



COMP9900 Computer Science / Information Technology  
Project

# Project Report

**Tingfeng Lin z5147976(Scrum Master)**

**Yi Xiao z5125251**

**Yousheng Sui z5156244**

**Yan Duan z5145036**

# Contents

<b>1</b>	<b>INTRODUCTION.....</b>	<b>1</b>
1.1	BACKGROUND .....	1
1.2	PROBLEM WITH EXISTING SYSTEM .....	1
1.3	AIMS.....	2
<b>2</b>	<b>PROJECT SCOPE.....</b>	<b>2</b>
2.1	PROJECT REQUIREMENTS .....	2
2.2	PROJECT FEATURES .....	3
<b>3</b>	<b>TECHNICAL IMPLEMENTATION .....</b>	<b>4</b>
3.1	PROJECT STRUCTURE .....	4
3.2	PROJECT DATA.....	5
3.2.1	<i>Data extraction</i> .....	5
3.2.2	<i>Database</i> .....	6
3.2.3	<i>Data flow chart</i> .....	7
3.3	FRONTEND: USER INTERFACE DESIGN .....	7
3.3.1	<i>Frontend module</i> .....	8
3.3.2	<i>Fronted JavaScript framework</i> .....	10
3.4	BACKEND .....	11
3.4.1	<i>Sign in and Login</i> .....	11
3.4.2	<i>User check appointment information</i> .....	12
3.4.3	<i>User personal information update</i> .....	14
3.4.4	<i>User questionnaire form</i> .....	14
3.4.5	<i>User appointment booking</i> .....	15
3.4.6	<i>Doctor appointment checking</i> .....	18
3.4.7	<i>Doctor question setting</i> .....	19
3.4.8	<i>Doctor personal information update</i> .....	19
3.4.9	<i>User/Doctor logout</i> .....	19
3.4.10	<i>Additional feature: Chatbot</i> .....	20
3.5	API DESIGN .....	20
3.6	SYSTEM FUNCTIONALITIES.....	21
3.7	USER AND DEVELOPER DOCUMENTATION.....	21
3.7.1	<i>User documentation</i> .....	21
3.7.2	<i>Developer documentation</i> .....	31
3.8	THIRD PARTY APPLICATION .....	32
3.8.1	<i>Remote database and database visualization tools</i> .....	32
3.8.2	<i>File server system and file convert API</i> .....	33
3.8.3	<i>Google map API</i> .....	34
3.8.4	<i>Email API</i> .....	35
<b>4</b>	<b>CHATBOT.....</b>	<b>35</b>

<b>5    IMPLEMENTATION.....</b>	<b>39</b>
5.1    PERFORMANCE .....	39
5.2    CHALLENGES.....	39
5.2.1 <i>Frontend</i> .....	39
5.2.2 <i>Backend</i> .....	39
5.2.3 <i>Chatbot</i> .....	39
<b>6    PROJECT MANAGEMENT .....</b>	<b>40</b>
6.1    PROJECT METHODOLOGY.....	40
6.1.1 <i>Agile project management</i> .....	40
6.1.2 <i>Scrum software development</i> .....	40
6.2    RISK MANAGEMENT .....	41
6.3    MILESTONES .....	41
<b>7    TEAM PROFILE .....</b>	<b>42</b>
7.1    ROLES.....	42
<b>8    REFERENCE .....</b>	<b>43</b>

# 1 Introduction

## 1.1 Background

With the improvement of people's living standards and the development of medical technology, advanced equipment and instruments play an increasingly important role in disease diagnosis. Medical imaging equipment is an indispensable component of hospital medical equipment, which not only provides important guarantee for clinical diagnosis and treatment, but also provides an important platform for clinical scientific research. What is a medical imaging device? Medical imaging equipment refers to various instruments that use various media as information carriers to reproduce the internal structure of the human body into images. The image information has a spatial and temporal distribution corresponding to the actual structure of the human body. Common medical imaging equipment such as conventional radiology, CT, MR, ultrasound, nuclear medicine, radiation therapy, interventional therapy, etc., these radiological techniques are currently an effective adjuvant treatment for clinical treatment. At present, more and more patients choose to diagnose and cooperate with doctors through some advanced treatment instruments to obtain more accurate and rapid diagnosis and treatment recommendations.

## 1.2 Problem with existing system

At the same time, complex medical examinations require patients to be fully prepared before they see a doctor. Medical imaging diagnosis usually requires different preparations because of different imaging principles and models. Especially related to radiological examination, due to the danger of radiation, clinicians and radiology technical operators should take necessary protective measures for doctors and patients in accordance with relevant regulations.

Many hospitals also provide guidelines for the use of medical devices on their websites, but few people visit them. At the same time, patients are also a group lacking professional medical knowledge. The analysis of specific medical conditions is only a description on the website and cannot meet the needs of patients. Delays or cancellations are often caused by the lack of understanding of the preparation of radiological examinations in many patients. This not only delays the patient's treatment time, but also reduces the efficiency of the radiology instrument. Therefore, how to improve the utilization of medical imaging equipment is a problem faced by patients and hospitals that our team wants to solve.

## 1.3 Aims

In order to better help patients to see patients in a timely manner and improve the efficiency of radiological examination, a smart online booking radiology inspection website came into being. The system can realize online appointment time and doctors, record basic patient information, and the chat robot can provide guidance for patients to prepare for medical treatment. The chat robot is the bridge between the doctor and the patient, understands the patient's condition for the doctor, and makes an online appointment for the patient.

- Enable a user to get the useful tips about physical health

The website will provide basic information about several radiological examinations and preparation guides for reference. Patients can search the website for the appropriate information they want to know.

- Assistant chatbots help patients prepare radiology examination

The assistant robot is provided to answer the patient's personal problem, and the assistant robot can conduct a simple conversation with the patient, ask the patient's physical condition, and prepare the personal information before the examination.

- Help the patients to book a radiology examination

The patients can make a reservation by the assistant robot including choosing the preferred radiologist. This website is a great way to help patients who lack medical expertise to prepare for exams and make appointments.

## 2 Project scope

Regarding the scope of the project, our team has decided to design a Python-based web chatbot. Flask has a RESTful extension that adds support for fast build REST APIs, making front-end and back-end development easier. In addition, using the HTTP or HTTPS protocol, the data transfer format uses JSON.

The website login is divided into two parts, the doctor's side and the patient's side. On the patient side, you can make an appointment to check the doctor and check the time online, fill out the basic information questionnaire online, and make a simple Q&A session with the chat robot to cancel the appointment. At the doctor's end, you can view the basic information form of the patient who has been reserved, set up special questions to communicate to the patient through the chat robot, cancel appointment. Both parts include login registration to change personal basic information and so on.

## 2.1 Project Requirements

For this website, there are a few requirements that need to be completed. The most

important thing is to help patients complete online appointments, followed by chat bots.

**Functionality requirement:**

- Doctors:**
1. View patient basic information questionnaire online.
  2. View, cancel appointment.
  3. Set own special question.

- Patients:**
1. Book, cancel appointment.
  2. Sample dialog with chatbot.
  3. fix personal basic information questionnaire.

**Reliability:**

The system will be accessible 100% of the time

The mean time between failures is required to be no less than one week

The system crash should be extremely low and never loss or leak data.

**Security:**

A password must be entered to use the system

Users can only modify and collect their own data

## 2.2 Project Features

**Code management:** Github

GitHub is the most popular platform for software developer to share, build and discover engineering.

**Program language:** Python, JavaScript, SQL

**Scrum:** Trello

**Testing:** LoadRunner

**Libraries:** Flask, NumPy, Pandas, Pytorch, etc.

NumPy and Pandas are very useful libraries of Python, especially in data science field and Flask contributes a lot in building Web APIs. Compared with TensorFlow, Pytorch is equally powerful and more friendly to deep learning novices.

# 3 Technical implementation

## 3.1 Project structure

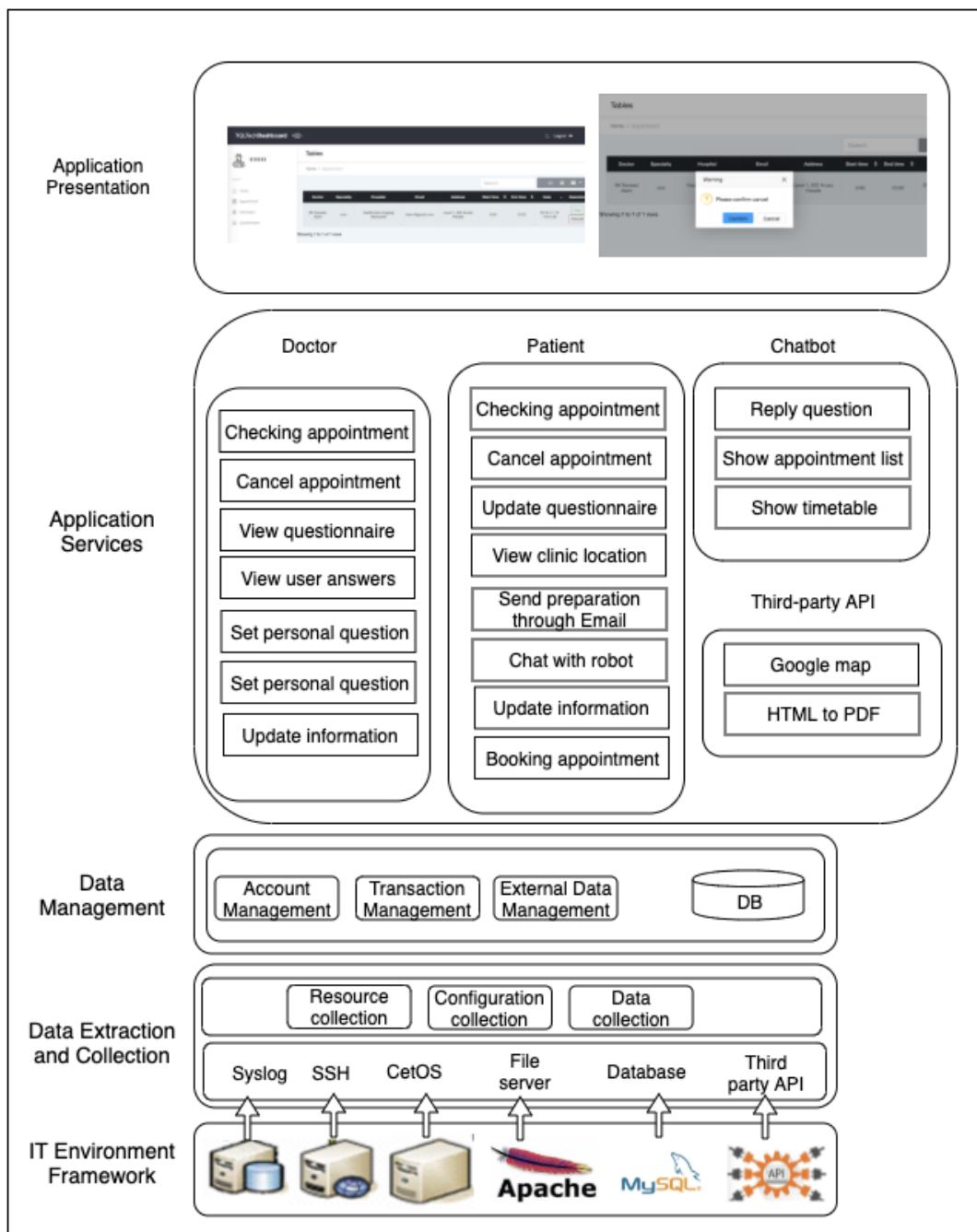


Image 3.1- 1 Project structure

## 3.2 Project data

### 3.2.1 Data extraction

Data extraction function is seen as the preprocessing of the chatbot training. It will extract the data from external data sources. We intend to extract relevant information from different clinics and hospitals and also filter some meaningless data. In order to build a data extraction function, we are going to create a web scraper using the third-party library (BeautifulSoup) to extract all relevant information from websites. The captured data will be stored into a cloud-based database (MySQL).

- In the official website, crawler technology is used to crawl real-time data from the world health organization, including pictures and text information. After that, the obtained data is stored in the database.

```
import scrapy.config as config
from bs4 import BeautifulSoup
import scrapy import_data as import_data
class home_page_news:
    def __init__(self):
        self.__start = 'https://hospitalnews.com/{'
        self.__header = {
            'Referer': 'https://hospitalnews.com',
        }
        self.__res = config.R(headers=self.__header)
        self.__home_page_news_urls = []
        self.__home_page_news_info = []
    def run(self):
        self.get_home_page_urls()
        for url in self.__home_page_news_urls:
            self.get_home_page_info(url)
        # print(self.__home_page_news_info)
    def get_home_page_info(self,url):
        content=self.__res.get_content(url,'utf-8')
        soup=BeautifulSoup(content,'html.parser')
        img_url=soup.find(class_='entry-thumb td-modal-image').attrs['img']
        if img_url is None:
            img_url=''
        second_title=soup.find(class_='wp-caption-text')
        if second_title is None:
            second_title=''
        else:
            second_title=second_title.text
        title=soup.find(class_='entry-title')
        if title is None:
            title=''
        else:
            title=title.text
        d=dict()
        d['title']=title
        d['second_title']=second_title
        d['img_url']=img_url
        # print(d)
        import_data.db().import_home_page_news_data(d)
```

