

The PKU System: Development of Online UTHM Health Center Appointment System

Tan Foo Yuan¹, Salama A. Mostafa^{1*}

¹Fakulti Sains Komputer dan Teknologi Maklumat,
Universiti Tun Hussein Onn Malaysia, Parit Raja, Batu Pahat, 86400, MALAYSIA

DOI: <https://doi.org/10.30880/aitcs.2022.03.02.059>

Received 3 August 2022; Accepted 28 October 2022; Available online 30 November 2022

Abstract: Medical appointment is the process of schedule a formal arrangement for an individual to meet doctor at a particular time. UTHM Health Center is chosen as the case study in this project as it provides health and medical services for UTHM students. In this project, The PKU System was proposed to improve the appointment management process efficiency and reduce the time consumed for the process. The intended users of the system are UTHM students and health center staff. The system helps students to easily book appointments via online. The software development methodology that used to develop this project is object-oriented model due to fast development times and easy integration of existing software modules. The development of this system provides an online platform for the health center staff to manage the appointments applied by the students.

Keywords: Appointment Management System, Mobile-based, Healthcare

1. Introduction

Appointment is a formal arrangement to meet or visit someone at a specific time, usually for a reason related to their work [1]. Online appointment system is one of the most used web-based applications and enables individuals to book their reservations and requests online securely and conveniently via digital devices such as laptop and smartphone. It has a highly efficient set of management tools to synchronize, computerize, and systematically record data assisted using Internet websites. Online appointment system can effectively increase customer satisfaction compared with manual appointment methods [2]. It has shown positive changes such as reduced customer no-show rate, decreased staff labor and decreased customer waiting time. Online appointment systems are widely used in hotels and hospitals to schedule appointments and keep track of patient records [3].

Health center is a building in which a group of doctors have offices or surgeries where their patients can visit them [4]. In this project, the case study selected is UTHM Health Center which is the department that provides health and medical services for UTHM students. Usually, students who suffer from sickness or pain and want to get treatment or advice for their health problems will go to the health center to meet with doctors. Nowadays, due to COVID-19 pandemic, people need to maintain social distance and follow standard operating procedures to minimize the risk of virus spread. Thus, students need to make appointment with doctor before going to the health center to prevent crowding at the counter. However, the current process of making appointments is that students need to make

*Corresponding author: salama@uthm.edu.my

phone calls to book appointments with staff. After that, the staff will record the appointment information in the record book.

There are some problems that occur in the current system. Firstly, the staff may miss some students' phone calls when they are busy, or the phone call is made outside of their working hours. Thus, the students will feel disappointed as they can't make appointments to get medical treatment. Next, the data loss problem occurs regularly because all the data such as student details and appointment information are stored in paperwork. Besides, the staff need to spend a lot of time retrieving appointment information from the record book when record checking is needed.

Therefore, The PKU System is an Android based application that will be developed in this project to assist students to easily book appointments via internet. A database is created to store the appointment information in systematic and organized ways. This application can reduce the stress on staff as a lot of work is automated. It is also easy to retrieve appointment information compared to manually method use. This application is also able to spread out students' visit throughout the day and reduce their waiting time as they can schedule their arrival close to their appointment time, which can considerably prevent crowding at the counter.

There are three main objectives in this project. The first objective is to design an online appointment system for UTHM Health Center using object-oriented approach. The second objective is to develop an online appointment system for UTHM Health Center using Android. The third objective is to test the functionality of the online appointment system using functional testing and user acceptance testing.

This article is organized into five sections. The first part is an introduction describing the context of the project. The second section describes the analysis of the relevant work. In the third section, the methodology, analysis, and design for the system development are explained. The fourth part discusses the system implementation and testing process carried out to test the developed system. In the last section, a conclusion with some instructions for future employment is given.

2. Related Work

Four existing applications are chosen to study and compare their advantages and disadvantages related to the proposed application. Table 1 shows the comparison between the existing applications and the proposed application based on several features.

