding MVC in Rails Projects.....	15
My Explanation on this diagram	15
6 Appendix C: References	15

1 Introduction

SHC-PMS is an information system developed using Ruby on Rails and its intended to be used in clinics across Ireland –

In this section we will list the main functionalities of the system and provide short description (detailed description and implementation details for each functionality are further provided in the Implementation section of the report).

- 1- Add Patient record – allow the doctor to add patient record with information like Name, Dob, Condition, Appointment time etc.
- 2- View/Edit Records – allows the doctor at a clinic to add, edit or view records of the current patients (based on their health condition).
- 3- Searching – allows the doctor to search the patients by their names.
- 4- Sorting – allows the doctors to sort the patient appointments/records (based on name and appointment).
- 5- Reporting – Allows the doctor to provide patients with reports for example – a prescription note
- 6- Doctors profile – allows the doctors to signup/login to the system, and also being able to edit or destroy their profile.
- 7- Appointments Validation – makes sure not more than 1 patient can have appointment at a particular date and time with the same doctor. Helps stop chaos.
- 8- Transfers Decorator - logic implemented in lib/transfer_decorator.rb

2 Development Strategy

2.1 Database Design

Let's understand What is Database Design?

Database design is the process in which we convert our ideas into data models of how we would like our database store data and how it will look like.

In other terms, Its helps in determining how data will be stored, relationships b/w our data and helps to structure data in a logical way.

Example:

We have a user's table, for a blogging site

Users table

User_id (Primary key)	User_name	email
1	Navjot Singh	email@example.com

Then we have a posts table as well

Posts table

Post_id (Primary Key)	Post_name	User_id (Foreign Key)
1	I am a example post	1

So here in this little example we can easily see – How our data is structured in different tables (here only two) we can see the relationships as well – in this case a user can have multiple posts (i.e. one to many relationships) and a post can only have one user.

Now, we understand database design and its benefits in detail lets, look at the database design for this application project –

```
navsinghorg:~/workspace (master) $ sqlite
bash: sqlite: command not found
navsinghorg:~/workspace (master) $ rails dbconsole
SQLite version 3.8.2 2013-12-06 14:53:30
Enter ".help" for instructions
Enter SQL statements terminated with a ";"
sqlite> .tables
doctors          prints           shifts           transfers
pats             schema_migrations  trans           users
sqlite> █
```

These are all the tables in the database.

Doctors Table

```
sqlite> PRAGMA table_info(doctors);
0|id|INTEGER|1||1
1|firstname|varchar|0||0
2|lastname|varchar|0||0
3|address|varchar|0||0
4|user_id|integer|0||0
5|phone|varchar|0||0
6|created_at|datetime|1||0
7|updated_at|datetime|1||0
sqlite> █
```

Transfers Table

```

sqlite> PRAGMA table_info(transfers);
0|id|INTEGER|1||1
1|firstname|varchar|0||0
2|lastname|varchar|0||0
3|firm|varchar|0||0
4|cost|decimal|0||0
5|detail|varchar|0||0
6|age|integer|0||0
7|created_at|datetime|1||0
8|updated_at|datetime|1||0
sqlite> █

```

Pats Table

```

sqlite> PRAGMA table_info(pats);
0|id|INTEGER|1||1
1|name|varchar|0||0
2|dob|datetime|0||0
3|phone|integer|0||0
4|condition|varchar|0||0
5|appointment|datetime|0||0
6|created_at|datetime|1||0
7|updated_at|datetime|1||0
sqlite> █

```

Users Table

```

sqlite> PRAGMA table_info(users);
0|id|INTEGER|1||1
1|email|varchar|1|'|0
2|encrypted_password|varchar|1|'|0
3|reset_password_token|varchar|0||0
4|reset_password_sent_at|datetime|0||0
5|remember_created_at|datetime|0||0
6|sign_in_count|integer|1|0|0
7|current_sign_in_at|datetime|0||0
8|last_sign_in_at|datetime|0||0
9|current_sign_in_ip|varchar|0||0
10|last_sign_in_ip|varchar|0||0
11|created_at|datetime|1||0
12|updated_at|datetime|1||0

```

Prints Table

```

sqlite> PRAGMA table_info(prints);
0|id|INTEGER|1||1
1|name|varchar|0||0
2|case|varchar|0||0
3|prescription|varchar|0||0
4|published|date|0||0
5|created_at|datetime|1||0
6|updated_at|datetime|1||0
sqlite> █