Image 3.2.1- 1 Data extraction code

id	title	second_title	img_url
40	New treatment for bipolar de	Dr. Fidel Vila-Rodriguez with t	https://i1.wp.com/hospitaln
41	New options for those living	The Stepped Care Model supp	https://i0.wp.com/hospitaln
42	Canada ranks last on number		https://i2.wp.com/hospitaln
43	Stitching together new evide		https://i2.wp.com/hospitaln
44	Ontario Shores celebrates a	Karim Mamdani, President anc	https://i1.wp.com/hospitaln
45	Reacting to real-time data is	David Musyj is the President a	https://i0.wp.com/hospitaln
46	A modern cancer plan to red		https://i0.wp.com/hospitaln
47	Health care transformation –	Julia Hanigsberg	https://i2.wp.com/hospitaln
49	Managing crystal meth withd		https://i2.wp.com/hospitaln
50	Genome-wide sequencing fo		https://i0.wp.com/hospitaln
51	A new wound-care initiative i	The care team at Providence F	https://i0.wp.com/hospitaln
52	Wound care: What's the stor		https://i1.wp.com/hospitaln
53	Wound care revolution: Put a	Dr. Sheila Wang and Dr. Greg I	https://i1.wp.com/hospitaln
54	Study shows where C. diffici	The canine detection team hel	https://i2.wp.com/hospitaln
55	Challenging the process for c		https://i0.wp.com/hospitaln
56	SickKids-led research team i		https://i1.wp.com/hospitaln
57	A path to wellness		https://i1.wp.com/hospitaln
58	Paid to play	RN Richard Booth uses robots	https://i2.wp.com/hospitaln
59	Teach. Don't preach.	Mireille "Mimi" Mitchell.	https://i1.wp.com/hospitaln
60	First patient in Canada treat		https://i1.wp.com/hospitaln
61	Surgeons insert 3D ankle boi		https://i0.wp.com/hospitaln
62	First-in-Canada digital syste	MSH Chief of Paediatrics, Dr. I	https://i2.wp.com/hospitaln
63	New treatment for bipolar de	Dr. Fidel Vila-Rodriguez with t	https://i1.wp.com/hospitaln
64	New options for those living	The Stepped Care Model supp	https://i0.wp.com/hospitaln
65	Ontario Shores celebrates a	Karim Mamdani, President anc	https://i1.wp.com/hospitaln
66	The Canadian Health Informa	2019 CHIA winners.	https://i2.wp.com/hospitaln
67	The potential of e-Health to i		https://i0.wp.com/hospitaln

Image 3.2.1- 2 Database

- In order to make the system more realistic, both doctor information and clinic information are obtained from real websites. Doctor information includes doctor's real name, doctor's profile, doctor's photo and other information. Clinic information includes clinic address, clinic phone number, clinic postcode, etc.

```

class InfoList:
    def __init__(self):
        self.__start='https://www.healthcareimaging.com.au/state-new-south-wales'
        self.__header={'Referer':'https://www.healthcareimaging.com.au/state-new-south-wales',
                      'Host':'www.healthcareimaging.com.au'}
        self.__res=config.R(headers=self.__header)
        self.__urls=[]
        self.__clinic_urls=[]
        self.__clinic_info_list=[]
    def run(self):
        print("*"*100)
        while True:
            if len(self.__urls) > 0:
                last_url = self.__urls.pop()
                content = self.__res.get_content(last_url, 'utf-8')
                if content is not None:
                    self.get_clinic_urls(content)
                else:
                    break
        while True:
            if len(self.__clinic_urls)>0:
                last_clinic_url=self.__clinic_urls.pop()
                print('fetching {}'.format(last_clinic_url))
                content=self.__res.get_content(last_clinic_url, 'utf-8')
                if content is not None:
                    self.get_clinic_detail(content)
                else:
                    break
        print('start writing to data.txt','*'*100)
        with open('../data.txt','w') as f:
            f.write(str(self.__clinic_info_list))
        print('end writing','*'*100)
    def get_clinic_urls(self,content):
        base_url='https://www.healthcareimaging.com.au'
        content=str(content)
        href_pattern=re.compile('<h2 class="list_direction_heading"><a href="(.*?)>(.*?)(?=<)')
        href=re.findall(href_pattern,content)
        [self.__clinic_urls.append(base_url+item[0]) for item in href]
    def get_clinic_detail(self,content):
        content=str(content)
        soup=BeautifulSoup(content,'html.parser')
        location=(soup.find(class_="loc_temp_details_txt"))
        location=location.p.text
        if location.find('\r\n')!=-1:
            index=location.index('\r\n')

```

Image 3.2.1- 3 Data extraction code

id	address	name	phone	v	suburb	postco
175	308-312 Beamish Street	Campsie Imaging Centre	(02) 9787-1011	Campsie	t	2194
176	178 Queen Street	Medical Imaging Campbelltown	(02) 4621-9000	Campbell	2560	
177	178 Queen Street	Macarthur Nuclear Medicine	(02) 4621-9030	Campbell	2560	
178	14 Warby Street	Macarthur Diagnostic Imaging	(02) 4625-3900	Campbell	2560	
179	14 Warby Street	MRI Campbelltown	(02) 4625-8531	Campbell	2560	
180	296 Queen Street	Campbelltown Healthcare Services	(02) 4627-4941	Campbell	2560	
181	Ground Floor, 105 Fren	Northern Beaches Medical In	(02) 9470 5200	Frenchs	F	2086
182	27 Breckenridge Street	Mid North Coast Diagnostic I	(02) 6554-8600	Forster	N	2428
183	10 Nelson Street	Fairfield Imaging Centre	(02) 9726-4295	Fairfield	N	2165
184	62 Windsor Parade	Western Plains Imaging	(02) 6841-9060	Dubbo	NS	2830
185	168 Brisbane Street	Orana Radiology	(02) 6841-0700	Dubbo	NS	2830
186	213-219 Darlinghurst Road	Darlinghurst Imaging Centre	(02) 8302-1130	Darlinghurst	2010	
187	Cnr Princes Highway & Dapto	Dapto Imaging Centre	(02) 4262-4560	Dapto	NS	2530
188	46-50 Underwood Street	Corrimonal Imaging Centre	(02) 4258-0250	Corrimonal	N	2518
189	Suite 6, 105-119 Llongs	Rayscan Chipping Norton	(02) 9600-7470	Chipping	I	2170
190	270 Victoria Avenue	Chatswood Imaging Centre	(02) 9413-2691	Chatswood	2067	
191	30-38 Short Street	Leichhardt Imaging Centre	(02) 9561-3360	Leichhardt	2040	
192	Suite 12, 16 Laurie Street	Mid North Coast Diagnostic I	1800 620 881	Laurieton	2443	
193	15 Kensington Street	St George MRI	(02) 9587-6122	Kogarah	N	2217
194	15 Kensington Street	Kogarah Imaging	(02) 9146-5850	Kogarah	N	2217
195	Level 2, 1 South Street	St George Private MRI	(02) 9598-5450	Kogarah	N	2229
196	St George Private Hospital	Bryant Radiology	(02) 9598-5440	Kogarah	N	2217
197	Level 2, 35 Belgrave Street	Mid North Coast Diagnostic I	1800 620 881	Kempsey	2440	
198	Level 1, 2 Nardoo Street	Ingleburn Imaging Centre	(02) 9829-3281	Ingleburn	2565	
199	6 Ormonde Parade	Hurstville Imaging Centre	(02) 9570-1788	Hurstville	2220	
200	37 Gloucester Road	Hurstville Private Imaging	(02) 9579-7762	Hurstville	2220	
201	83 Dalton Street	Central West Radiology	(02) 6362-1467	Orange	N	2800
202	Ewing Street	QDI Murwillumbah Hospital	(02) 6672-3560	Murwillumbah	2484	
203	Suite 3, 145 Church Street	Mudgee Radiology	(02) 6372-6544	Mudgee	N	2850
204	22 Jirrang Close	Mt Druitt Imaging	(02) 9881-5260	Mt Druitt	I	2770
205	14 McFarlane Street	Merrylands Imaging Centre	(02) 9682-4100	Merryland	2160	
206	23 Merewether Street	Lingard Imaging	(02) 4964-2700	Merewether	2291	
207	342 Marrickville Road	Marrickville Imaging Centre	(02) 9560-0166	Marrickville	2204	
208	Shop 92, 34 Victoria Road	Marrickville Metro Medical Centre	(02) 9565-4020	Marrickville	2204	
209	Level 1, 832 Anzac Parade	Healthcare Imaging Maroubra	(02) 9344-7545	Maroubra	2035	
210	41-43 Goulburn Street	Rayscan Liverpool	(02) 9731-7100	Liverpool	2170	
211	33 Belmont Street	Sutherland Diagnostic Imaging	(02) 9521-5322	Sutherland	2232	

Image 3.2.1- 4 Database

### 3.2.2 Database

This project uses remote MySQL database which is deployed on the Alibaba remote server using CentOS system. The database scheme contains doctor table, patient table and so on. The data model is shown below.

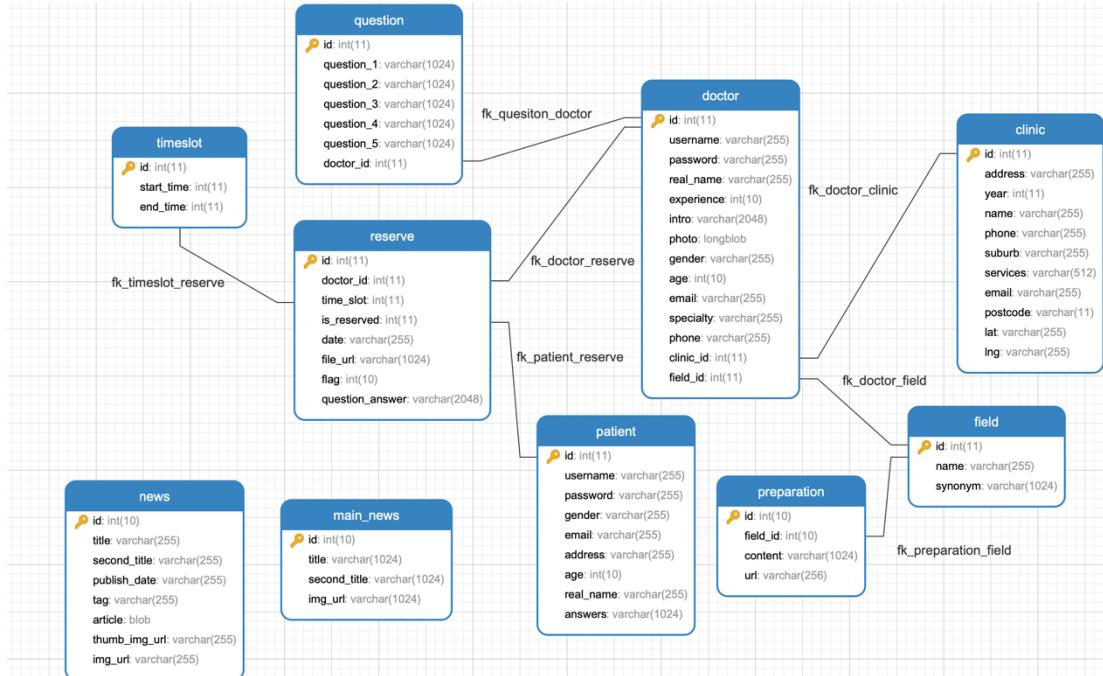


Image 3.2.2- 1 Database scheme

### 3.2.3 Data flow chart

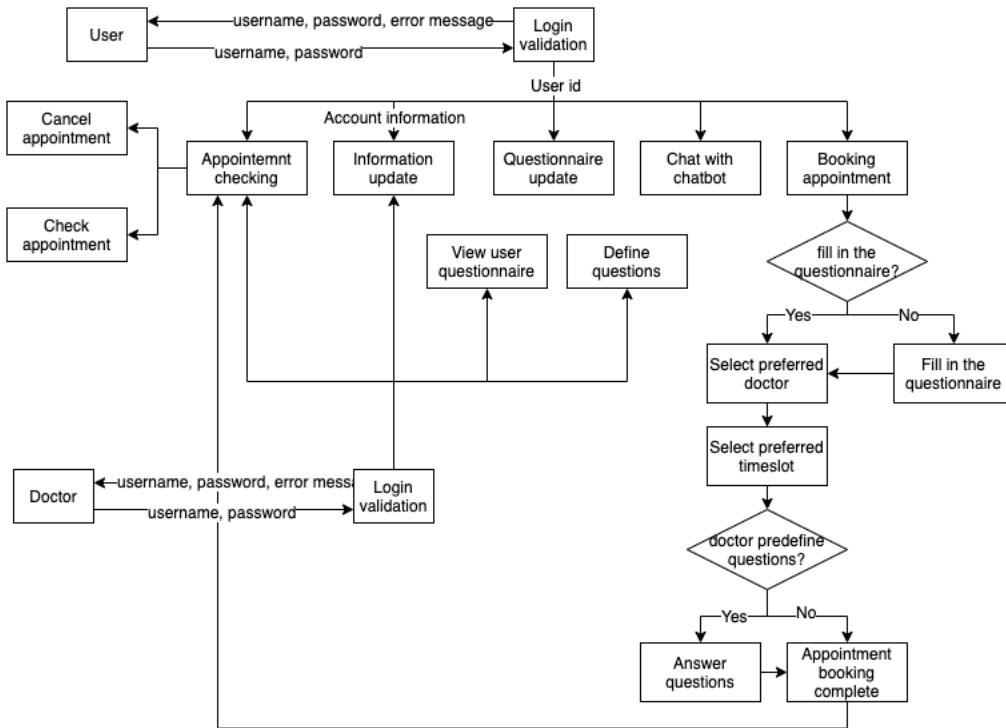


Image 3.2.3- 1 Data flow chart

### 3.3 Frontend: user interface design

The UI module is responsible for handling user's interaction and getting the response from the backend to the frontend. In order to make this project more realistic, we would like to make an official website for this project.

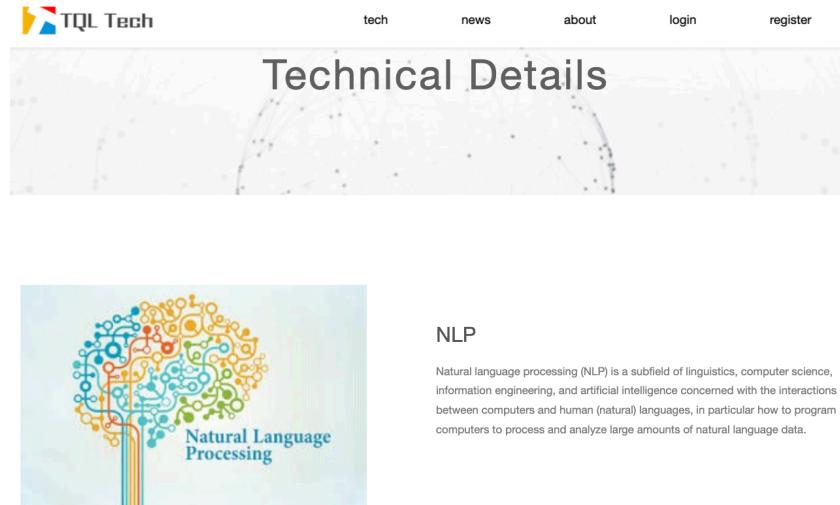
The official website contains five parts which are technical, news, about, login and register.

