Online Hospital Registration and Appointment Management System (OHRAMS)

Chew Jing Qiao
School of Computing
Asia Pacific University of Technology and Innovation (APU)
Kuala Lumpur, Malaysia
TP057785@mail.apu.edu.my

Lee Kim Keong
School of Computing
Asia Pacific University of Technology and Innovation (APU)
Kuala Lumpur, Malaysia
kimlee@apu.edu.my

Reshiwaran Jegatheswaran
School of Technology
Asia Pacific University of Technology and Innovation (APU)
Kuala Lumpur, Malaysia
reshiwaran@apu.edu.my

***Abstract*—The waiting time for a patient to be served in a hospital has become time consuming. This is due to a poor management system. A well-developed management system would reduce the time spent by patients in the hospitals. With a well-designed Online Hospital Registration and Appointment Management System (OHRAMS) would overcome certain issues such as patient would no longer face long waiting time. With the proposed Online Hospital Registration and Appointment Management System (OHRAMS), there would have an increase on the efficiency of the registration process.**

Keywords—Healthcare, Online System, management system, queuing system, Appointment management system

I. INTRODUCTION

Hospitals are health care organizations and medical institutions which is normally in a form of a building whereby doctors, specialized surgeons, nurses, and other hospital staffs provide suitable treatment to patients that are sick or in need of medical treatment with the help of hospital tools and equipment.

A. Background Study

Nowadays, a good online hospital management system that is to manage patients' information and other related processes is indeed demanding in the information technology field. Some aspects of the online hospital management system will determine its user satisfaction and usefulness. The aspects may consist of service quality, information quality, and system quality (Nunes & Javier, 2014). The satisfaction of the patients does not only come from the care and treatment provided by the doctors and nurses, but also the quality of the services provided by the hospital management system when comes to the mandatory processes that the patients must undergo (M, 2018).

Throughout the years, the demand for online hospital management systems regarding to the functions and performance are increasing day-by-day (Abraham & Joyce, 2016).

B. Problem Statement

Long waiting time of hospital staffs and patients is one of the biggest problems of current online hospital management system. Firstly, long patient waiting time happens during the registration process. Patients usually spend a lot of time when waiting to be registered due to the long queue and waiting to be serviced after registration. Besides that, the registration process is time-consuming as well. This happens because staffs are doing these processes manually, such as keying-in new patient information, retrieving old patient information and much more (Gadhari & Kadam, 2016). On top of that, the patients that are in queue may miss their turn when they are away, such as going to the toilet.

According to Anpan et al. (2020), patients will have to queue up one by one to register at the registration counter, and the registration process will be done by the hospital staffs. Taking that into account, the current online hospital management system lacks immediate data retrieval, trouble in storage, and human error are indirectly causing the problem of long waiting time. Although current online hospital management system that uses manual or semi-manual technique in many processes is quite uncomplicated to implement but will lead to issues that will cause long waiting time indirectly, especially in situations like big number of patients walking in (M, 2018). All margins, column widths, line spaces, and text fonts are prescribed; please do not alter them. You may note peculiarities. For example, the head margin in this template measures proportionately more than is customary. This measurement and others are deliberate, using specifications that anticipate your paper as one part of the entire proceedings, and not as an independent document. Please do not revise any of the current designations.

C. Rationale

. Firstly, the system allows patients to register and check in by themselves, which solves the issue of long waiting time, as well as the human error issue. One thing to mention, patients that are in queue can keep track of the current queue status even if they are away from the queue for a quick snack or going to the toilet. Besides that, the proposed system can increase the efficiency of the registration process which is beneficial in terms of time and cost towards the patients and the hospital staffs. Taken that into account, less complaints regards to registration related issue will be reported. Furthermore, the appointment management page in the proposed system allows patients to make appointments based on the available slots of the specific doctors and making changes on existing appointments. This will minimize the burdens that the hospital staffs and doctors had to go through.

On top of that, the proposed Online Hospital Registration and Appointment Management System will be built in web application form, whereby smartphones and personal computers can easily access to the proposed system. Taken that into account, the target users can access to the web application at ease as smartphones had already been a necessity item to majority of the people nowadays.

D. Potential Benefits

Through the preparation of the proposed system, two types of potential benefits are obtained, which are tangible benefits and intangible benefits. Below are the lists of tangible benefits and intangible benefits.

1) Tangible Benefits

The tangible benefits that are obtained throughout the preparation of the proposed system are as follows:

- Minimized waiting time of registration and check in process.
- Reduction in the volume of work of the hospital staffs.
- Preventing the risk of human errors when comes to registration and booking appointment by the hospital staffs.
- Increasing the efficiency of the registration, check in, and appointment booking process, hence increase in productivity.

2) Intangible Benefits

Below is the list of intangible benefits that are gathered throughout the preparation of the proposed system:

- Patients or visitors of the hospital will have a better experience in which increasing the reputation of the hospital, as well as the increment in customer satisfaction.
- Reduce confusion and burden when comes to problems related to booking appointment, such as changing appointment time and the available slots of the specific doctor.
- Easier data retrieval for patients and doctors to check specific details, such as current queueing number, patient records and much more.

E. Aim

To determine a suitable queuing theory and algorithm in General Hospital in Malaysia. Also, designing and developing an effective medical appointment web system to improve irregular or regular patients experience in medical consultations.

F. Objectives

i. To study the medical consultation appointment process in General Hospital, Malaysia.

ii. To investigate different queueing theories and algorithms to minimize patient waiting time.

iii. To design and develop a queuing solution for the medical appointment registration process in Hospital.

iv. To test the medical registration, appointment booking as well as effectiveness of the developed queuing system.

v. To evaluate different scenarios in registering medical appointments of the system.