```

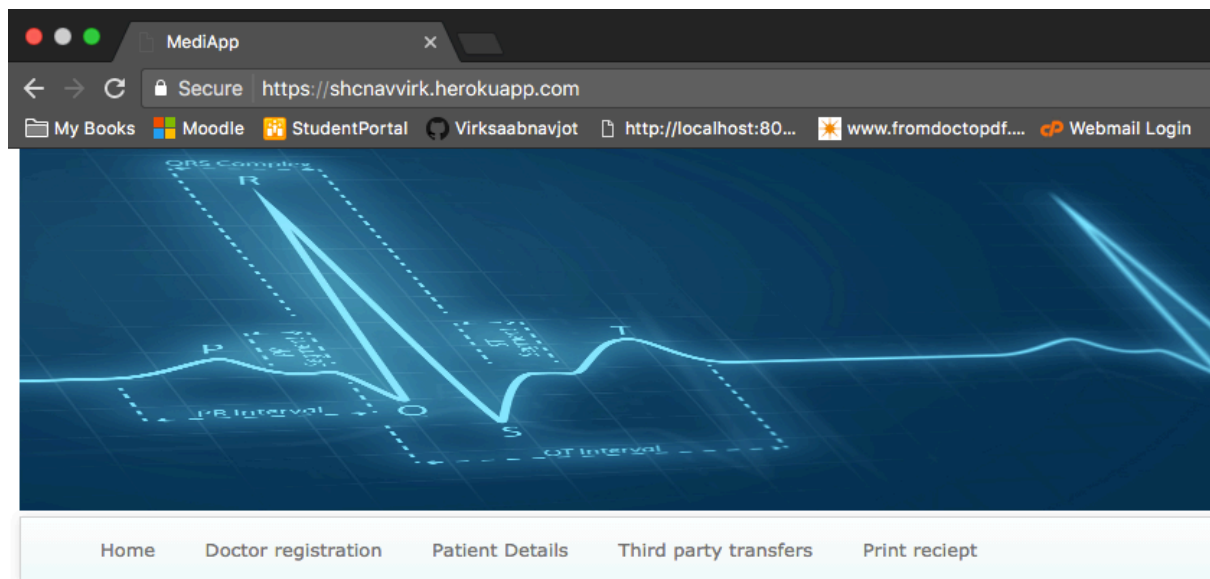
2.2 Implementation

This section contains details on different functionalities/features implemented in this application.

Functionality 1: *Add Patient Record* – Allows the doctor to add, edit or remove a patient record with Patient details like Name, DOB, Appointment Time, Condition etc.

Approach: The Add Patient Record functionality is implemented using MVC approach (explained in Appendix B section of the report). The functionality is implemented in the following files –