- For the tech detail tab, some key techniques are introduced such as Natural language processing, python scrapy, flask and so on.
- In news tab, real time news information is retrieved from WHO.
- The about tab shows the developers information and responsibilities information.
- The login part is divided into two login type, the doctor login portal and user (patient) login portal.
- Registration tab is used to sign in a user account for user.

In frontend part, it is based on bootstrap framework. In addition, some third-party JavaScript framework are implemented such as layer.js, validation.js, bootstrap-table.js and so on.

### 3.3.1 Frontend module

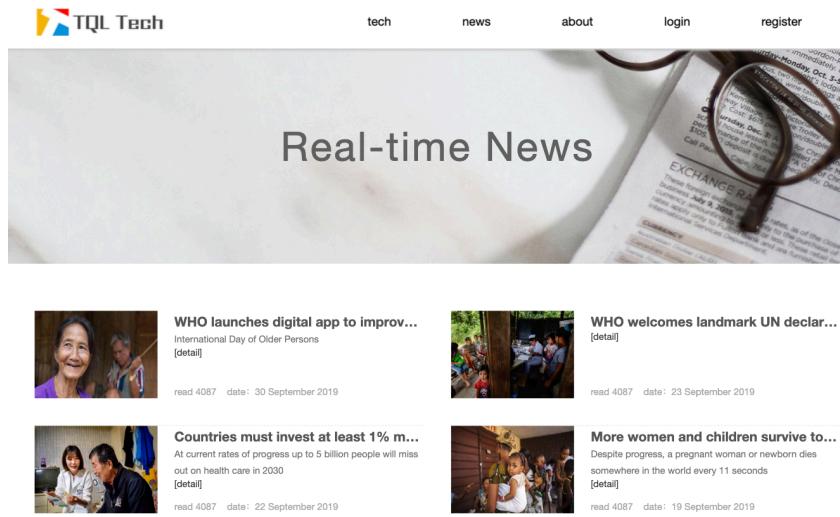
- Technical Tab



*Image 3.3.1- 1 Technical detail page*

In this part, it mainly focuses on the key technical detail of the system and introduce the them in this page.

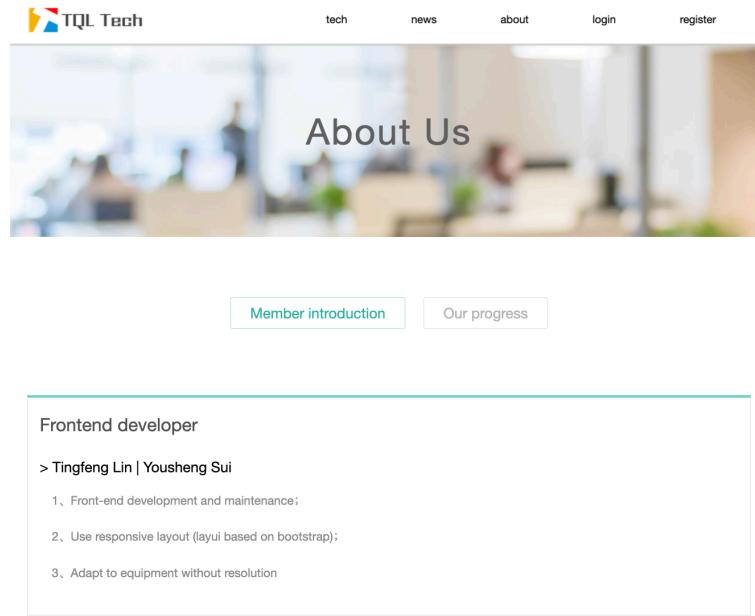
- News tab



*Image 3.3.1- 2 News page 1*

In this page, the front-end users ajax technique to make a request to the data stored in the database using python scrapy. After retrieving recodes from the database, data is rendered on the web page.

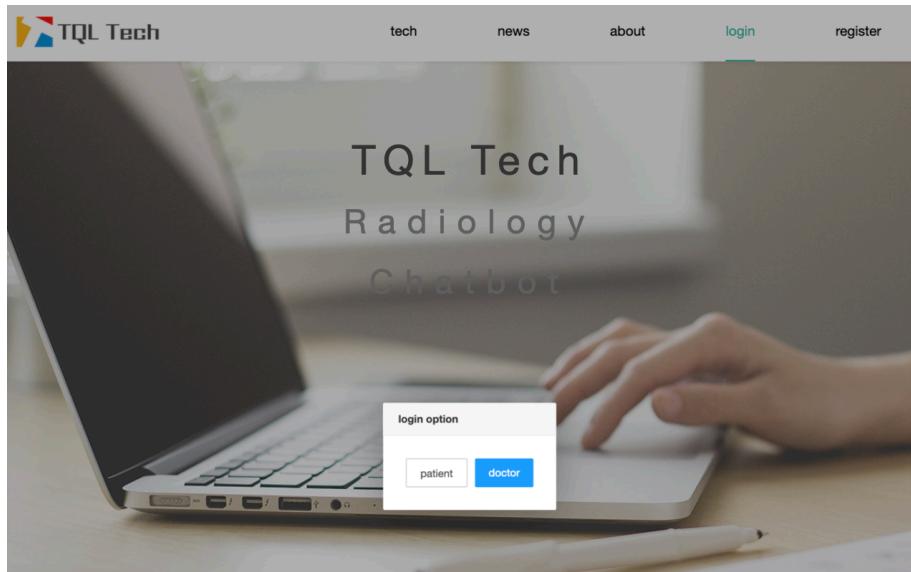
- About tab



*Image 3.3.1- 3 About page*

This page is used to display information about team members and the tasks they are responsible for. In addition, we will update the task bar based on the tasks completed each week.

- Login tab



*Image 3.3.1- 4 Login page*

Clicking the login page button, the inquiry box will pop up. It divided into user login page and doctor login page. This is used to manage permissions.

- Registration tab

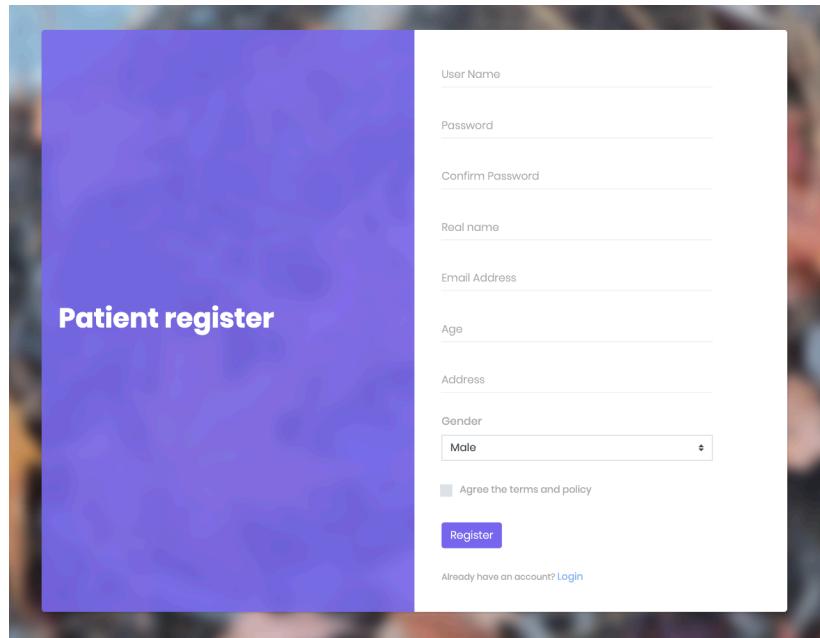


Image 3.3.1- 5 Registration page

This page is for user registration. Users need to enter legal information according to the system prompt.

### 3.3.2 Fronted JavaScript framework

- layer.js

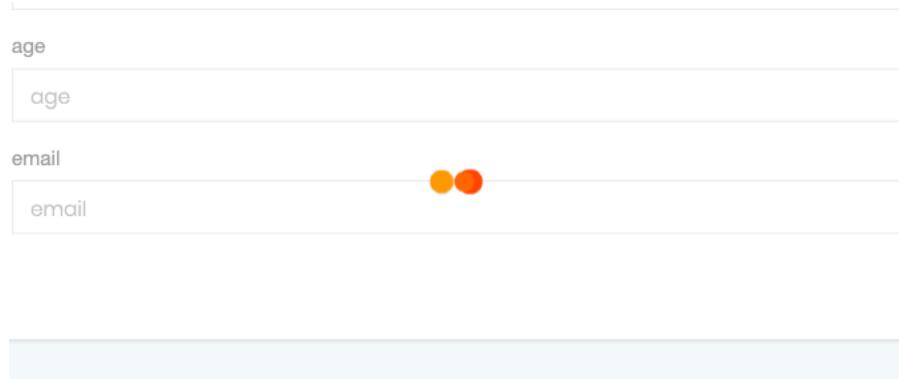


Image 3.3.2- 1 layer.js loading icon

This project mainly users the pop-up layer effect in the framework. To prevent the user from operating when the user clicks the button. Visually gives the user the effect of being loaded. Make web pages more interactive.

- jquery-validation.js

User Name  
d  
At least 6 characters

Password  
•  
At least 6 characters

Confirm Password  
•  
Please enter at least 6 characters.

Real name

Email Address  
e  
Please enter a valid email address.

*Image 3.3.2- 2 Validation.js rules checking*

The framework can automatically detect the user's input, which uses in the user registration function and user information update function. In addition, the designer is able to customize detection rules and custom prompt messages.

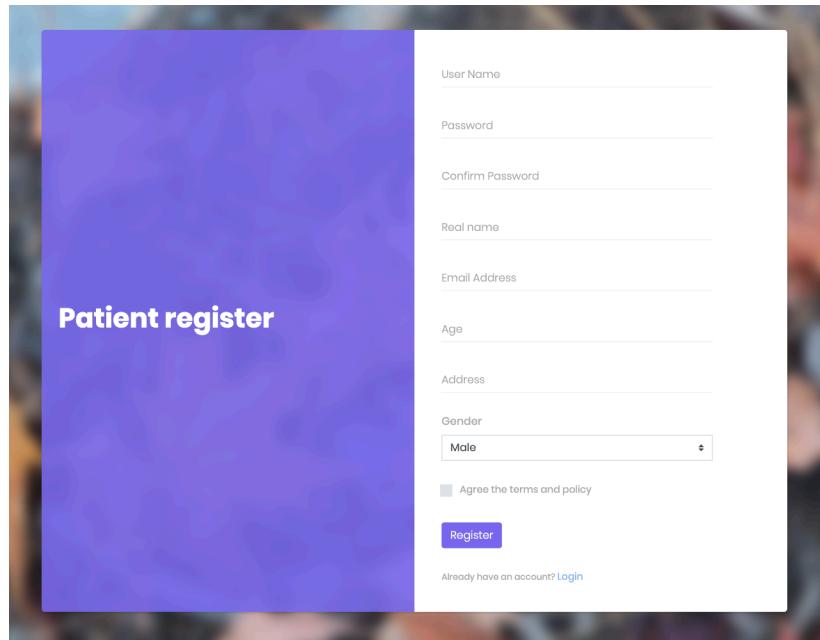
- **bootstrap.js**

The project frontend is based on Bootstrap framework. Bootstrap's responsive CSS is adaptive to desktops, tablets, and mobile phones. It provides a concise and unified solution for developers to create interfaces. It includes powerful built-in components and is easy to customize. It also provides web-based customization.

## **3.4 Backend**

### **3.4.1 Sign in and Login**

On our homepage, if the user has an account, click on the Login button to select as a patient or doctor. If you log in as a patient and enter the correct account number and password, the user can use all the features of our website. If login as a doctor, the user can have additional ability to check his/her schedule and more.



*Image 3.4.1- 1 Sign in page*

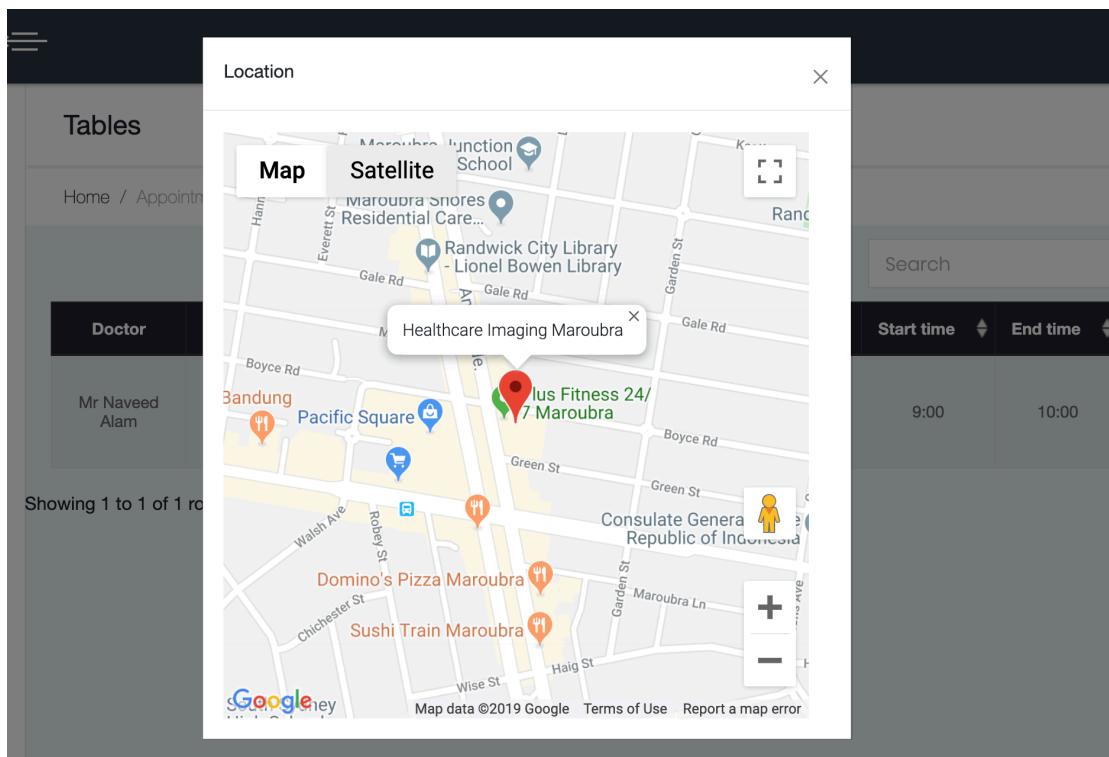
If the user does not have an account, he/she will be treated as a guest. Guests have limited ability to use the browser as a website to find information as our introduction. In addition, users can click on the sign-up button to create their own account, and in a few simple steps, users will be able to fully utilize our website features.