G. Deliverables

Firstly, patients will register themselves before checking in to the hospital. After that, the hospital staffs will receive the request and approve the check in request to start the queueing, which the patient will then receive the queueing number and some details. Furthermore, the patients can book appointments according to the availability of the specific doctor. Next, the hospital staffs can perform a list of functions, such as managing patient profiles, managing appointments, queue status, manage medical reports, and much more. On top of that, doctors can accept or reject appointment request, view patient profiles, and managing the medical report.

H. Nature of Challenges

To start off, the first challenge is to understand and utilize queue algorithm which is to be implemented in the registration process. This algorithm comes in once a patient is registered. After that, the patient will receive a queue number. The sequence of the queue number depends on the seriousness, type of consultation, and other factors. Besides that, the security of the system must be taken into consideration as well. This is because the users of the system can easily retrieve specific data and information, hence there are risk of data leakage (Mun, 2022). To solve this issue, the developer must carry out the encryption of data to prevent the loss of the confidential data (K, 2018). Moreover, time management is also one of the hardest challenges for the developers. This is due to the reason that they must ensure the preparation and development of the proposed system will be completed on time.

II. System Architecture

A. Patient Features

This Online Hospital Registration and Appointment Management System (OHRAMS) allows patient user type to sign up for a new account as patient and to login into the system. A list of credentials will be needed from the patient to register whereas only the IC Number and password is needed for the patient to login into the system. Once the patient has logged into the system, the system will redirect to the patient dashboard page.

Furthermore, patients can view the current queue status including the current queue number and the following queue number. Besides that, patients can check in now or check in for later to join the queue for a general consultation. Taken that into account, checking in for a queue is the core feature of OHRAMS, whereby when a patient requests for a queue, this system will arrange the queue according to the request date time in the correct order. On top of that, patients can navigate to the queue management page to view all queues that are requested. Cancelling of queue requests can also be done by patients.

Next, patients can view list of all appointments and list of requested appointments in the appointment management page. In addition, patients can cancel appointments that have status of “Requested”. In this page, patients can book a new appointment with the required credentials as well, which is as well the main feature of OHRAMS. Once an appointment is requested, the corresponding doctor will receive the request. The credentials include appointment date time, reason of booking the appointment, and the favoured doctor.

B. Hospital Staff Features

This Online Hospital Registration and Appointment Management System (OHRAMS) also allows hospital staff user type to login into the system by entering the correct Staff ID and password. After logging in, the system will redirect to the hospital staff dashboard page. One thing to mention, hospital staffs has the most access to all functionalities.

Furthermore, hospital staffs can manage patients, doctors, and hospital staffs, including register new patient, register new doctor, register new hospital staff, delete patient account, search user profile, view all user profile, and update all user profile.

Moreover, hospital staffs can view current queue status, including the current queue number and the following queue number. In addition, hospital staffs can manage queue, which includes ending current queue and cancel queue. One thing to mention, ending current queue to move the proceeding queue as the ongoing queue is the main feature of this OHRAMS. Next, hospital staffs can also view appointment list, search for appointments, and update appointment details.

Hospital staffs can view and update own user profile in the user profile page. Lastly, hospital staffs can also logout of the system.

C. *Doctor Features*

The first features for doctor user type view list of patients, doctors, and hospital staffs. Besides that, doctors can view and search for the user profile of all users. Searching users will require entering the ID of the user. Furthermore, doctors can view list of appointments as well as to filter the appointment list to show only the requested appointments. Most importantly, doctor user type possesses the core feature of this OHRAMS, which is to manage appointments. Doctors will receive appointment requests, whereby doctors can accept or reject appointment requests. Moreover, doctors can also create new medical reports for patients and to view the medical reports of patients.

Doctors can view and update own user profile in the user profile page as well like the other user types. Lastly, doctors can also logout of the system by clicking the logout button.

D. *Abstract Architecture*

Fig 1. shows the system architecture design of Online Hospital Registration and Appointment Management System (OHRAMS). Firstly, users can access to the web application using almost all types of devices that have access to web browsers, including smartphones, personal computers, and laptops. Users will interact with the user interfaces through the devices, which will send the interactions to the front end. The example of the user interface interactions include clicking button, filling in input fields, and much more.

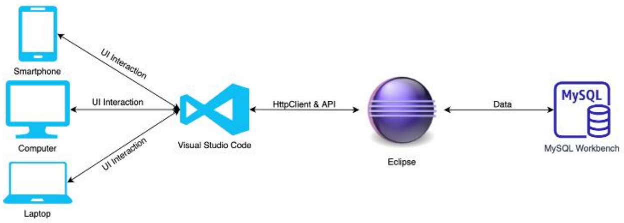


Fig. 1. System architecture design

After that, the front end will receive the data created by the user interface interactions. In OHRAMS, the front end development software used in Visual Studio Code. In this layer, Visual Studio Code is responsible to take in the data received and to perform the corresponding functions, which will then be sent back to the user interface for the users. In addition, this layer is only capable to perform limited functions depending on the user interface interactions, hence, communication to the backend will be mandatory.

The backend development software used for OHRAMS is Eclipse. The front end will need to communicate to the backend to call APIs with or without data. Through calling APIs in the backend, the API will fire the function needed to get required data, which will then be sent back to the front end, and the front end will then send the new data to the user interface. For example, users click the button to display user profile, visual studio code will accept the User ID, which then Eclipse will use the User ID to get the user profile details and send back to front end and back to the user interface. One thing to mention, the communication used between front end and back end is HTTP Client. The purpose of HTTP Client is to send requests and retrieve responses.