Model:	pat.rb
View:	Views/pats folder
Controller:	pats_controller.rb



New Patient

Name

Dob

2017 July 27 — 11 : 23

Phone

Condition

Select One

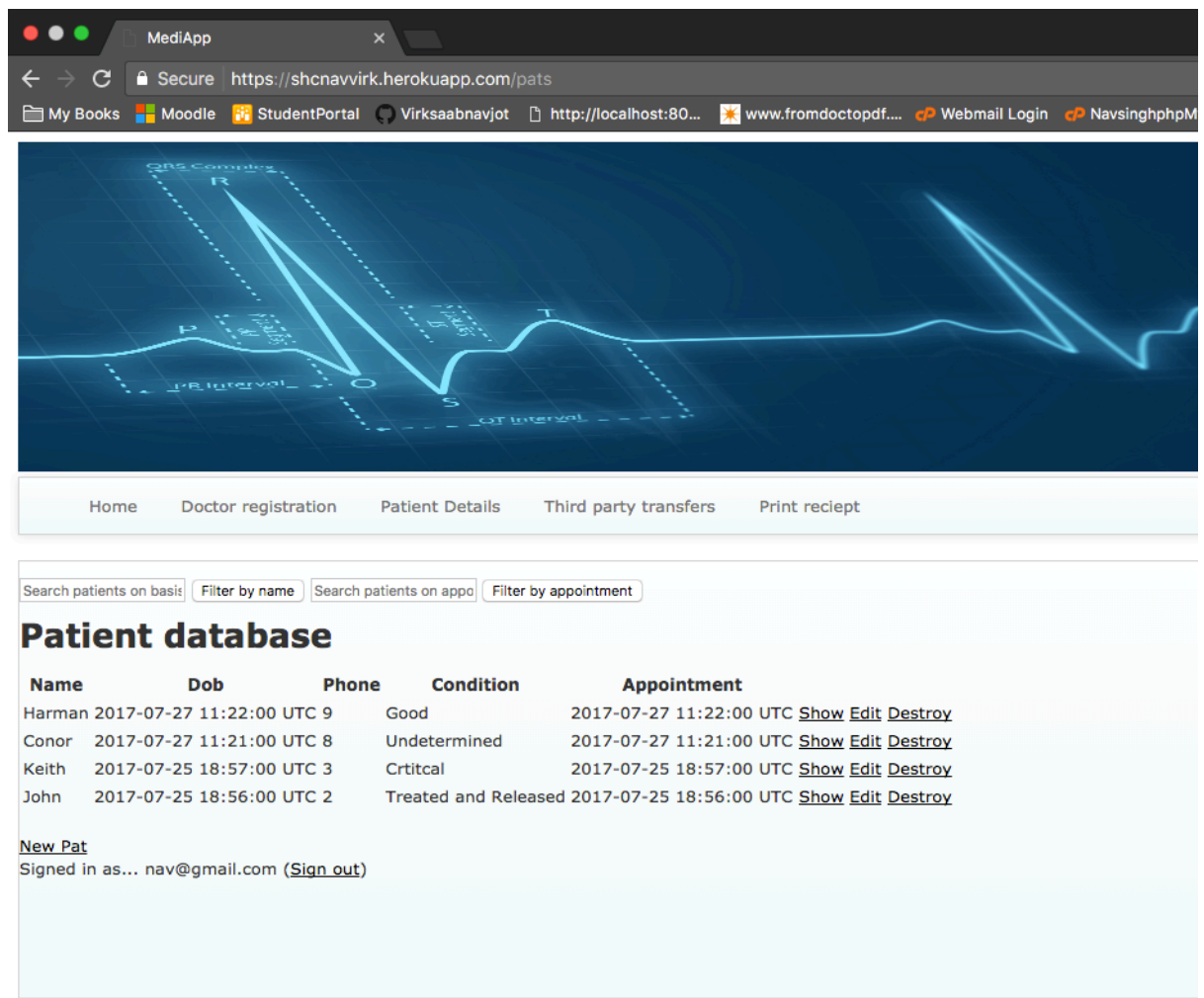
Appointment

2017 July 27 — 11 : 23

submit patient details

[Back](#)

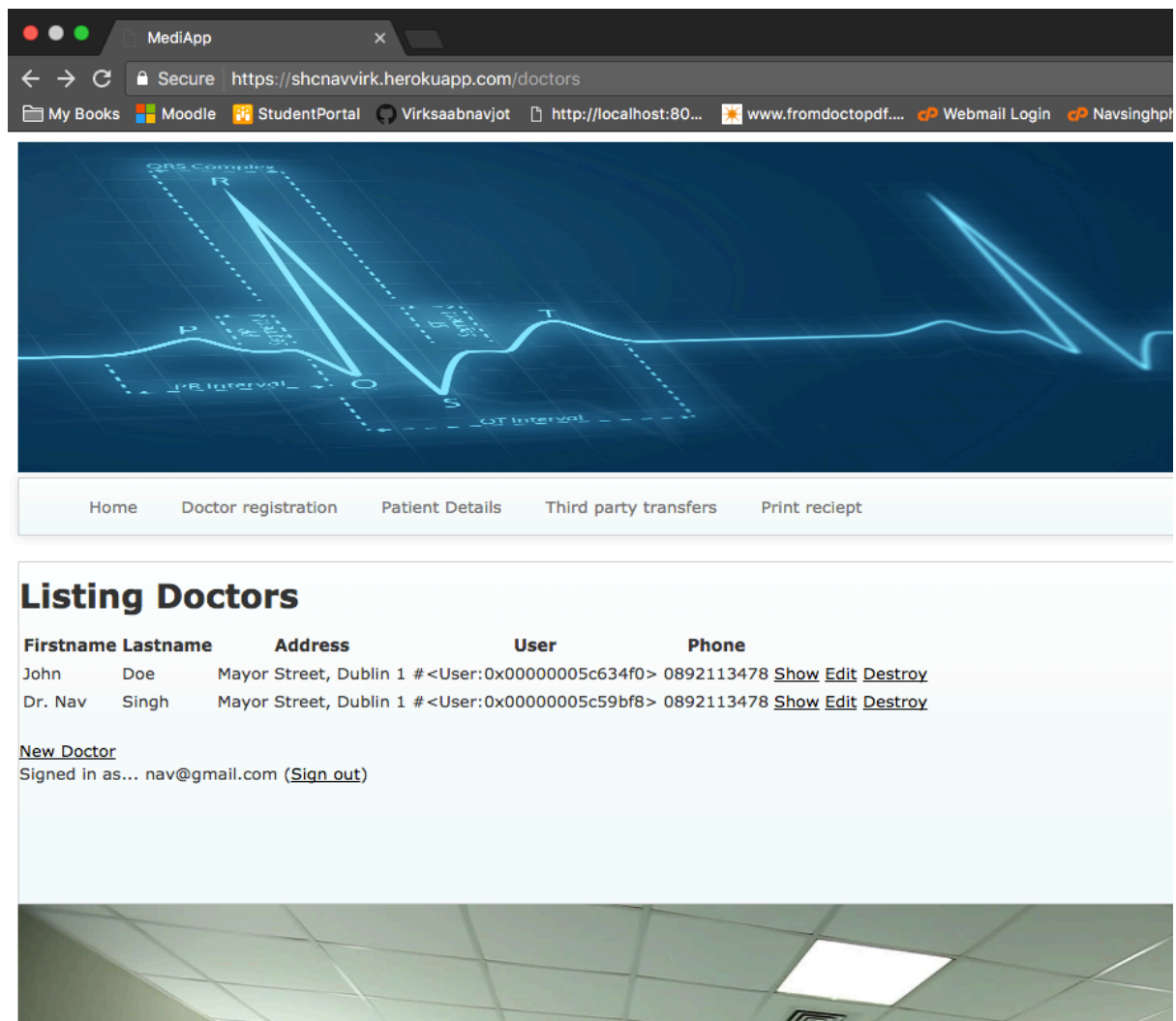
Signed in as... nav@gmail.com ([Sign out](#))