### 3.4.2 User check appointment information

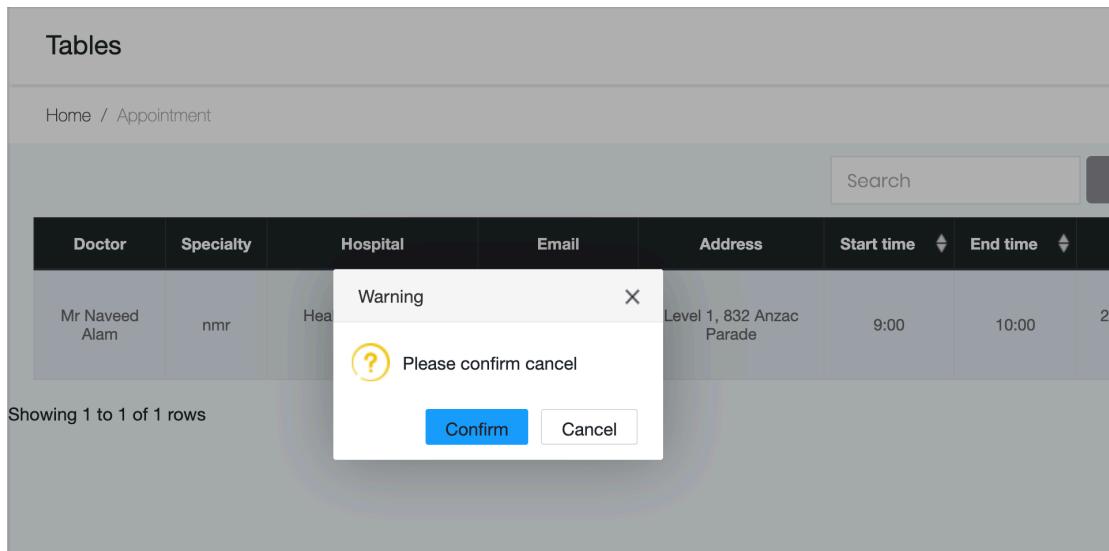
A screenshot of a dashboard titled "TQLTechDashboard". On the left, there's a sidebar with a user icon and the ID "111111". Below this are sections for "MAIN", "Appointment", "Information", and "Questionnaire". The main area is titled "Tables" and shows a table for "Appointment". The table has columns: Doctor, Specialty, Hospital, Email, Address, Start time, End time, Date, and Operations. One row is visible: "Mr Naveed Alam", "nmr", "Healthcare Imaging Marouba", "Alam@gmail.com", "Level 1, 832 Anzac Parade", "9:00", "10:00", "2019-11-13 14:07:08", with "View" and "Cancel" buttons at the end of the row. A search bar and filter icons are at the top of the table. At the bottom, it says "Showing 1 to 1 of 1 rows".

*Image 3.4.2- 1 User appointment checking page*

User can check the appointment that has been booked. In this page, user can search any key word in the searching area then the table will reply for an answer. In addition, if user want to hide some information of the appointment, click the rightmost on the tool bar, some feature can be hidden. This table can be sorted according to user's preference.



*Image 3.4.2- 2 Clinic location*



*Image 3.4.2- 3 Appointment cancel operation*

Moreover, user can even check the location of the clinic by clicking the view button on the operation cell and cancel the appointment by clicking the cancel button.

### 3.4.3 User personal information update

The screenshot shows the 'Personal information' section of the TQLTechDashboard. On the left, there is a sidebar with a user icon and the ID '111111'. Below it, under 'MAIN', are links for 'Home', 'Appointment', 'Information', and 'Questionnaire'. The main content area has a header 'Personal information' and a breadcrumb 'Home / Information'. It contains a 'Account information' form with fields for 'username' (111111), 'password' (redacted), 'realname' (111111), 'age' (1), 'email' (25147976@ad.unsw.edu.au), and 'address' (1). A 'submit' button is at the bottom right.

Image 3.4.3- 1 User information update

First, user information might be reloaded on each cell then users can update their information after validating by the system.

### 3.4.4 User questionnaire form

The screenshot shows the 'Personal questionnaire' section of the TQLTechDashboard. The sidebar and main navigation are identical to the previous screenshot. The main content area has a header 'Personal questionnaire' and a breadcrumb 'Home / Questionnaire'. It contains a 'Account information' form with a 'RADIOLOGY QUESTIONNAIRE' section. This section includes a 'Patient Information' note about iodine-containing contrast material. It then lists several medical history questions with radio button options: 'Are you pregnant or breast feeding?' (radio buttons for yes/no), 'Are you allergic to any medications?' (radio buttons for yes/no), 'Are you asthmatic and are on daily inhalers?' (radio buttons for yes/no), 'Have you had a previous injection of IV contrast medium?' (radio buttons for yes/no), 'Have you been pre-medicated for today's CT exam?' (radio buttons for yes/no), 'Do you have high blood pressure?' (radio buttons for yes/no), and 'Do you have diabetes?' (radio buttons for yes/no).

Image 3.4.4- 1 User questionnaire page

For each user, a questionnaire must be filled before booking an appointment. This questionnaire is used to help the doctor to know more about the patient. Users can update their questionnaire answers according to their actual situation.

### 3.4.5 User appointment booking

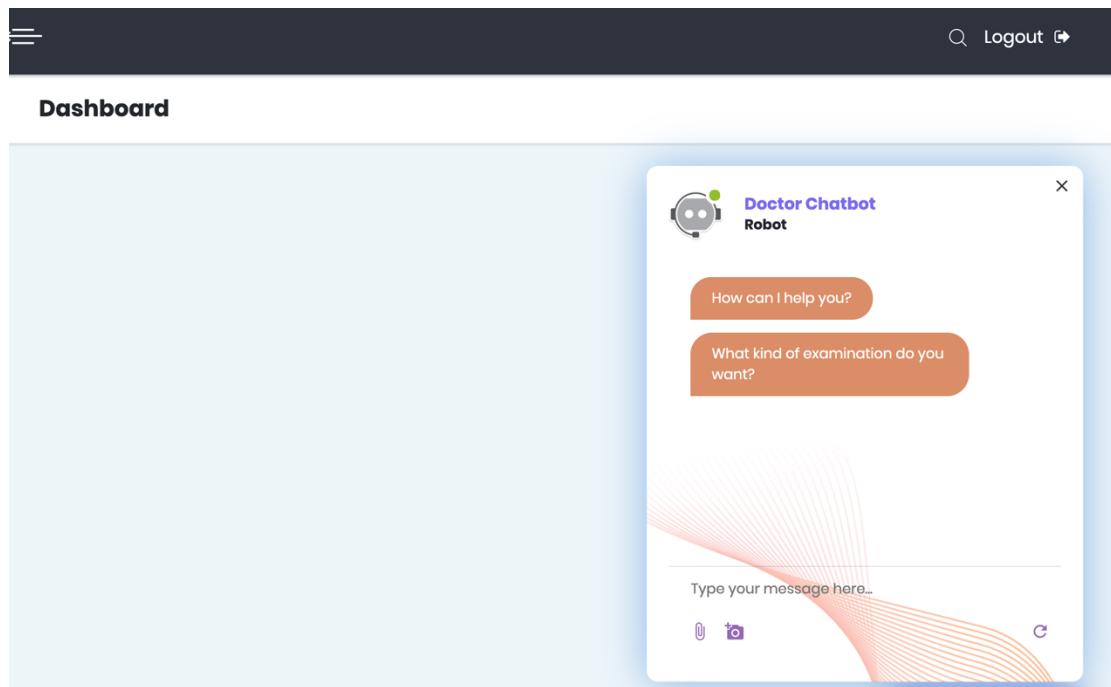


Image 3.4.5- 1 Chatbot

In the user main page, user can chat with the robot by typing specific question such as greeting, general questions and appointment related questions. The robot can handle whatever user's input.

When user typing the question or message that the robot cannot be identified, the robot will reply information that ask user to type another question.

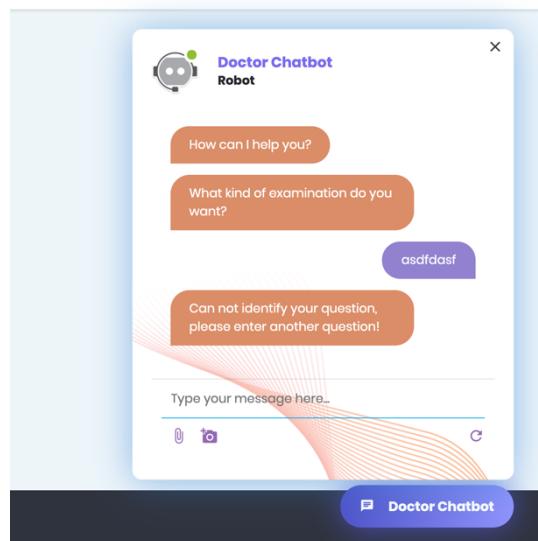


Image 3.4.5- 2 Chatbot

When user typing the question or message that the robot cannot be identified, the robot will reply information that ask user to type another question.

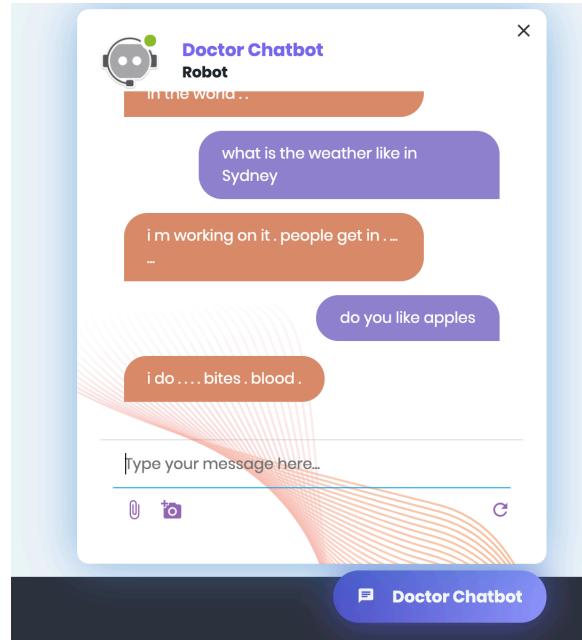


Image 3.4.5- 3 Chatbot

When user typing general question such as Do you like apples? The robot will reply its answer by using NLP network.

The screenshot shows a table of appointment bookings and a Doctor Chatbot interface. The table has columns: Address, Clinic, Real name, Gender, and Operation. The rows show:

- Address: Level 1, 832 Anzac Parade, Clinic: Healthcare Imaging Maroubra, Real name: Dr Barry Leaney, Gender: male, Book button.
- Address: Level 2, 1 South Street, Clinic: St George Private MRI, Real name: Dr Michael Martin, Gender: male, Book button.
- Address: Ewing Street, Clinic: QDI Murwillumbah Hospital, Real name: Dr Pat Page, Gender: male, Book button.
- Address: Rouse Hill Town Medical Centre, Clinic: Rouse Hill Radiology, Real name: Dr Hayden Prime, Gender: male, Book button.
- Address: 15 Kensington Street, Clinic: Kogarah Imaging, Real name: Dr Brooke Sawyer, Gender: male, Book button.

At the bottom, it says "Showing 1 to 5 of 7 rows" and has page navigation buttons (1, 2, 3).

The Doctor Chatbot overlay has sections: "Preparation Links" (Resource 1), "Send these information to email!", "Check location on the map.", and a "Map" button. It also has a "Type your message here.." input field and a "Doctor Chatbot" button.

Image 3.4.5- 4 Chatbot and appointment list

When user typing specific question that related to booking appointments. The robot will reply preparation information, appointment list, location information and so on.

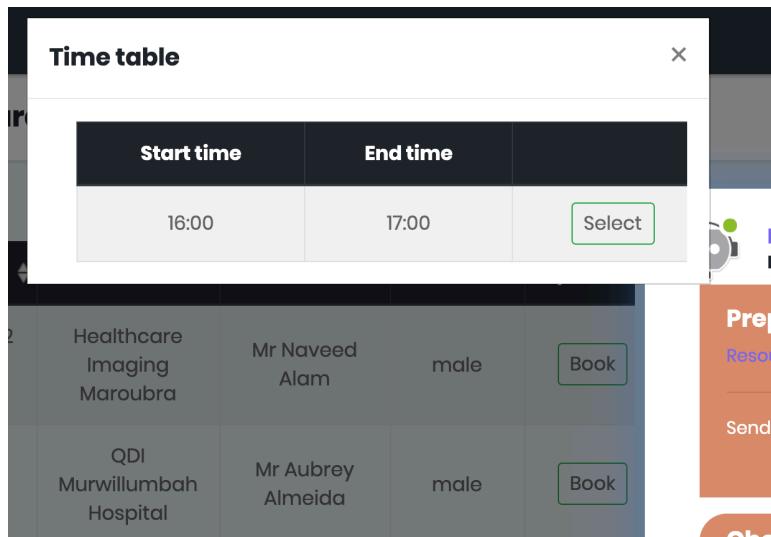


Image 3.4.5- 5 Timetable

By clicking the Book button, a timeslot window will be pop up. It will show the available timeslot and user can select the prefer time.