Finally, the system architecture design of OHRAMS also includes a database management system, which in this case is the MySQL Workbench. The backend will communicate with MySQL Workbench to retrieve required data. Besides that, the backend will as well send data to MySQL Workbench for the purpose of storing new data and updating existing data.

In general, the system architecture design of OHRAMS is where users interacts the user interface through devices, sending data to the frontend, which then the frontend communicates with the backend to call the correct API, and the backend then retrieves data from or sends data to the database.

III. TEST PLAN

Taken that into account, the testing that are chosen for this system are unit testing and user acceptance testing.

Firstly, unit testing is conducted by testing every single unit with test conditions and example data. In addition, there are a total of 31 sets of test plans, having 216 test cases in total. The purpose of this unit testing is to ensure that the actual result of each single test case are same as the expected result. By performing unit testing, developers can easily discover bugs and errors of the system, which includes input validation, typing errors, false results, and much more. Hence, developers can immediately take action to fix the bugs and errors before conducting user acceptance testing.

Next, user acceptance testing is conducted as well after conducted unit testing. In addition, the 5 participants of this user acceptance testing consists of 3 patients, 1 doctor, and 1 hospital staff. Each participant will test on their corresponding user type by logging in to the respective user type. Every participants will provide rating on different aspects provided in the table given. After that, participants will also provide feedbacks and suggestions of the system, therefore the developers can take the feedbacks into considerations to make appropriate modifications to the system, which in turns satisfying the needs of the users.

IV. IMPLEMENTATION

A. *Screenshot of Home page*

The user type that can access this page includes patient, doctor, and hospital staff. This page is the home page of Online Hospital Registration and Appointment Management System (OHRAMS). This page does not have any features except for the navigation bar that is to be redirected to login and registration page. Fig 2. Shows the screenshot of homepage.

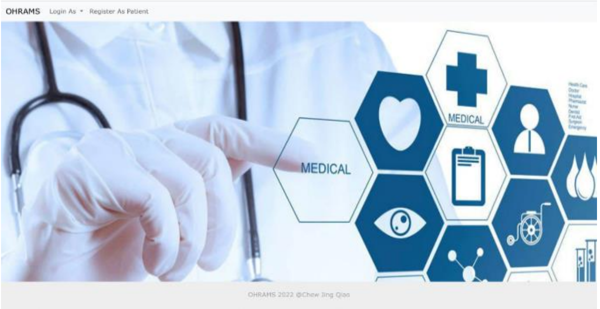


Fig. 2. Screenshot of homepage

B. Screenshot for Patient Registration Page

Everyone can access this patient registration page that is used to register a new patient account. Users must fill in all the input fields in the form and click "Register" button to successfully register an account. One thing to mention is that the IC Number entered must not be an existing IC Number. Fig 3. Shows the screenshot for patient registration page.

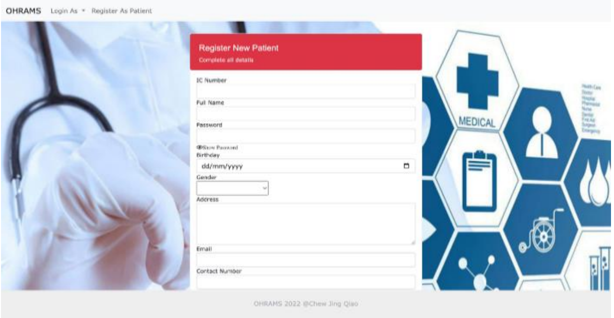


Fig. 3. Screenshot for patient registration page

C. Screenshot for Hospital Staff Login Page

This Hospital Staff Login page serves the purpose for hospital staff users to login into the system. Hospital staffs must enter the correct Hospital Staff ID and password and click the "Login" button to login into the system, else if the credentials are incorrect, the system will display an error message alert. Fig 4 Shows the screenshot of hospital staff login page.

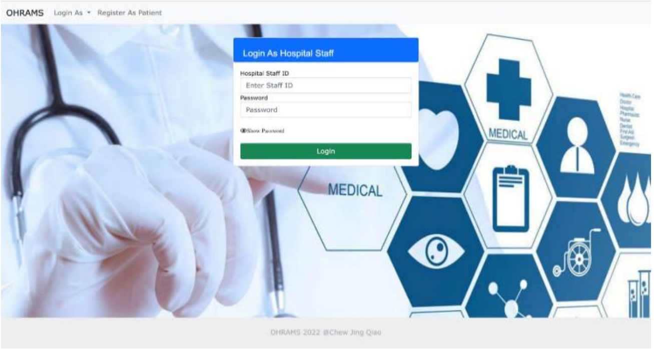


Fig. 4. Screenshot of hospital staff login page

D. Screenshot for Doctor Login Page

This Doctor Login page serves the purpose for doctor users to login into the system. Doctors must enter the correct Doctor ID and password and click the "Login" button to login into the system, else if the credentials are incorrect, the system will display an error message alert. Fig 5 shows the screenshot of doctor login page.