Functionality 2: Doctor Profile – profile for doctors can be viewed and edited

Approach: This functionality is implemented using MVC approach. The functionality is implemented in the following files –

Model:	Doctor.rb
View:	Views/doctors folder
Controller:	Doctors_controller.rb



Functionality 3: Transfers Decorator – Allows the doctor to transfer the patient to different hospitals for x-ray, CT-Scan and Endoscopy

Approach: Design Principle - Decorator

Location:	Lib/transfer_decorator.rb
N/A	/
N/A	/

MediApp

Secure https://shcnnavirk.herokuapp.com/shifts/new

My Books Moodle StudentPortal Virksaabnavjot http://localhost:80... www.fromdoctopdf... Webmail Login

Home Doctor registration Patient Details Third party transfers Print receipt

Patient transfer

Firstname
Vikram

Lastname
Singh

Firm

Cost
10

Description
Needs x ray

Color

X ray: ☒ St. James's Hospital

Ct scan: ☐ Mater Misericordiae University Hospital

Endoscopy : ☐ Rotunda Hospital

Do you have a medical card? : ☐

[Back](#)

Signed in as... nav@gmail.com ([Sign out](#))

Functionality 4: *Appointment Validation*- This functionality is validation functionality which makes sure if the a patient has an appointment on a date and time an another patient can't have the appointment at the same time with the same doctor.