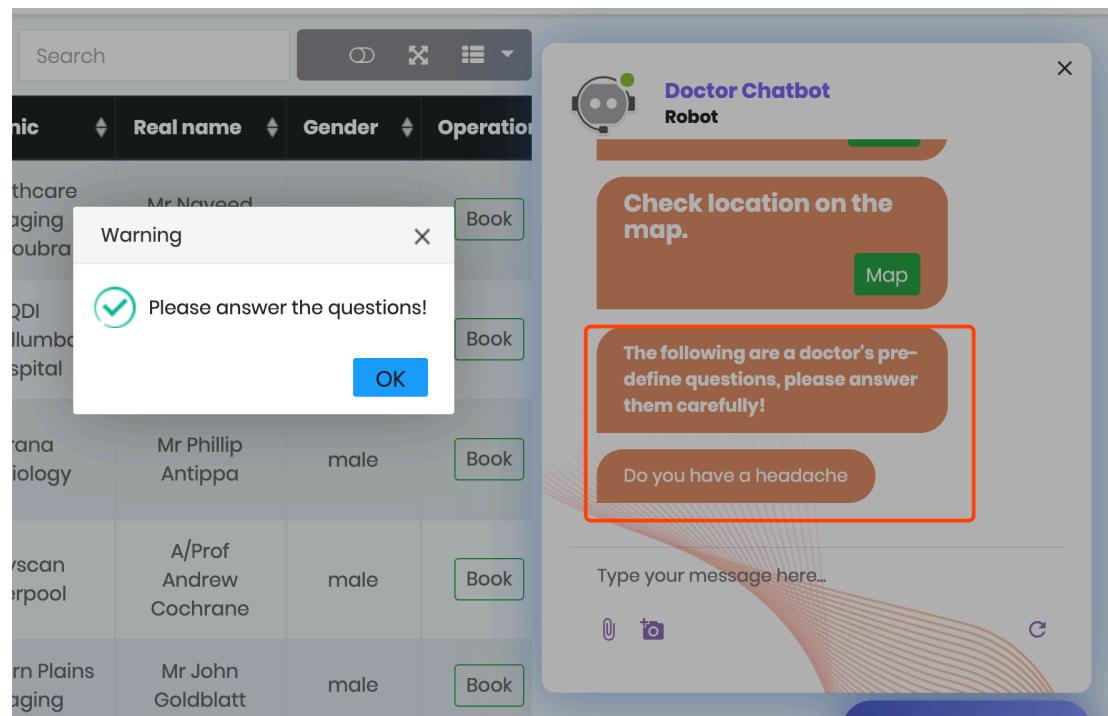
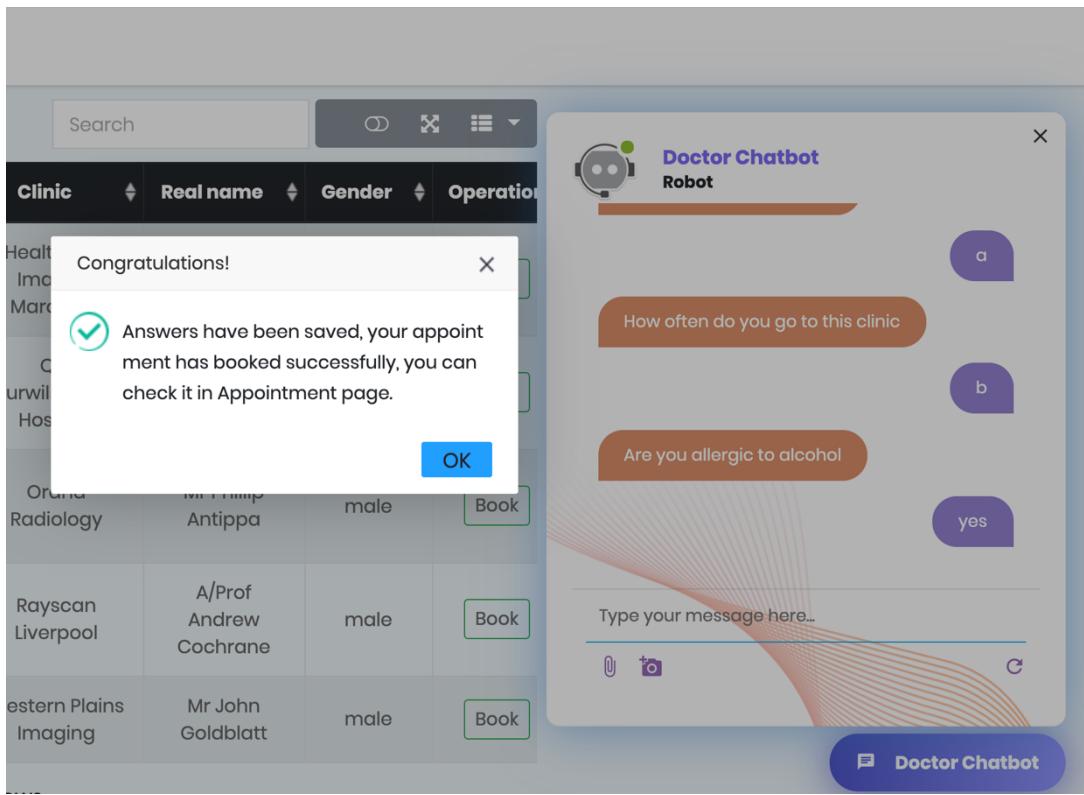


Image 3.4.5- 6 Doctor questions

After the patient selects the appointment, the chat robot will first send a basic information questionnaire to the patient. After the form is confirmed, if the selected doctor has defined questions, the user must answer these questions to finish the booking process.



*Image 3.4.5- 7 Appointment booking complete*

When answering all related questions. The booking appointment process is all finish, then the patient will be notified by email.

### 3.4.6 Doctor appointment checking

Home / Appointment									
Name	Gender	Age	Email	Address	Start time	End time	Date	Operation	
11111	male	1	z5147976@ad.unsw.edu.au	1	16:00	17:00	2019-11-19 15:26:41		<span style="border: 1px solid green; padding: 2px;">Form</span> <span style="border: 1px solid blue; padding: 2px;">Question</span> <span style="border: 2px solid red; padding: 2px;">Cancel</span> <span style="border: 1px solid cyan; padding: 2px;">Finish</span>
Rowing 1 to 1 of 1 rows									

*Image 3.4.6- 1 Doctor appointment checking page*

In the doctor appointment booking page, the doctor can view the information of patients who have been booked. The doctor can view the user's questionnaire, see the answer of the defined questions, delete the appointment and complete the appointment

### 3.4.7 Doctor question setting

The screenshot shows the 'Question' section of the TQLTechDashboard. On the left sidebar, there's a user profile for 'Alam.Naveed' and a navigation menu with 'MAIN' selected, containing links for Home, Appointment, Question, and Information. The main content area has a header 'Question' and a breadcrumb 'Home / Question'. Below this is a section titled 'Basic Question' with the instruction 'Please fill in the questions.' Three input fields are provided: 'Question\_1' (text: 'Do you have a headache'), 'Question\_2' (text: 'How often do you go to this clinic'), and 'Question\_3' (text: 'Are you allergic to alcohol'). At the bottom are 'Add more', 'Cancel', and a purple 'submit' button.

Image 3.4.7- 1 Doctor questions setting page

Doctors have the right to set questions about their field. Each doctor can set up to maximum five questions. Doctors can also modify and delete problems according to their own needs. Users must answer questions before making an appointment. The appointment cannot be completed until the questions are answered.

### 3.4.8 Doctor personal information update

The screenshot shows the 'Personal information' section of the TQLTechDashboard. The sidebar is identical to the previous screenshot. The main content area has a header 'Personal information' and a breadcrumb 'Home / Information'. Below this is a section titled 'Account information' with fields for 'username' (text: 'Alam.Naveed'), 'password' (text: '\*\*\*\*\*'), 'realname' (text: 'Mr Naveed Alam'), 'age' (text: '53'), and 'email' (text: 'Alam@gmail.com'). A purple 'submit' button is at the bottom right.

Image 3.4.8-1 Doctor questions setting page

Doctor can update the personal information according to the actual situation.

### 3.4.9 User/Doctor logout

After the user logs out, the information stored in session and local storage are cleared

to avoid personal information leakage.

### 3.4.10 Additional feature: Chatbot

There is a well-trained deep learning network model and an NLP module for particular tasks in chat bots. For specific features such as booking or displaying a subscription, when the chatbot receives a request from the user, it passes the request to the model.

The task-oriented module distinguishes the meaning of the request and returns a flag telling the chat bot what it should do for the user. The general dialogue chatbot response general conversation from users. If the patient has a request for distance or radiologists, they can limit the database output. More importantly, after the chat robot provides the necessary preparation information, the patient selects a specific radiological examination. At the same time, chat bots ask basic questions to help patients better prepare for radiology and additional question from some radiologist.

The chatbot module can provide a high accuracy and most relevant answer to users. NLP technique will be applied into this chatbot. To train this chatbot, several phases such as data cleansing, keyword extraction, synonym matching, and post processing might be implemented in this system. This module will be also connected to the middleware module.

## 3.5 API Design

In this project, API is divided into six part. Each section is distinguished by a different keyword.

patient	patient api	>
GET	/patient/answer/{id}	🔒
GET	/patient/appointment/cancel/{id}	🔒
GET	/patient/appointment/{id}	🔒
POST	/patient/info/update	🔓
GET	/patient/info/{id}	🔒
POST	/patient/login	🔓
GET	/patient/logout/{id}	🔒
POST	/patient/question/info	🔓
GET	/patient/question/load	🔒
POST	/patient/regist	🔓
doctor	doctor api	>
home	home api	>
news	news api	>
admin	admin api	>
chatbot	chatbot api	>

Image 3.5- 1 API design

Interface uses get or pos methods. For interfaces that involve transferring private data, use post for requests. Get requests are used for transferring plain or meaningless data. All API interfaces follow the restful design style.

- Client-server: by separating the user UI from the data store, we can simplify the server components to improve portability and scalability of user interfaces across multiple platforms. It can be expressed as a frontend separation of ideas.
- Stateless: every request from the client to the server must contain all the information needed to understand the request and cannot take advantage of any stored context on the server. This means that sessions are avoided as much as possible and allow the client to identify the session state itself. (token)
- Specification interface: REST interface constraint definition: resource identification; Request action; Response information; It means to identify the resource you want to operate on by URI, identify the operation you want to perform by request action (HTTP method), and indicate the execution result of this request by return status code.

For example, in order to cancel the appointment, the API interface needs to send the appointment ID to the backend system through HTTP request. Since the reservation information ID does not involve privacy information, it uses get request, and the request format is /patient/appointment/cancel/{id}

API module is the key component communicating the frontend and backend. In this project, the backend system will be implemented using Python flask framework. In building applications, backend system can only expose an URL to frontend. It is useful for decoupling different modules.

## 3.6 System functionalities

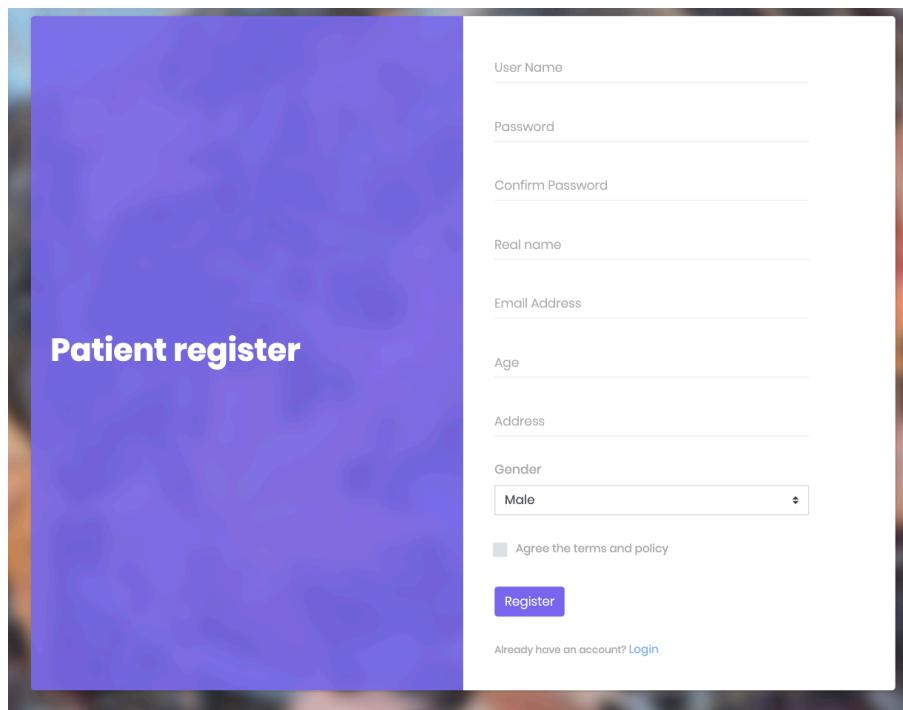
Please see 3.4.

## 3.7 User and Developer documentation

### 3.7.1 User documentation

- **User registration**

Prerequisite	None
Required data	User information



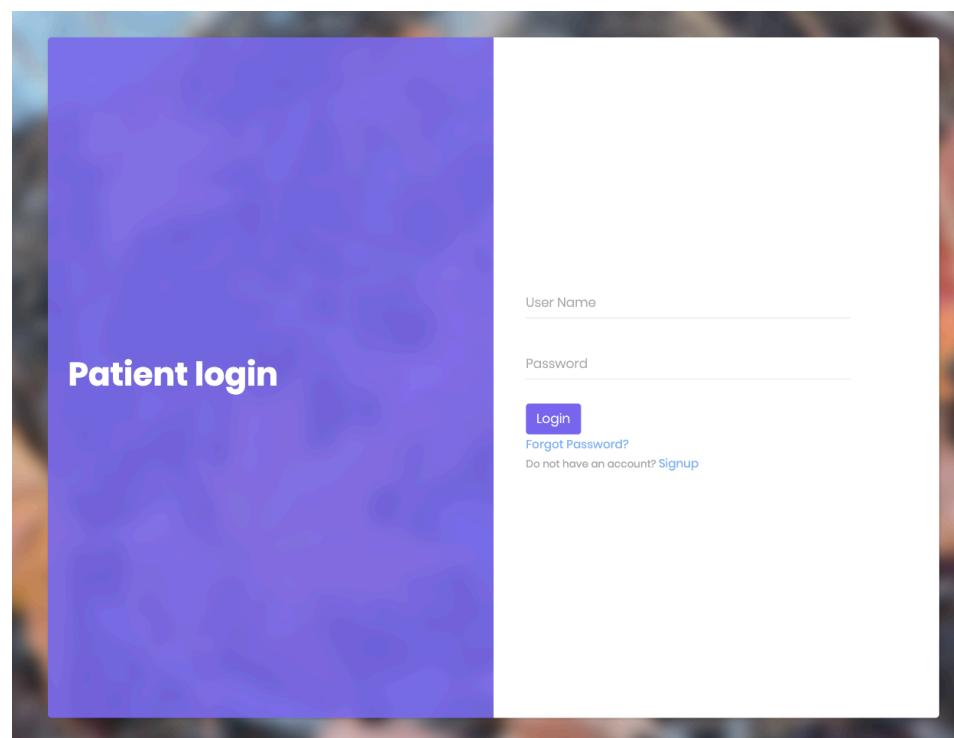
The image shows a registration form titled "Patient register". The form includes fields for User Name, Password, Confirm Password, Real name, Email Address, Age, Address, Gender (with a dropdown menu showing "Male"), and a checkbox for Agree the terms and policy. A "Register" button is at the bottom, and a link to "Login" is also present.