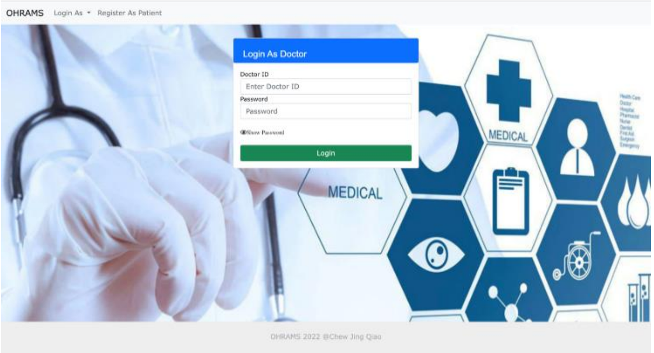


Fig. 5. Screenshot of doctor login page

This Doctor Login page serves the purpose for doctor users to login into the system. Doctors must enter the correct Doctor ID and password and click the "Login" button to login into the system, else if the credentials are incorrect, the system will display an error message alert.

E. Screenshot for Patient Login Page

This Patient Login page serves the purpose for doctor users to login into the system. Patients must enter the correct IC Number and password and click the "Login" button to login into the system, else if the credentials are incorrect, the system will display an error message alert. Fig 6 shows the screenshot of patient login page.

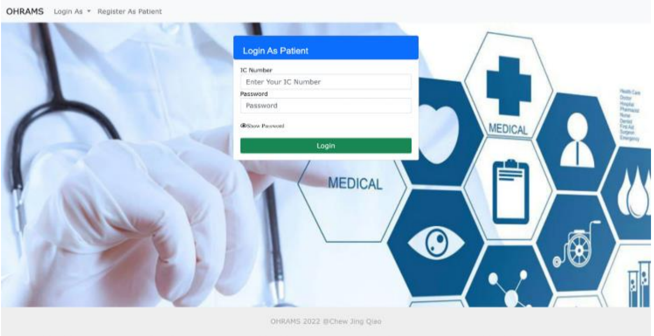
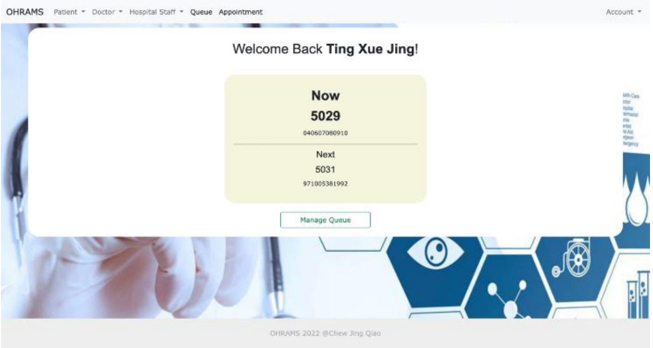


Fig. 6. Screenshot of patient login page

F. Screenshot for Hospital Staff Dashboard Page

This Hospital Staff Dashboard Page is accessible for hospital staff user type. This page shows the current queue status with the current queue and next queue. Besides that, by clicking the "Manage Queue" button, the system will redirect to Queue Management Page. In addition, the navigation bar on top can be redirected to different pages. paper is styled. Fig 7. shows the screenshot of hospital staff dashboard page.



Welcome Back **Ting Xue Jing**!

Now

5029

040607080910

Next

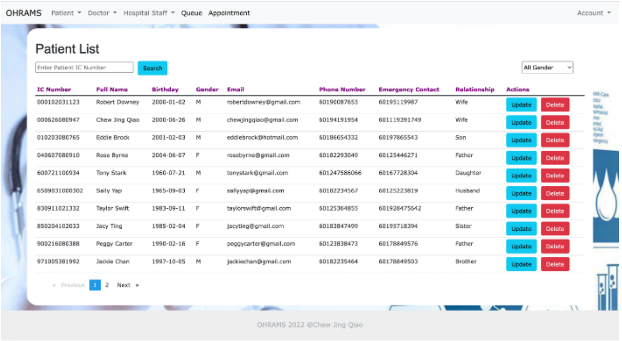
5031

971005381992

Fig. 7. Screenshot of hospital staff dashboard page

G. Screenshot of Patient List – User Hospital Staff

This Patient List page can be accessed by hospital staff user type. This page shows the list of patients in the table. Besides that, the drop down menu can be used to filter the list into list of all male patients, list of all female patients, and list of all patients. Moreover, hospital staffs can search for patient profile by entering a valid IC Number in the text field and click search. Furthermore, hospital staffs can update patient profile by clicking the "Update" button and delete patient account by clicking "Delete" button. Lastly, the pagination at the bottom of the table is used to switch from pages to pages. Fig 8. shows the screenshot of patient list.