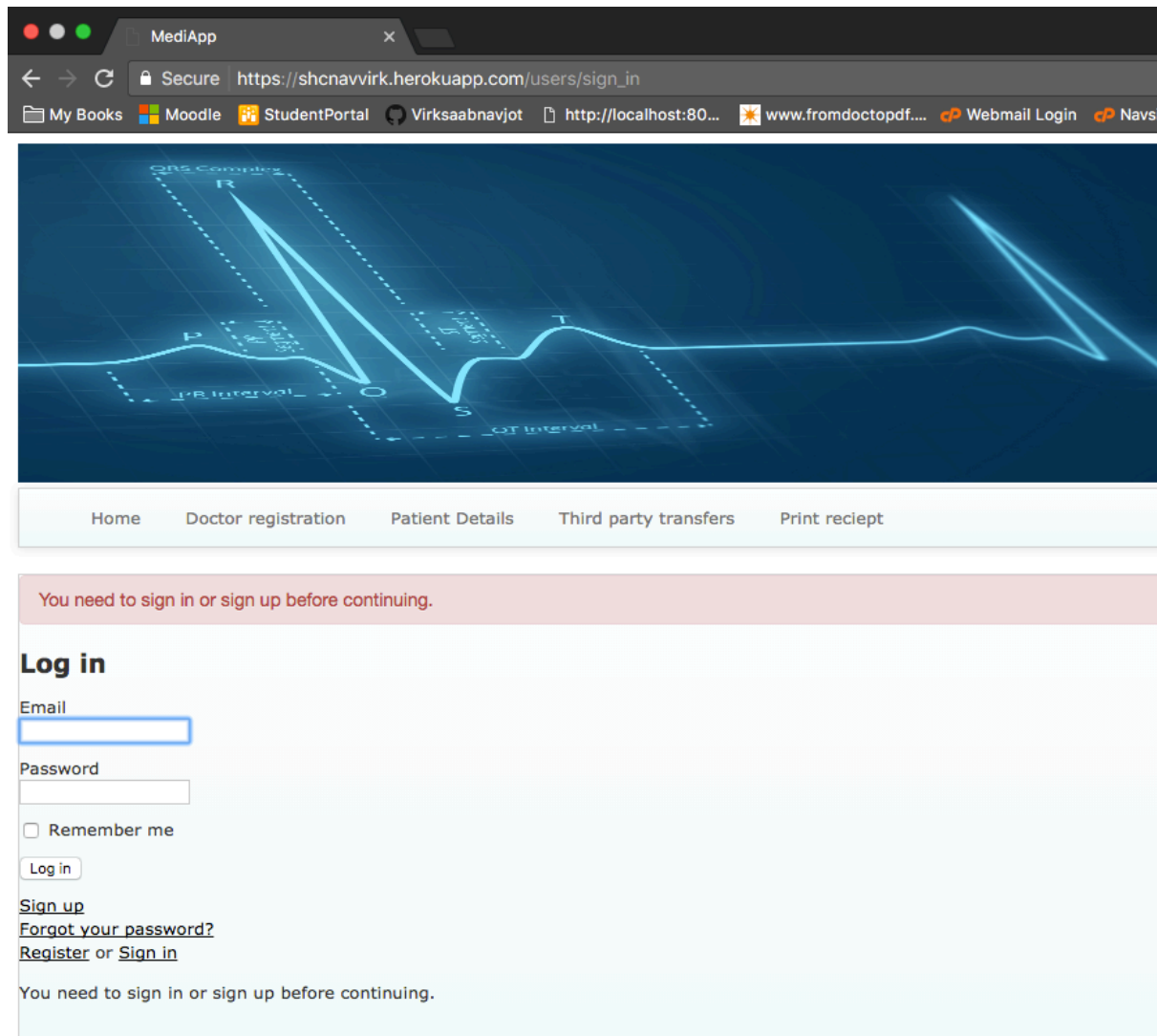
Its implemented in the Model of Patients (Pats)

Model:	pat.rb
View:	N/A
Controller:	N/A

Functionality 5: (Allows) User Registration and Login on the Site

Approach: This functionality is implemented using MVC approach. The functionality is implemented in the following files –