User Name
Password
Confirm Password
Real name
Email Address
Age
Address
Gender
Male
<input type="checkbox"/> Agree the terms and policy
<b>Register</b>
Already have an account? <a href="#">Login</a>

The user must complete the account registration according to the system prompt input specification information.

- **User login**

Prerequisite	User already has an account
Required data	Valid username and password



The image shows a login form titled "Patient login". It features fields for User Name and Password, a "Login" button, a "Forgot Password?" link, and a "Do not have an account? Signup" link.

User Name
Password
<b>Login</b>
<a href="#">Forgot Password?</a>
<a href="#">Do not have an account? Signup</a>

Enter your account and password and click the login button.

- **User appointment checking and cancelation**

Prerequisite	User already has an account and login successfully
Required data	None

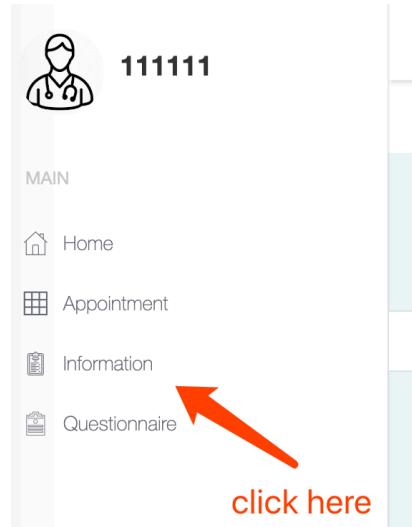
The screenshot shows a user interface with a sidebar on the left labeled 'MAIN' containing 'Home', 'Appointment' (which is highlighted with a red box and has a red arrow pointing to it with the text 'click this button'), 'Information', and 'Questionnaire'. To the right is a large panel titled 'Dashboard' with a light blue background. Below the sidebar, the URL 'Home / Appointment' is visible. The main content area displays a table of appointment details:

Doctor	Specialty	Hospital	Email	Address	Start time	End time	Date	Operations
Mr Naveed Alam	nmr	Healthcare Imaging Maroubra	Alam@gmail.com	Level 1, 832 Anzac Parade	16:00	17:00	2019-11-19 15:26:41	<span style="border: 1px solid green; padding: 2px;">View</span> <span style="border: 1px solid red; padding: 2px;">Cancel</span>
Mr Naveed Alam	nmr	Healthcare Imaging Maroubra	Alam@gmail.com	Level 1, 832 Anzac Parade	9:00	10:00	2019-11-13 14:07:08	<span style="border: 1px solid green; padding: 2px;">View</span> <span style="border: 1px solid red; padding: 2px;">Cancel</span>

On this page, the user can view information about the clinic. Click the View button to check the specific location of the clinic on the Google map. Users will also be able to delete an appointment. In addition, user can search appointment information by entering key word in the searching area.

- **User update personal information**

Prerequisite	User already has an account and login successfully
Required data	User personal information



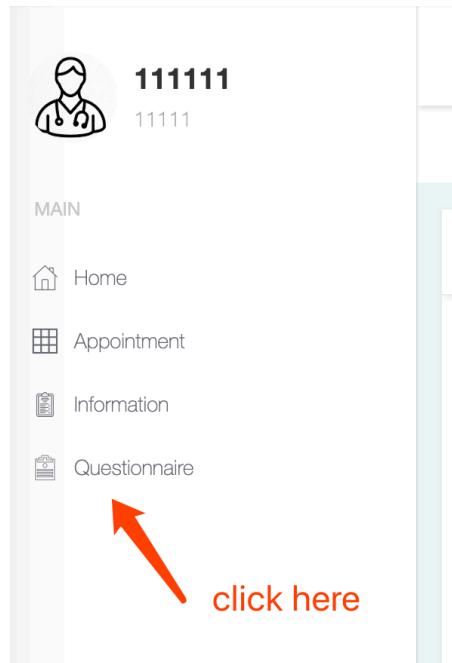
**Account information**

username	111111
password	*****
realname	11111
age	1
email	z5147976@ad.unsw.edu.au
address	1
<input type="button" value="submit"/>	

The user enters new personal information as required. The system validates user input. After filling information, click the submit button.

- **User update questionnaire**

Prerequisite	User already has an account and login successfully
Required data	Questionnaire answers



#### Account information

##### RADIOLOGY QUESTIONNAIRE

###### Patient Information

The imaging procedure you are about to have may require the use of iodine – containing contrast material. Depending on the type of procedure you are having, the contrast material may be injected into a vein, into a joint or other parts of the body, or you may be asked to drink the contrast material.

Contrast material is used by radiologists to help highlight the important structures we are examining and to improve the accuracy of the test you are about to have.

Certain medical conditions may influence the way you react to the contrast material and it is important that we know about these before proceeding. Therefore, please answer the questions below.

Are you pregnant or breast feeding? \*

yes  no

Are you allergic to any medications? \*

yes  no

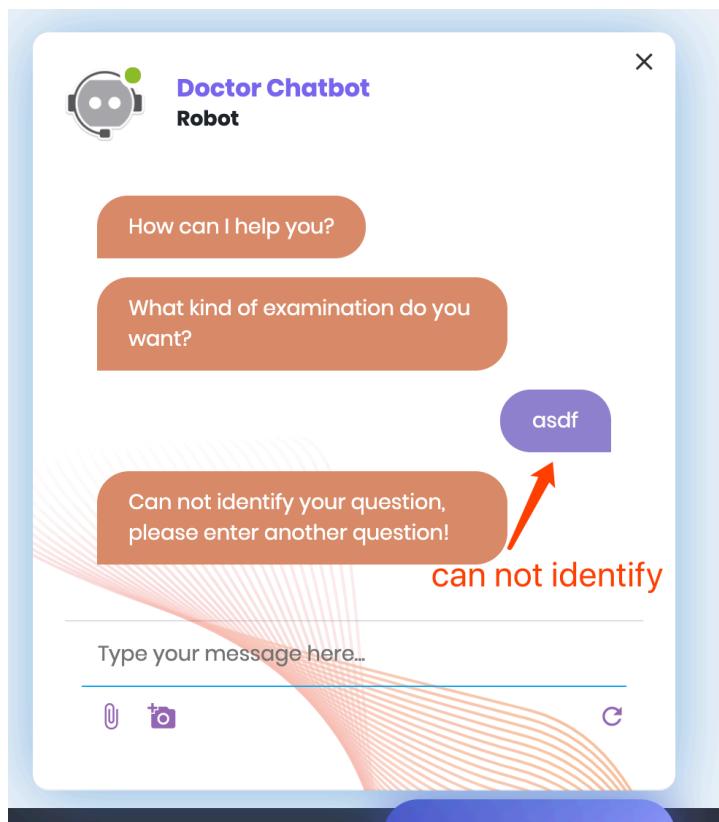
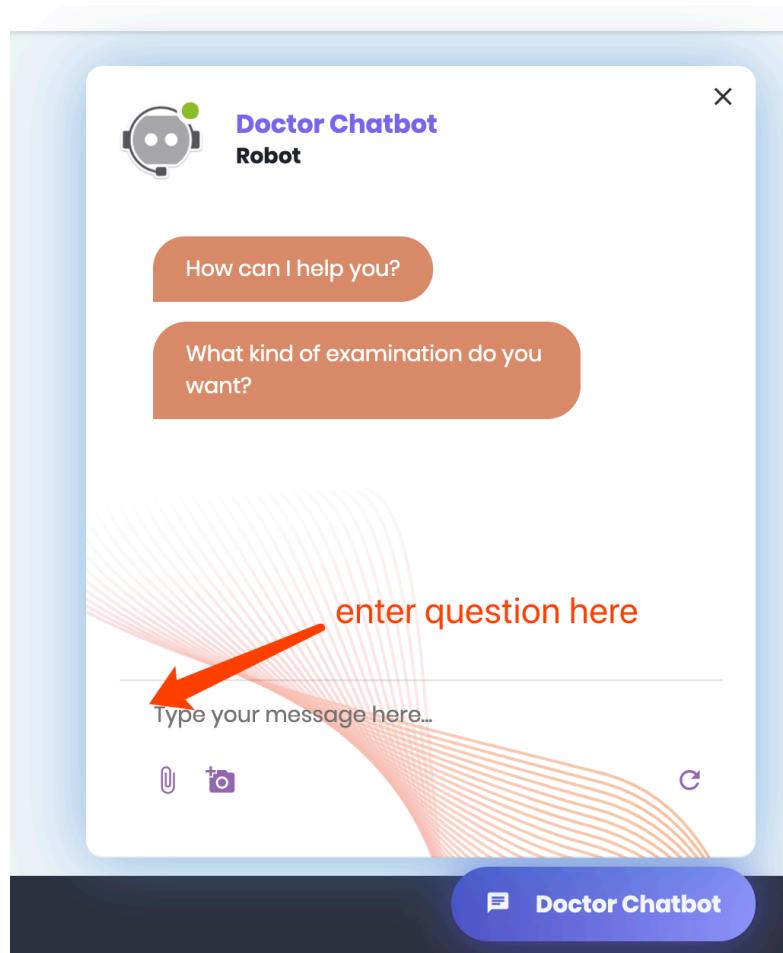
Are you asthmatic and are on daily inhalers? \*

yes  no

Update the questionnaire answers according to actual situation, then click the submit button to update the information.

- **User appointment booking**

Prerequisite	User already has an account and login successfully
Required data	None

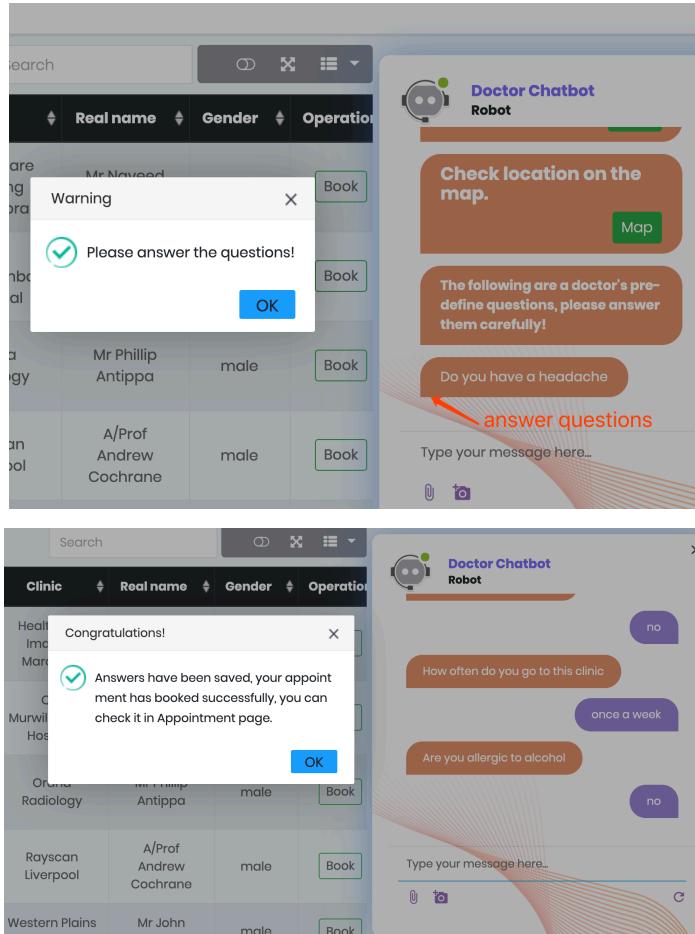


The screenshot shows a 'Doctor Chatbot' interface. On the left, there is a table titled 'appointment list' with columns: Address, Clinic, Real name, Gender, and Operation. The table contains five rows of data. An orange arrow points to the 'Address' column header. On the right, there is a 'preparation info' section with a 'Doctor Chatbot Robot' icon. It includes a 'Preparation Links' section with a link to 'Resource 1' and a button to 'Send' information via email. Another section says 'Check location on the map.' with a 'Map' button. Below these is a message input field with placeholder text 'Type your message here...' and a 'Doctor Chatbot' button.

User can enter whatever they want to ask to the robot, then the robot will response the user immediately. If the robot cannot identify the question, it will ask the user to enter another question. If user enter the message that is not related to any of the appointment, the robot might reply the user when the training dataset contains related answer. If the user asks the robot about the appointment, it will reply the doctor appointment list and the location about all the clinics and the preparation information which also can be sent be email.

The screenshot shows a 'Time table' window with a 'Dashboard' sidebar on the left. The sidebar lists clinic addresses: Level 1, 832 Anzac Parade, Ewing Street, 168 Brisbane Street, 41-43 Goulburn Street, and 62 Windsor. The main window displays a grid of time slots from 9:00 to 17:00, each with a 'Select' button. To the right of the grid, there is a vertical sidebar with sections for 'Preparation Links', 'Resource 1', 'Send', 'Check location on the map.', and a message input field.