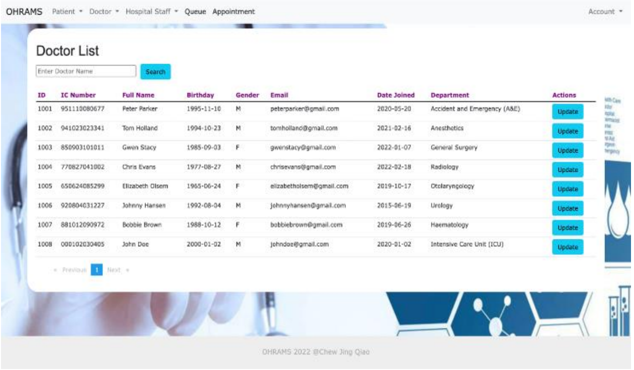
Patient List

IC Number	Full Name	Birthday	Gender	Email	Phone Number	Emergency Contact	Relationship	Actions
000102031123	Robert Downey	2000-01-02	M	robertdowney@gmail.com	60190087653	60195119987	Wife	Update Delete
000626080947	Chew Jing Qiao	2000-06-26	M	chewjingqiao@gmail.com	60194191954	601119391749	Wife	Update Delete
010203080765	Eddie Brock	2001-02-03	M	eddiebrock@hotmail.com	60186654332	60197865543	Son	Update Delete
040607080910	Rose Byrne	2004-06-07	F	rosebyrne@gmail.com	60182293049	60125446271	Father	Update Delete
600721100934	Tony Stark	1960-07-21	M	tonystark@gmail.com	601247586066	60167728304	Daughter	Update Delete
6509031000302	Sally Yap	1965-09-03	F	sallyyap@gmail.com	60182234567	60125223819	Husband	Update Delete
830911021332	Taylor Swift	1983-09-11	F	taylorswift@gmail.com	60125364855	601928475642	Father	Update Delete
850204102033	Jacy Ting	1985-02-04	F	jacyting@gmail.com	60183847499	60195718394	Sister	Update Delete
900216080388	Peggy Carter	1990-02-16	F	peggycarter@gmail.com	60123838473	60178849576	Father	Update Delete
971005381992	Jackie Chan	1997-10-05	M	jackiechan@gmail.com	60182235464	60178849503	Brother	Update Delete

Fig. 8. Screenshot of patient list

H. Screenshot of Doctor List Page – User Hospital Staff

This Doctor List page can be accessed by hospital staff user type. This page shows the list of doctors in the table. Moreover, hospital staffs can search for doctor profile by entering a valid Doctor Name in the text field and click search. Furthermore, hospital staffs can update doctor profile by clicking the "Update" button. Lastly, the pagination at the bottom of the table is used to switch from pages to pages. Fig 9. shows the screenshot of doctor list page.



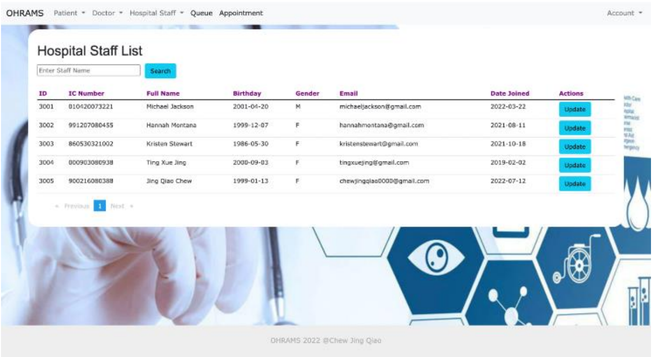
Doctor List

ID	IC Number	Full Name	Birthday	Gender	Email	Date Joined	Department	Actions
1001	951110080677	Peter Parker	1995-11-10	M	peterparker@gmail.com	2020-05-20	Accident and Emergency (A&E)	Update
1002	941023023341	Tom Holland	1994-10-23	M	tomholland@gmail.com	2021-02-16	Anesthetics	Update
1003	850903101011	Gwen Stacy	1985-09-03	F	gwenstacy@gmail.com	2022-01-07	General Surgery	Update
1004	770827041002	Chris Evans	1977-08-27	M	chrisevans@gmail.com	2022-02-18	Radiology	Update
1005	650624085299	Elizabeth Olsem	1965-06-24	F	elizabetholsem@gmail.com	2019-10-17	Otolaryngology	Update
1006	920804031227	Johnny Hansen	1992-08-04	M	johnnyhansen@gmail.com	2015-06-19	Urology	Update
1007	881012090972	Bobbie Brown	1988-10-12	F	bobbiebrown@gmail.com	2019-06-26	Haematology	Update
1008	000102030405	John Doe	2000-01-02	M	johndoe@gmail.com	2020-01-02	Intensive Care Unit (ICU)	Update

Fig. 9. Screenshot of doctor list page

I. Screenshot of Hospital Staff List Page – User Hospital Staff

This Hospital Staff List page can be accessed by hospital staff user type. This page shows the list of hospital staffs in the table. Moreover, hospital staffs can search for staff profile by entering a valid Staff Name in the text field and click search. Furthermore, hospital staffs can update staff profile by clicking the "Update" button. Lastly, the pagination at the bottom of the table is used to switch from pages to pages. Fig 10. shows the screenshot of hospital staff list page.



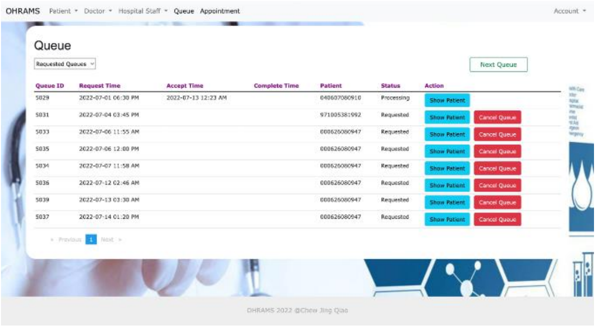
Hospital Staff List

ID	IC Number	Full Name	Birthday	Gender	Email	Date Joined	Actions
3001	010420073221	Michael Jackson	2001-04-20	M	michaeljackson@gmail.com	2022-03-22	Update
3002	991207080455	Hannah Montana	1999-12-07	F	hannahmontana@gmail.com	2021-08-11	Update
3003	860530321002	Kristen Stewart	1986-05-30	F	kristenstewart@gmail.com	2021-10-18	Update
3004	000903080938	Ting Xue Jing	2000-09-03	F	tingxuejing@gmail.com	2019-02-02	Update
3005	900216080388	Jing Qiao Chew	1999-01-13	F	chewjingqiao0000@gmail.com	2022-07-12	Update

Fig. 10. Screenshot of Hospital staff list page

J. Screenshot of Queue Management Page – User Hospital Staff

This Queue Management page is accessible by hospital staff users. This page shows the list of queues in the table. In addition, the drop down menu is used to filter the list to show the list of requested queues, list of incoming queues, list of complete queues, and list of all queues. Besides that, hospital staffs can click the "Next Queue" button to end current queue to move to proceeding queue. Furthermore, hospital staffs can click the "Show Patient" button or click the Patient IC Number and the system will redirect to the Patient Profile page of the selected patient. Moreover, the "Cancel Page" button is used to delete the selected queue. Lastly, the pagination at the bottom of the table is used to switch from pages to pages. Fig 11. shows the screenshot of queue management.