Model:	User.rb
View:	Views/devise folder
Controller:	Sessions_controller.rb

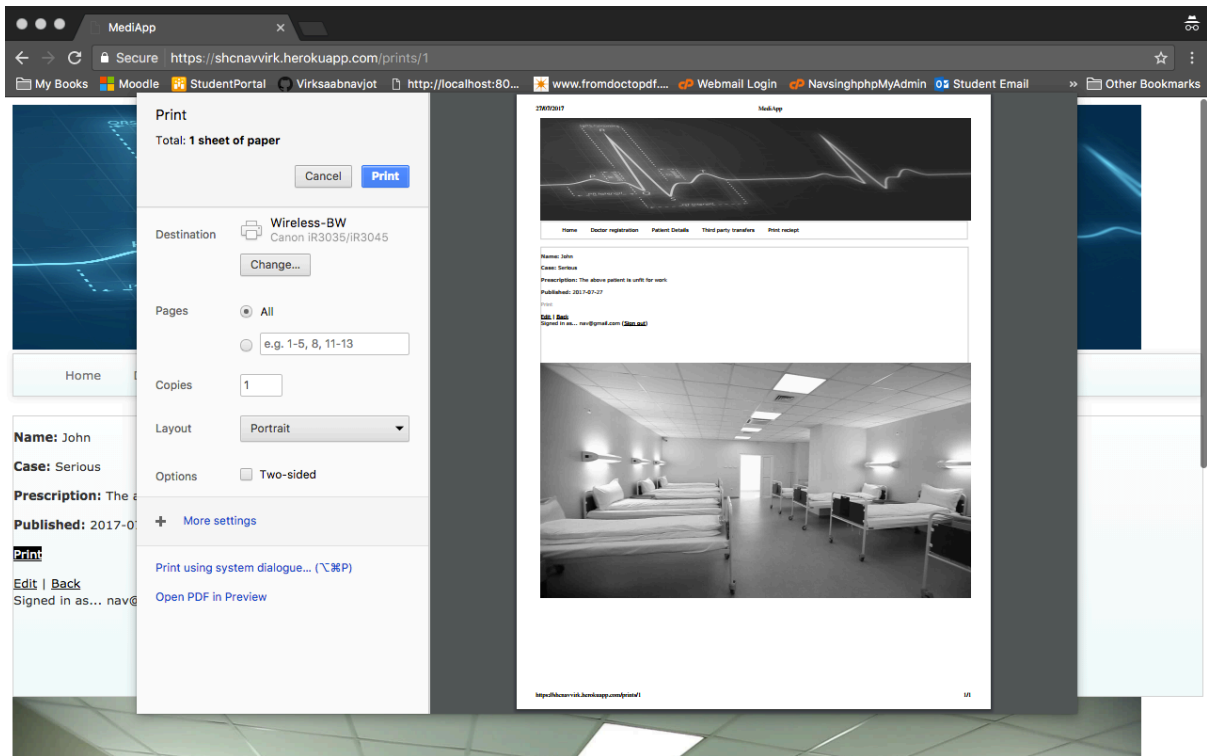


Functionality 6: Prescription Note (Additional Functionality) – Allows the doctor to give prescription to the patient and print it using the print button.

Approach: This functionality is implemented using MVC approach. The functionality is implemented in the following files –

Model:	Print.rb
---------------	----------

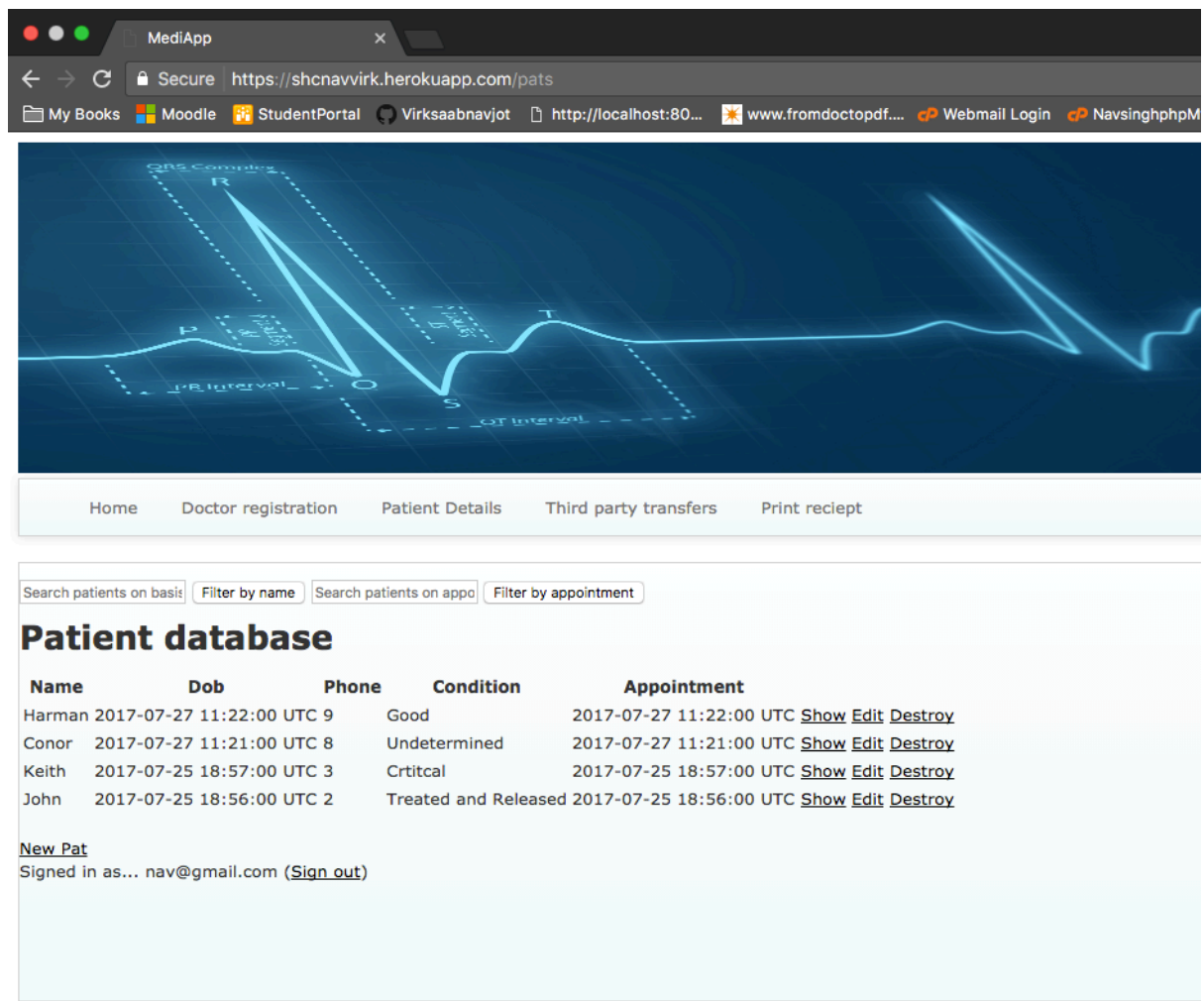
View:	Views/prints folder
Controller:	Prints_controller.rb