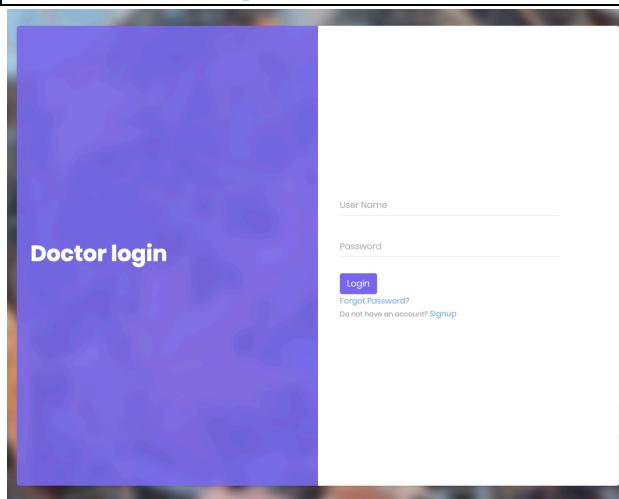
After selecting a preferred doctor, timeslot also can be selected by clicking the Book button.



Once select a preferred timeslot, the robot might ask the user some questions which are customized by the selected doctor. When the user answers all the questions, the booking process is finished

### ● Doctor login

Prerequisite	<ul style="list-style-type: none"> <li>● Doctor already has an account.</li> <li>● By default, doctor username depends on doctor's real name and the initial password is set as 123456.</li> </ul>
Required data	Valid username and password



The doctor enters the correct username and password and clicks the login button.

- **Doctor appointment checking**

Prerequisite	Doctor already has an account and login successfully
Required data	None

The screenshot shows a mobile application interface. At the top, there is a user profile section with a circular profile picture of a man, the name "Alam.Naveed", and the initials "nmr". Below this is a navigation menu with the following items:

- Home
- Appointment
- Question
- Information

A red arrow points to the "Question" option, with the text "click here" written next to it. The main content area displays a table with the following data:

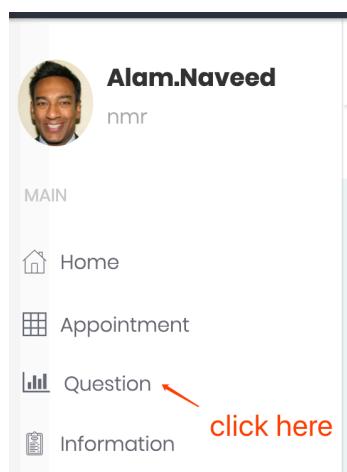
Name	Gender	Age	Email	Address	Start time	End time	Date	Operation
11111	male	1	z5147976@ad.unsw.edu.au	1	16:00	17:00	2019-11-19 15:26:41	<a href="#">Form</a> <a href="#">Question</a> <a href="#">Cancel</a> <a href="#">Finish</a>

At the bottom left of the table, it says "Showing 1 to 1 of 1 rows".

On this page, the doctor can view the answers of the user's questionnaire, see the answers of the user to the questions set by the doctor, delete the appointment and complete the appointment by clicking related button.

- **Doctor question setup**

Prerequisite	Doctor already has an account and login successfully
Required data	Related questions



**Basic Question**

Please fill in the questions.

Question\_1  
Do you have a headache

Question\_2  
How often do you go to this clinic

Question\_3  
Are you allergic to alcohol

Add more questions(max 5 questions) Add more questions(max 5 questions)

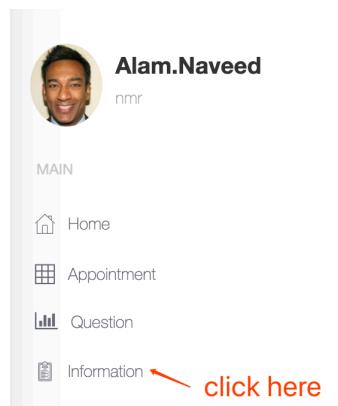
Add more Add more Cancel Cancel questions

submit

Doctors can customize basic questions as follow-up consultations. Each doctor can customize up to five questions. The doctor can also delete the previous questions. After the operation, click the submit button to update the content.

### ● Doctor information update

Prerequisite	Doctor already has an account and login successfully
Required data	Personal information



**Account information**

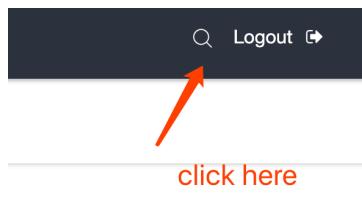
username	Alam.Naveed
password	*****
realname	Mr Naveed Alam
age	53
email	Alam@gmail.com

**submit**

The doctor enters new personal information as required. The system validates user input. After filling information, click the submit button.

- **Doctor/User system logout**

Prerequisite	Doctor/User already has an account and login successfully
Required data	None



The doctor or user exits the system by clicking the logout button.

### 3.7.2 Developer documentation

The project API was developed using a flask framework that uses the swagger doc to fully explain the API interface. In the background code, the necessary comments are also added. This allows developers to read the code and understand the required parameters during development.

The screenshot shows the Flask Swagger documentation for a POST request to '/patient/regist'. The 'Parameters' section includes a 'payload \* required' parameter of type '(body)', which is described as 'patient\_register api'. The 'Responses' section indicates a response content type of 'application/json'.

Image 3.7.2- 1 Flask swagger documentation

```
...  
doctor login api  
...  
@doctor.route('/login')  
class login(Resource):  
    @doctor.response(200, 'OK')  
    @doctor.response(404, 'Invalid Username or Password')  
    @doctor.doc(description="login api")  
    @doctor.expect(doctor_login_model)  
    @get_logger.get_logger_aop  
    @.all_exception  
    def post(self):  
  
        content = request.get_json()  
        username = content['username']  
        password = content['password']  
  
        print(username, password)  
  
        code, doctor = doctor_service.doctor_services().get_doctor_by_username(username, password)  
        print(doctor, code)  
        if code == 200 and patient is not None:  
            session['username'] = doctor['username']  
            session['id'] = doctor['id']  
            res = make_response(jsonify(data=doctor, code=200), 200)  
        else:  
            url = 'index.html'  
            res = make_response(jsonify(redirecit_url=url, code=404), 200)  
    return res
```

Image 3.7.2- 2 Backend code

## 3.8 Third party application

### 3.8.1 Remote database and database visualization tools

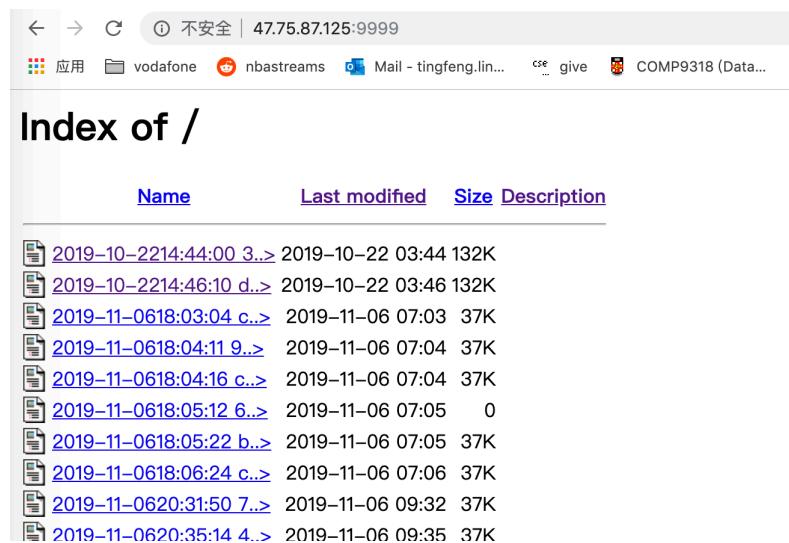
The screenshot shows the MySQL Workbench interface. On the left, the 'project' tree view lists databases like 'comp9322', 'comp9900', and 'test1'. The central area displays a table named 'timeslot' with columns: 名 (Name), 行 (Rows), 数据长度 (Data Length), 引擎 (Engine), 创建日期 (Create Date), 修改日期 (Modify Date), 排序规则 (Sort Rule), and 注释 (Comment). The table contains 12 rows of data.

名	行	数据长度	引擎	创建日期	修改日期	排序规则	注释
clinic	42	16.00 KB	InnoDB	2019-11-10 12:19:39	2019-11-10 12:28:36	utf8mb4_0900_ai_ci	
doctor	39	2.41 MB	InnoDB	2019-09-26 15:35:22	2019-11-13 05:20:35	utf8mb4_0900_ai_ci	
field	6	16.00 KB	InnoDB	2019-10-15 04:44:33	2019-10-23 06:09:40	utf8mb4_0900_ai_ci	
main_news	27	16.00 KB	InnoDB	2019-10-08 07:10:19		utf8mb4_0900_ai_ci	
news	42	512.00 KB	InnoDB	2019-10-01 13:06:07		utf8mb4_0900_ai_ci	
patient	20	16.00 KB	InnoDB	2019-11-07 10:59:24	2019-11-13 03:04:05	utf8mb4_0900_ai_ci	
preparation	6	16.00 KB	InnoDB	2019-10-15 04:06:05		utf8mb4_0900_ai_ci	
question	2	16.00 KB	InnoDB	2019-10-28 01:53:53	2019-11-11 02:48:26	utf8mb4_0900_ai_ci	
reserve	38	16.00 KB	InnoDB	2019-11-09 06:07:48	2019-11-13 04:49:00	utf8mb4_0900_ai_ci	
timeslot	8	16.00 KB	InnoDB	2019-09-24 06:10:42		utf8mb4_0900_ai_ci	

*Image 3.8.1- 1 Remote database*

For more efficient collaborative development, the project USES a remote database. The visual operation software (Navicat) is used to realize the connection with the database and simplify the operation of the database. The use of remote database enables team members to develop at anytime and anywhere, while ensuring the security and reliability of the data.

### 3.8.2 File server system and file convert API



*Image 3.8.2- 1 Apache file server*

```
import pdfcrowd
import sys
import yaml

f = open('application.yml','r',encoding='utf-8')
cont = f.read()
x = yaml.load(cont)['html2pdf']

username=x['user']
key=x['key']

def html2pdf(url,pdf_name):
    try:
        # create the API client instance
        client = pdfcrowd.HtmlToPdfClient(username, key)

        # run the conversion and write the result to a file
        client.convertUrlToFile(url,pdf_name)
    except pdfcrowd.Error as why:
        # report the error
        sys.stderr.write('Pdfcrowd Error: {}\n'.format(why))

    # handle the exception here or rethrow and handle it at a higher level
    raise
```

*Image 3.8.2- 2 HTML converter*

In order to better present the questionnaire filled by the user to the doctor, when the user completes the questionnaire, the third-party API interface is invoked to convert the HTML page into a PDF file, and then the converted file is uploaded to the file server. The doctor can download or check the file online when viewing the questionnaire. The

apache file server is applied to this project. When files are uploaded to the server, each file has a unique filename and the file path is stored in the database.

### 3.8.3 Google map API

```
import googlemaps
import yaml
f = open('application.yml', 'r', encoding='utf-8')
cont = f.read()
x = yaml.load(cont)[‘googlemap’]

# Geocoding an address
def address2geocode(address):
    gmaps = googlemaps.Client(x[‘key’])
    geocode_result = gmaps.geocode(address)
    return geocode_result[0][‘geometry’][‘location’]

# Look up an address with reverse geocoding
def geocode2address(lat,lng):
    gmaps = googlemaps.Client(x[‘key’])
    reverse_geocode_result = gmaps.reverse_geocode((lat, lng))
    print(reverse_geocode_result)
    print(reverse_geocode_result[0][‘address_components’][1][‘long_name’])
```

Image 3.8.3- 1 Google map API

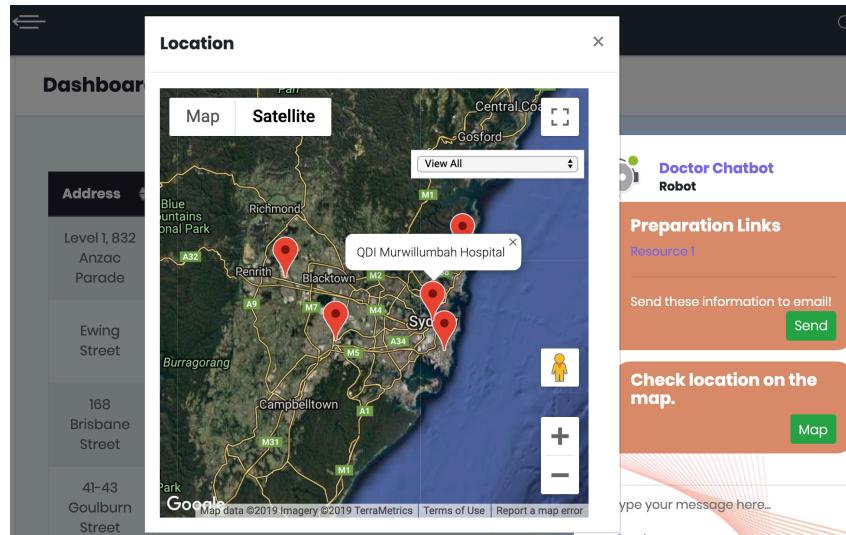


Image 3.8.3- 2 Google map implementation

In order to make it more intuitive for the user to know the location of the clinic, the Google map API is used in the function of the user to view the reserved clinic and booking appointment.

### 3.8.4 Email API

```
| def add_content(self, content: str, _type: str = 'txt'):
|     if _type == 'txt':
|         print('txt')
|         print(content)
|     text = MIMEText(content, 'plain', 'utf-8')
|     if _type == 'html':
|         text = MIMEText(content, 'html', 'utf-8')
|
|     self._email_obj.attach(text)
|
| def add_file(self, file_path: str):
|     email_file = MIMEText(open(file_path, 'rb').read(), 'base64', 'utf-8')
|     email_file['Content-Type'] = 'application/octet-stream'
|     file_name = os.path.basename(file_path)
|     email_file["Content-Disposition"] = f'attachment; filename="{file_name}"'
|     self._email_obj.attach(email_file)
|
| def send_email(self):
|     print(self._email_obj)
|     self._server.sendmail(from_addr=self.SENDER, to_addrs=self.RECEIVER, msg=self._email_obj.as_string())
|
| def quit(self):
|     self._server.quit()
|
| def close(self):
|     self._server.close()
```

Image 3.8.4- 1 Email API

The third-party mail API is used in the functions such as user registration completion and sending preparation materials. Relevant information will be sent to the specified user mailbox.