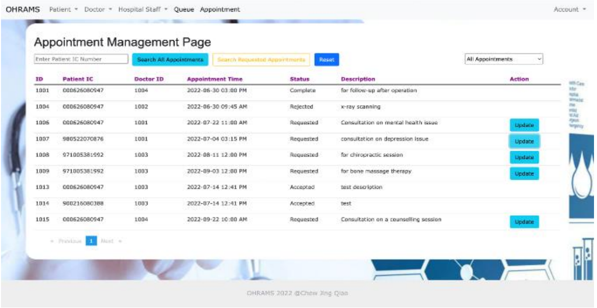
Queue

Queue ID	Request Time	Accept Time	Complete Time	Patient	Status	Action
5029	2022-07-01 06:30 PM	2022-07-13 12:23 AM		040607080910	Processing	Show Patient
5031	2022-07-04 03:45 PM			971005381992	Requested	Show Patient Cancel Queue
5033	2022-07-06 11:55 AM			000626080947	Requested	Show Patient Cancel Queue
5035	2022-07-06 12:00 PM			000626080947	Requested	Show Patient Cancel Queue
5034	2022-07-07 11:58 AM			000626080947	Requested	Show Patient Cancel Queue
5036	2022-07-12 02:46 AM			000626080947	Requested	Show Patient Cancel Queue
5039	2022-07-13 03:30 AM			000626080947	Requested	Show Patient Cancel Queue
5037	2022-07-14 01:20 PM			000626080947	Requested	Show Patient Cancel Queue

Fig. 11. Screenshot of queue management

K. Screenshot of Appointment Management Page – User Hospital Staff

This Appointment Management page is accessible by hospital staff users. This page shows the list of appointments in the table. In addition, the drop down menu is used to filter the list to show the list of requested appointments, list of accepted appointments, list of rejected appointments, list of completed appointments, and list of all appointments. Besides that, hospital staffs can enter a valid Patient IC Number and click the "Search All Appointments" button to show list of all appointments under the specific patient whereas to click the "Search Requested Appointments" to show list of requested appointments under the specific patient. Furthermore, the reset button is used to reset the table in the page to show all appointments. If the hospital staff clicks the "Update" button, the system will display a pop-up box for updating appointment details. Lastly, the pagination at the bottom of the table is used to switch from pages to pages. Fig 12. shows the screenshot of appointment management.



Appointment Management Page

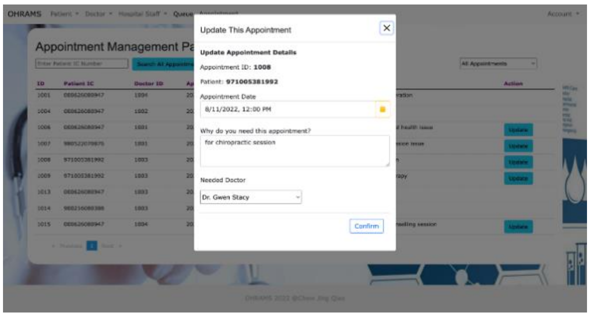
Enter Patient IC Number

ID	Patient IC	Doctor ID	Appointment Time	Status	Description	Action
1001	000626080947	1004	2022-06-30 03:00 PM	Complete	for follow-up after operation	
1004	000626080947	1002	2022-06-30 09:45 AM	Rejected	x-ray scanning	
1006	000626080947	1001	2022-07-22 11:00 AM	Requested	Consultation on mental health issue	Update
1007	980522070876	1001	2022-07-04 03:15 PM	Requested	consultation on depression issue	Update
1008	971005381992	1003	2022-08-11 12:00 PM	Requested	for chiropractic session	Update
1009	971005381992	1003	2022-09-03 12:00 PM	Requested	for bone massage therapy	Update
1013	000626080947	1003	2022-07-14 12:41 PM	Accepted	test description	
1014	900216080388	1003	2022-07-14 12:41 PM	Accepted	test	
1015	000626080947	1004	2022-09-22 10:00 AM	Requested	Consultation on a counselling session	Update

Fig. 12. Screenshot of appoinment management

L. *Screenshot of Update Appointment Pop- up Box – User Hospital Staff*

This Update Appointment Pop-up Box is accessible by hospital staff users. This page is used to update details of a specific appointment by modifying the desired input fields and then click the “Confirm” button to update. In addition, system will display error message alert if there are incomplete fields. Fig 13. shows the screenshot of appointment update.