Functionality 7: Search and Sort – allows the doctor to search and sort the patient based on name and appointments.

Approach: This functionality is implemented using MVC approach. The functionality is implemented in the following files –

Model:	Pats.rb
View:	Views/pats folder
Controller:	Pats_controller.rb



Feature/Functionality 8: UI Improvements – The GUI of the application

Aim: Allows the user to easily use the system, menu for easy navigation, some images and background color etc.

2.3 Design Patterns

Let's first understand, What is Design Patterns? Then we will understand it in context of software application development.

Let me explain with an example -

For example: Lets, take survival extinct for human beings – Every day, as human beings we all have some common survival needs/problems – like we feel hungry, thirsty etc.

And as we all know, we have universal solutions to these needs like when we feel hungry we eat (unless you are on a diet – well let's not go into that just joking), so we eat, similarly when we drink water when we are thirsty.

So, design patterns is basically a common solutions to problems that basically every person faces, like in the example above as humans for survival we need to eat and drink.

Now, let's understand design patterns in context of software development in this case ruby on rails development and list the once used in this project-

There are two types of Design Patterns used in Rails projects – Observer and Decorator

For this project, I am using Decorator – implemented (lib/transfer_decorator.pat)

The screenshot shows a web browser window with the title 'MediApp'. The address bar shows a secure connection to 'https://shcnnavirk.herokuapp.com/shifts/new'. The browser's bookmark bar includes 'My Books', 'Moodle', 'StudentPortal', 'Virksaabnavjot', 'http://localhost:80...', 'www.fromdoctopdf....', 'Webmail Login', and 'N'. The application's navigation bar contains links for 'Home', 'Doctor registration', 'Patient Details', 'Third party transfers', and 'Print reciept'. The main content area is titled 'Patient transfer' and contains the following form elements:

- Firstname:
- Lastname:
- Firm:
- Cost:
- Description:
- Color:
- X ray: ☒ St. James's Hospital
- Ct scan: ☐ Mater Misericordiae University Hospital
- Endoscopy : ☐ Rotunda Hospital
- Do you have a medical card? : ☐
-
- [Back](#)
- Signed in as... nav@gmail.com ([Sign out](#))

3 Deployment

I am using c9 editor for coding and heroku.com for project deployment.

The project is running at <https://shcnavvirk.herokuapp.com/>

Command used to achieve deployment (c9 , Github, Heroku) –
Using the c9 terminal window,

\$ git status - to check everything is upto date with git

(Git add, commit, push)

\$ heroku login – in order to login to heroku (Provide credentials)

\$ git push heroku master – for pushing/deploying the code to heroku

\$ heroku run rake db:migrate

Deployment complete

4 Appendix A :Project Setup, Problems faced and Details

Date: 11/July/2017, 00:22 (Dublin,Ireland)

Rails Project

IDE: Cloud 9 (<https://c9.io>)

SCM: Github.com

Problem faced: I first cloned the project using https url for the repo and the c9 was not able to commit to github. See fix below -

Used this to clone github repo into Cloud 9:

[git@github.com:Virksaabnavjot/SHC.git](https://github.com:Virksaabnavjot/SHC.git)

and was able to perform all git commands through c9 terminal.