## 4 Chatbot

There are two chatbots module in this project: general dialogue chatbot and task-oriented chatbot.

General dialogue chatbot applies seq2seq with attention to response general sentences from user. Task-oriented chatbot applies NLP technology to analysis rough meaning of input sentences through synonyms or related words of radiology examination. Project frontend transfers input sentences to Task-Oriented chatbot firstly.

### General dialogue Chatbot:

The training dataset of this Chatbot comes from Kaggle named Cornell movie-dialogs corpus which contains around 25MB movie conversation.

- Preprocess:  
At first, from *movie\_lines.txt* and *movie\_conversation.txt*, we extract all pairs of sentences and transform their format from Unicode to Ascii and write them into a new file in local directory. We can use these new formatted data directly in training process which is more efficient.
- Save Variables in advance:  
Voc object and pairs list, both containing necessary training sentences and related

data, are stored in local directory. But before that, we have to do some extra preprocessing work on training data:

- 1) Replacing all non-alphabetic characters except basic punctuation to Space;
- 2) Removing the pairs if two sentences are longer than  $MAX\_LENGTH$ , because we need briefly and concise sentences;
- 3) Removing those pairs containing the words whose frequencies are less than  $MIN\_COUNT$  as they are meaningless, or too particular.

Then we store voc object and pairs list by Pickle library.

- Network:

There are 3 classes (Encoder, Attn, Decoder) to build seq2seq network model with attention.

- 1) Encoder:

We use bidirectional GRU<sup>10</sup>, a kind of RNN unit, to build Encoder part. In order to avoid padding characters affecting training performance negatively, we use `nn.utils.rnn.pack_padded_sequence` and `nn.utils.rnn.pad_packed_sequence` to pack and pad batch sequence in forward function.

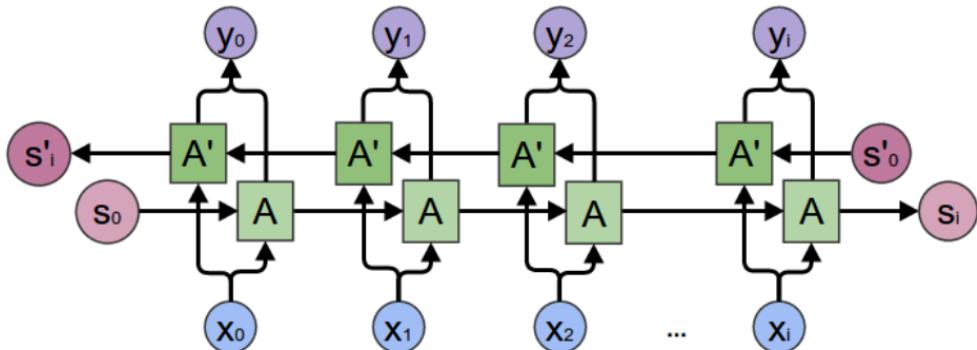


Image 4- 1 Bidirectional Recursive Neural Networks

Image source: [https://jeddy92.github.io/JEddy92.github.io/ts\\_seq2seq\\_intro/](https://jeddy92.github.io/JEddy92.github.io/ts_seq2seq_intro/)

Input:

- a batch of sequence;
- lengths of all sequence in input batch;
- previous hidden state.

Output:

- The output features from the last layer of the GRU;
- Hidden state.

- 2) Attn:

The reason why we need to use attention in seq2seq network is that if we only rely

on the context vector to encode the entire input sequence, then we are likely to lose information. This greatly limits the capabilities of vanilla seq2seq decoders, especially when dealing with long input sequences.

Thus, we use *Global Attention mechanism*<sup>11</sup> to improve our network model which means all hidden states are considered when driving context vectors.

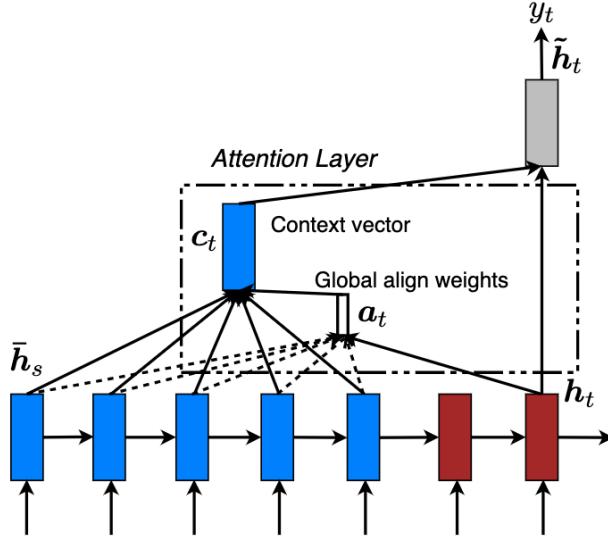


Image 4-2 Global attentional model at each time

Image source : <https://arxiv.org/abs/1508.04025>

There are 3 different alternatives to calculate attention score:

$$\text{score}(\mathbf{h}_t, \bar{\mathbf{h}}_s) = \begin{cases} \mathbf{h}_t^\top \bar{\mathbf{h}}_s & \text{dot} \\ \mathbf{h}_t^\top \mathbf{W}_a \bar{\mathbf{h}}_s & \text{general} \\ \mathbf{v}_a^\top \tanh(\mathbf{W}_a[\mathbf{h}_t; \bar{\mathbf{h}}_s]) & \text{concat} \end{cases}$$

In our model, we choose the first dot product which is the most common function. In this method,  $h_t$  is current target state and  $h_s$  are all source state.

### 3) Decoder:

Compared with Encoder, GRU is unidirectional in decoder. After forward propagation, we calculate attention score by Attn class and multiply it with output of Encoder to get weighted context vector. We concatenate target hidden state and source-side context vector by:

$$\tilde{\mathbf{h}}_t = \tanh(\mathbf{W}_c[\mathbf{c}_t; \mathbf{h}_t])$$

And then next prediction word was generated by softmax:

$$p(y_t|y_{<t}, x) = \text{softmax}(\mathbf{W}_s \tilde{\mathbf{h}}_t)$$

Input:

- a batch of sequence;
- last hidden state of Encoder;
- Encoder outputs.

Output:

- A softmaxed tensor containing the probabilities that each word is the next output;
- word in the decoding sequence;
- Last hidden state.

#### 4) Loss function and optimizer:

Since the training data are padded, we would better not consider those padding characters in the training process. Thus, Padded sequences are used to generate binary mask, and then apply target tensor mask and output tensor mask into *NLLLoss()* to calculate loss. *Adam* optimizer is good choice for normal text classification task.

- Training:

*Voc* object and pairs list are fed from local directory to network model. In training process, there are two concepts to improve performance:

- 1) we apply *teacher\_frocing* randomly to improve training performance and decrease probability of instability. When random number is smaller than *teacher\_frocing\_ratio*, the next input is the current real target output, otherwise, next input is the decoder's guess.
- 2) *Gradient clipping* helps to avoid gradient explosion problems.

For above two, *teacher\_frocing\_ratio* and *clipping threshold* are manually set.

- Evaluate:

There is a concept that greedy search to find the most likely response sentences. After generating input batch sequence, we push input words sequentially into *Greedy Search Decoder* class object and output tokenization sequence is transformed to words sequence which is the prediction.

#### Task-Oriented chatbot:

Task-Oriented chatbot is used *NLTK* and *Spacy* library to do NLP work and analysis rough meaning of input sentences, especially for booking appointment. There is a large number of synonyms of key words about radiology examination in database. If there is no positive response from Task-Oriented chatbot, input sentences are transferred into general dialogue chatbot and there is a response general conversation, Otherwise, Task-Oriented chatbot returns the flag of related radiology examination to backend.

# 5 Implementation

## 5.1 Performance

This system uses the Alibaba server that set up in China. The database is installed on the server. In addition, between module and module, it uses HTTP request. Response time is the key point that might affect the user experience of the system.

## 5.2 Challenges

### 5.2.1 Frontend

- User experience is an important part of design. For example, when a user clicks a button, the website needs to let the user know that the system is responding to the user's action. In addition, chatbot design and interface presentation is also a difficult part of front-end design. How to make the user clear about the reservation process.
- There was a lot of difficulty in calling the Google map API. The official example was found on the official website, but during the implementation, the map information could not be displayed. It was finally implemented with the third generation of Maplace JavaScript.
- In addition, the design of the questionnaire is also a difficulty, how to embed the questionnaire template of the third party into the existing page.
- The request for the frontend is entirely http-based. How to store user information and variables becomes a difficult problem for the front end to render the page. The system uses session and local storage to store information.

### 5.2.2 Backend

- How to solve the cross-domain problem is also an important difficulty, because the project uses the architecture of microservices. All module use HTTP requests. The system can be divided into three parts, front end, back end and chatbot. Different modules are deployed on different ports. Hence, in order to integrate all parts into the system, cross-domain problems should be handled.

### 5.2.3 Chatbot

- A suitable general dialogue dataset is necessary as it should contain general topics and have medium size for acceptable training time. The final dataset has been found

- in Kaggle for one week.
- In order to improve model performance and avoid gradient explosion, teacher\_forcing and gradient clipping are pretty helpful.
- Data preprocessing and cleaning spend much effort and parameters adjustment as well.

## 6 Project Management

### 6.1 Project Methodology

#### 6.1.1 Agile project management

Our team use the agile project management method. According to the characteristics of agile project management, our team completed the project plan in five stages.

- Conception: determine the product conception, the scope of the project and the way the team works together.
- Speculation: develop function-based vision plans, milestones, and iteration plans to ensure delivery of the envisioned product.
- Exploration: provide tested functionality in a short period of time, continuously working to reduce project risks and uncertainties.
- Adaptation: review the submitted results, current situation and team performance, and make adjustments if necessary.
- Termination: terminate the project and exchange major learning achievements.

#### 6.1.2 Scrum software development

Our team chose to use scrum as a project management method. Scrum is an agile project management method, usually a set of specific rules to follow when practicing agile software development.

- Sprint planning meeting 1: work together with the original demander, product owner, and team to prioritize tasks and set Sprint goals and project Backlog.
- Sprint planning meeting 2: the team broke down each item in the established project Backlog into multiple tasks. Each task is limited to 10 days.
- Scrum weekly meeting: a progress coordination meeting between project team members. Meetings are held at the same time and place every week. The time is limited to one hour.
- Sprint acceptance meeting: the original demander or product owner determines whether the actual released functionality is consistent with the established Sprint goals.
- Sprint retrospective: the project team analyzes the successes and obstacles

encountered in the Sprint.

The cycle (time box) of the entire Sprint is determined to be 8 weeks, and the specific schedule is as follows:

Specific Schedule	Time
Sprint meeting	8 days
Development	40 days
Acceptance & summary	8 days

## 6.2 Risk Management

Risk management is the process of identifying, assessing and controlling threats to an organization's capital and earnings. These threats, or risks, could stem from a wide variety of sources, including financial uncertainty, legal liabilities, strategic management errors, accidents and natural disasters. Risk is the uncertainty that leads to project failure. Risk management can help us complete tasks on time and in volume. There are some risks listed below may occur in our project.

Risk Event	Likelihood	Impact	Ranking	Response
Some bugs we may not detect	3	5	1	Enough test
Underestimation of the time/resource commitment	2	4	2	Careful schedule
Group member Absence	1	3	3	Flexible meeting time
Too many compromise	3	2	4	Set some core functions

## 6.3 Milestones

A project management milestone is a measure or method of monitoring the progress of a project. In this project, the first important milestone is the realization of chat bots. The technical realization of chat bots is a sign of more than half of our projects. It is website innovation. The second important milestone is the successful connection between the chat bot and the web page, which represents the basic completion of the technical aspects of the website.

No.	Tasks	Target date	Actual date
1	Database setup, proposal	29/09/2019	27/09/2019
2	Frontend, data extraction from external sources	07/10/2019	05/10/2019
3	Chatbot, backend	18/10/2019	21/10/2019
4	Integrating all parts	03/11/2019	04/11/2019
5	Testing	15/11/2019	17/11/2019

## 7 Team Profile

### 7.1 Roles

Z_id	Name	Role	Responsibilities
Z5147976	Tingfeng Lin	Scrum master and Developer	Web develops Backend developer API develop
Z5125251	Yi Xiao	Developer	Deep learning Chatbot Algorithm
Z5156244	Yousheng Sui	Developer	Frontend developer UI designer
Z5145036	Yan Duan	Developer	Reporter UI designer

## 8 Reference

1. A general guide to patient preparation for radiological investigations  
<https://repository.up.ac.za/handle/2263/64336>
2. How to Prepare for an MRI <https://www.wikihow.com/Prepare-for-an-MRI>
3. Community Radiology Associates <https://www.radnet.com/community-radiology/>
4. Swiss radiology portal -patient preparation for radiological procedures  
<http://homepage.hispeed.ch/piben/radiologie/preparationE.html>
5. I-MED 1 Procedures - I-MED Radiology <https://i-med.com.au/procedures>
6. Architecture for building Conversational Agents that support Collaborative Learning  
<http://www.cse.unsw.edu.au/~cs9900/19T3/For-ConvrstionalAgents/KumarAndRoseInPress.pdf>
7. Dialogue and Conversational Agents [http://www.cse.unsw.edu.au/~cs9900/19T3/For-ConvrstionalAgents/Dialogue\\_and\\_Conversational\\_Agentsling\\_intro\\_2017.pdf](http://www.cse.unsw.edu.au/~cs9900/19T3/For-ConvrstionalAgents/Dialogue_and_Conversational_Agentsling_intro_2017.pdf)
8. Conversational Agents, Dan Jurafsky Stanford University  
<https://web.stanford.edu/class/cs124/lec/chatbot.pdf>
9. Dialogflow - Build natural and rich conversational experiences <https://dialogflow.com>
10. Seq2Seq with Bidirectional GRU  
[https://pytorch.org/tutorials/intermediate/seq2seq\\_translation\\_tutorial.html](https://pytorch.org/tutorials/intermediate/seq2seq_translation_tutorial.html)
11. Global attention <https://arxiv.org/abs/1508.04025>