Update This Appointment

Update Appointment Details

Appointment ID: **1008**

Patient: **971005381992**

Appointment Date: 8/11/2022, 12:00 PM

Why do you need this appointment? for chiropractic session

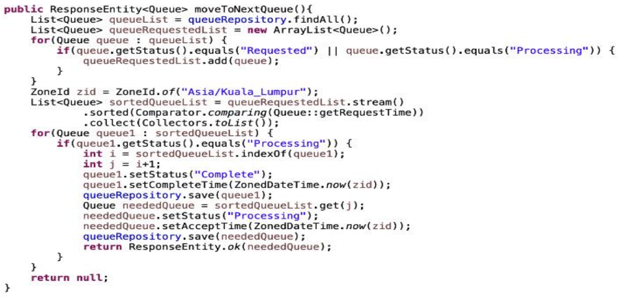
Needed Doctor: Dr. Gwen Stacy

Fig. 13. Screenshot of appoinment update

V. Sample Code

A. *Sample Codes Written for Ending Current Queue To Proceed Next Queue*

Fig 14. shows the code snippet of end current queue to move to next queue. The purpose of this code is to end the current queue and to move the next queue forward and set as current queue. Firstly, the queue list is retrieved and stored in a list. Then, an empty array list called “queueRequestedList” is declared. After that, a for loop is used to filter out the queue that has the status of “Requested” or “Processing”, which are then stored into the “queueRequestedList”.



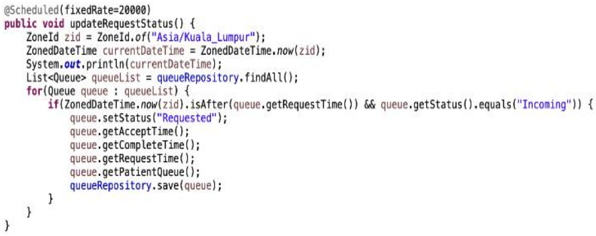
```java
public ResponseEntity<Queue> moveToNextQueue(){
    List<Queue> queueList = queueRepository.findAll();
    List<Queue> queueRequestedList = new ArrayList<Queue>();
    for(Queue queue : queueList) {
        if(queue.getStatus().equals("Requested") || queue.getStatus().equals("Processing")) {
            queueRequestedList.add(queue);
        }
    }
    ZoneId zid = ZoneId.of("Asia/Kuala_Lumpur");
    List<Queue> sortedQueueList = queueRequestedList.stream()
            .sorted(Comparator.comparing(Queue::getRequestTime))
            .collect(Collectors.toList());
    for(Queue queue1 : sortedQueueList) {
        if(queue1.getStatus().equals("Processing")) {
            int i = sortedQueueList.indexOf(queue1);
            int j = i+1;
            queue1.setStatus("Complete");
            queue1.setCompleteTime(ZonedDateTime.now(zid));
            queueRepository.save(queue1);
            Queue neededQueue = sortedQueueList.get(j);
            neededQueue.setStatus("Processing");
            neededQueue.setAcceptTime(ZonedDateTime.now(zid));
            queueRepository.save(neededQueue);
            return ResponseEntity.ok(neededQueue);
        }
    }
    return null;
}
```

Fig. 14. Code Snippet of End Current Queue and Move To Next Queue

Next, this “queueRequestedList” is being sorted in ascending order by comparing the “requestTime” attribute. After that, the status of the queue with the status of “Processing” is being set to “Complete”, which as well set the “completeTime” as the date time now with the Zone ID of “Asia/Kuala_Lumpur”. Next, the following queue is being updated by setting the status as “Processing” and the “acceptTime” is set as date time now, which means this queue is being moved forward as the new current ongoing queue.

B. *Sample Codes Written for Scheduling Update Incoming Queue Status*

Fig 15. shows the code snippet of scheduling update incoming queue status. The purpose of this code is to schedule an update the incoming queue status when reach the current date time. Firstly, the annotation @Scheduled that is used is to execute this function in a fixed rate, which in this case is 20 seconds. A variable named “currentDateTime” is declared to store the current date and time with the Zone ID of “Asia/Kuala_Lumpur”. Next, a variable named “queueList” is declared to store all queues. After that, for each queue in the “queueList”, if the queue has passed the current date time and the status is equals to “Incoming”, this queue will be updated in which the status will be set as “Requested”.

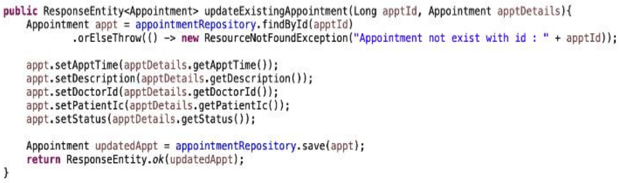


```java
@Scheduled(fixedRate=20000)
public void updateRequestStatus() {
    ZoneId zid = ZoneId.of("Asia/Kuala_Lumpur");
    ZonedDateTime currentDateTime = ZonedDateTime.now(zid);
    System.out.println(currentDateTime);
    List<Queue> queueList = queueRepository.findAll();
    for(Queue queue : queueList) {
        if(ZonedDateTime.now(zid).isAfter(queue.getRequestTime()) && queue.getStatus().equals("Incoming")) {
            queue.setStatus("Requested");
            queue.getAcceptTime();
            queue.getCompleteTime();
            queue.getRequestTime();
            queue.getPatientQueue();
            queueRepository.save(queue);
        }
    }
}
```

Fig. 15. Code Snippet of Scheduling Update Incoming Queue Status

C. *Sample Codes Written for Updating Details of Existing Appointments*

Fig 16. shows the code snipper of updating the details of existing appointments. The purpose of this code is to allow users to update details of specific appointment. Firstly, this function will takes in two parameters, which are “apptId” and “apptDetails”, which stands for Appointment ID and appointment details respectively. Next, a variable named “appt” is declared to store the appointment through finding by ID using the “apptId” parameter. If the “apptId” does not exist, it will then throw a “ResourceNotFoundException” exception. If the appointment is found, the new details are then stored in the appointment that is declared as “appt”, which is then being updated in the database.



```java
public ResponseEntity<Appointment> updateExistingAppointment(Long apptId, Appointment apptDetails){
    Appointment appt = appointmentRepository.findById(apptId)
            .orElseThrow(() -> new ResourceNotFoundException("Appointment not exist with id : " + apptId));

    appt.setApptTime(apptDetails.getApptTime());
    appt.setDescription(apptDetails.getDescription());
    appt.setDoctorId(apptDetails.getDoctorId());
    appt.setPatientIc(apptDetails.getPatientIc());
    appt.setStatus(apptDetails.getStatus());

    Appointment updatedAppt = appointmentRepository.save(appt);
    return ResponseEntity.ok(updatedAppt);
}
```