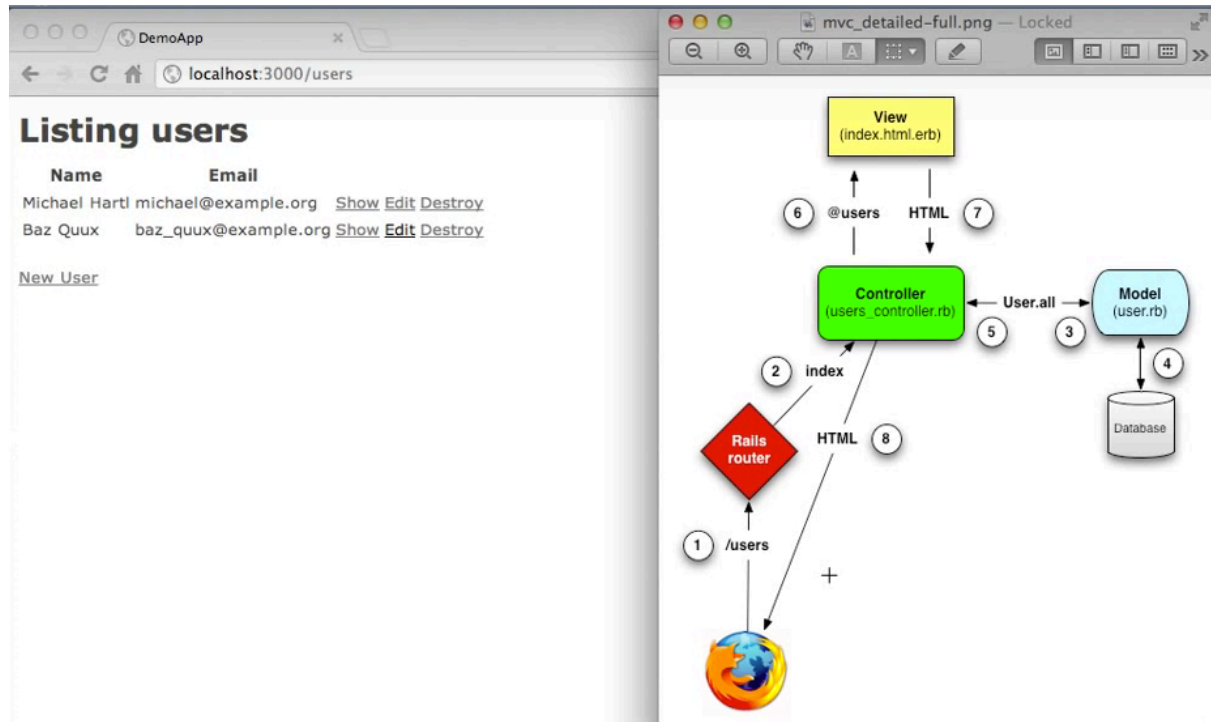
Used this command to create a new rails application: \$ rails new ./

Other problems: included getting into errors and spend hours sometimes in order to solve them like in one case I had major issues with heroku git and deployment.

5 Appendix B :Understanding MVC in Rails Projects

During the ruby on rails learning process found this easy/simple but yet very self explanatory MVC explanation that helped me understand How MVC works in Ruby on Rails projects. I have presented my understanding of it through a short explanation.

Image Source: https://www.youtube.com/watch?v=7Vj0u_q-iEw



[My Explanation on this diagram](#)

Here, our browser makes a request for users (/users) to the rails router, and the router goes to the users_controller class, from there we interact with the Model (that deals with our database) to retrieve all the users in the database (using User.all), once this is done we call the view class (@users) which gives HTML representation to the controller and then the results are returned to the browser in HTML format. Which we can see on the left hand side of the screenshot.

6 Appendix C: References

The project would not have been possible without the help of the listed resource. Intensive care has been taken to reference any third-party resource used while the development of this project – either code or the project report.

Softcover.io. (2017). *Ruby on Rails Tutorial (Rails 5)*. [online] Available at: <https://www.railstutorial.org/book/beginning#sec-deploying> [Accessed 23 Jul. 2017].

YouTube. (2017). *Creating a Rails 5 application with postgres on Cloud 9*. [online] Available at: <https://www.youtube.com/watch?v=JPk5QzRj6XU> [Accessed 13 Jul. 2017].

YouTube. (2017). *Deploying a Ruby on Rails Application to Heroku*. [online] Available at: <https://www.youtube.com/watch?v=mabGJ-vuABc> [Accessed 23 Jul. 2017].

YouTube. (2017). *Ruby on Rails Tutorial: Developing a Demonstration Application*. [online] Available at: https://www.youtube.com/watch?v=7Vj0u_q-iEw [Accessed 23 Jul. 2017].

YouTube. (2017). *Using Rails 5 on Cloud 9 cloud service*. [online] Available at: <https://www.youtube.com/watch?v=kfB2m9j9jvw> [Accessed 12 Jul. 2017].

exist, H. (2017). *Heroku problem : The page you were looking for doesn't exist*. [online] Stackoverflow.com. Available at: <https://stackoverflow.com/questions/7376966/heroku-problem-the-page-you-were-looking-for-doesnt-exist> [Accessed 23 Jul. 2017].

Different types of Patient Conditions are taken from here:

http://www.hopkinsmedicine.org/news/media/patient_condition_updates.html