Fig 16. Code Snippet of Updating Details of Existing Appointments

VI. Conclusion

A. *Critical Evaluation*

This critical evaluation is conducted to serve the purpose to evaluate whether OHRAMS is designed and built successfully to meet the needs of relevant users as well as to evaluate on the benefits that this system will bring to the users in real life deployment. On top of that, by conducting this critical evaluation will also help the developer to learn whether this final system is implemented with all stated objectives and requirements. In a nutshell, carrying out critical evaluation is extremely crucial when comes to designing and building a program to ensure the program fulfil all objectives and will lead to system benefits.

B. System Benefits

Firstly, the check in functionality implemented in OHRAMS will allow patients to check in to a queue now or even to check in for later according to the desired date and time. This functionality will lead to benefits such as preventing long waiting time of the patients when comes to checking in for a consultation. Besides that, patients can also view the status of the queue, which includes the current queue and the next queue. One thing to mention, the queue algorithm implemented in OHRAMS is to allow patients to check in according to their desired time. Therefore, the queue is arranged in an order based on the requested check in date and time. Besides that, hospital staffs will be able to end the current queue which will then move the proceeding queue to the current ongoing queue. Apart from that, patients can always cancel the requested queue, which may be due to certain reasons, such as having an emergency where the patient must leave the queue now.

Furthermore, patients also can book for appointments online through using OHRAMS. Currently, majority of the hospitals will require patients to book for appointments through phone calls or by notifying the doctors by walking in, which is a troublesome process. On top of that, OHRAMS allows doctors to create new medical reports for patients as well as to check their history medical reports. In addition, patients can also view their medical records. Currently, most of the hospitals do not provide this functionality for patients to check their medical records.

C. Conclusion & Reflection

the research on deciding which framework and development software to be used to program this also a crucial aspect. Due to the reason that we use Angular Framework to program the front end of this proposed system, Visual Studio Code is the best choice as the development software. After that, the back end development software that we decided to use is Eclipse as this software is easy to use, which as well we have more experience on using Eclipse. Therefore, Eclipse is the to-go choice to program the back end of my proposed system. In addition, Spring Boot is used as the tool to program my back end. As for the database management system, MySQL workbench is the used to manage the data of the proposed system. By the end of the project, we had successfully gain more knowledge and experience on using these development software. Most importantly, we had also learnt more on programming by using Angular Framework and Spring Boot tool, which will be extremely useful in my future career life. In a nutshell, we gained the experience to program a system in full stack, which includes coding in TypeScript, HTML, and CSS for front end, Java for backend, and SQL for the database.

Furthermore, the documentation also included on conducting unit testing in the test plan section. Through carrying out unit testing, we had also gain the experience and knowledge on code tracing to find the code that is causing the bugs and errors. This skills is also important when comes to system or software maintaining to fix bugs and errors. On top of that, the user acceptance testing also provide a chance for the developer to modify the functions to fulfil the needs of the testers.

D. Limitation & Future Enhancement

Although the proposed system is successfully designed and developed, but there are still limitations. Taken that into account, user acceptance testing is performed for a reason. As stated by a tester, the proposed system needs to be implemented with a feature to check the position of the requested queue, in order to know the number of queues that is in front of the requested queue. Apart from that, a recommendation from tester stated that whenever an appointment request is accepted by the doctor, the patient should also receive a queue number for the day of the appointment. Other than that, testers had gave an overall great rating on the proposed system.

References

Nunes, G. S., & Javier, M. G. F. (2014). Hospital Information System Satisfaction in Brazil: Background and Moderating Effects. International Journal of Research Foundation of Hospital and Healthcare Administration, 2(1), 1–9. https://doi.org/10.5005/jp-journals- 10035-1007

M, D. S. O. O. E. C. (2018). Interlinked Hospital Management System. International Journal of Science and Research (IJSR), 7(2), 1378–1383. https://www.researchgate.net/publication/334587568_Interlinked_Hospital_Management_ System

Abraham, I., & Joyce, A. C. (2016). Designing A Web Based Hospital Management System For MOUAU Clinic. International Journal of Trend in Research and Development, 2(6), 2394–9333. www.ijtrd.com

Gadhari, D. H., Kadam, Y. P., & Suman, P. (2016). Hospital management system 1 1,2,3. International Journal for Research in Engineering Application & Management (IJREAM), 1(11), 1–4. https://www.ijream.org/papers/INJRV01I110 06.pdf

Anpan, P., Udasi, R., Jagtap, S., Thakre, S., & Kamble, C. (2020). Hospital Management System Project. International Research Journal of Engineering and Technology (IRJET), 7(4), 1415–1418. https://www.researchgate.net/publication/343 149774_Hospital_Management_System

Mun, Y. T. (2022). Implementation of Identity and Access Management Application in Learning Management System to Mitigate Credential Hacking Issues. Journal of Applied Technology and Innovation, 6(4), 10-15.

Eria, K. (2018). Security and Privacy of Health Data: A Review of the Challenges and Approaches. Journal of Applied Technology and Innovation, 2(2), 26-31.