2.1 Mr. Doc: A Doctor Appointment Application

This system is an android-based application that makes scheduling a doctor's appointment simple and dependable for users [5]. The system user module is implemented in mobile application, while the administrator module is implemented in website. Data is shared between both the website and mobile application by using API. There are several features and functions that are developed in this system. The first feature is login. This feature only allows authorized users to login into the system using the login information such as username, email, and password. Users must register in the application on first use. The login information is different for both user and administrator as their authority in accessing and performing functions are different. When they enter the correct information, they are directed to the main menu screen. In the user module, they can view the hospital details by choosing the hospital from the hospital list. Google map is provided in the system for users to view the location of the hospital. Besides that, users can select any doctor from the list of the available doctors and view the doctor's profile. Users can send a request for appointment by clicking the book appointment button in the selected doctor details page. They can get a notification message if their appointment booking is successful. User module also includes a health information function that provides a different health schedule for each age group. For the administrator module, they can register doctors into the system. They also can view the patients' details and the appointment details.

2.2 Clinic Appointment Scheduling System

This system is a mobile application focused on the schedule of the patient appointment [6]. The system is applied to priority scheduling with first come first serve to manage the appointment booking section. Priority scheduling method will function based on the priority level of the patient booking. All users must register to authenticate the user to access the system. In the administrator module, the administrator can manage the patient, doctor and appointment requested by the patient. Administrator also can generate report by retrieving data from patient records in the database. In the patient module, patients can update their profile details. Patients also can request appointment and view the confirmation after the administrator approves the appointment. In the doctor module, doctors can update their profile information. Doctors can manage appointments after administrator assign the appointment to attend. Doctors also can update the condition of patient and save the info to the patient record.

2.3 Appointment Consultation System

This system is a web-based application developed with the aim of making it easier for Sultan Zainal Abidin University students to book appointments online with their lecturers [7]. The system is developed using Web Modelling Language. The login feature in the system allows users and administrator to use the system. The login information is different for both user and administrator as their authority in accessing and performing functions are different. For first time users, they need to register before they can login into the system. The system is used by three users namely students, lecturers, and administrator. Firstly, in the student interface, students can update their profile information. Students can check the status of the appointment and book the available appointment date. They also can check their appointment history and view the comments provided by the lecturers after the appointment. Next, in the lecturer interface, lecturers can update their profile information. Lecturers can manage and reschedule the appointment date whether the appointment is still available or has been booked by the student. Lecturers also can view the precious appointment records of students. Lastly, in the administrator interface, administrator can register and manage students and lecturers. Administrator also can book and cancel the appointment for the students.

2.4 Square Appointments

This system is a mobile appointment scheduling application that is currently selling in the market [8]. This system contains two types of users which are administrators and clients. The system has good user interface design which the content display is clean and clear for the users to understand. The system also provides tutorials to guide users in the steps of setting up available time slots and appointment types such as consultation, meeting, discussion and so on. The system supports automated email and text reminders features. Clients can receive appointment reminders via text message and email to remind them to attend their appointments. Besides that, the system is synchronized with Google Calendar. The system has implemented some great features such as custom cancellation policy and online payment which involves payment to upgrade the appointment scheduling process. However, most of the features are not suitable to apply in healthcare applications as this system focuses more on business.

2.5 Comparison of Related System

Table 1: System Comparison

Features	Mr. Doc: A Doctor Appointment Application	Clinic Appointment Scheduling System	Appointment Consultation System	Square Appointments	The PKU System
Login	Yes	Yes	Yes	Yes	Yes
Registration	Yes	Yes	Yes	Yes	Yes
Appointment booking	Yes	Yes	Yes	Yes	Yes

Table 1: (cont)

Features	Mr. Doc: A Doctor Appointment Application	Clinic Appointment Scheduling System	Appointment Consultation System	Square Appointments	The PKU System
Appointment management	No	Yes	Yes	Yes	Yes
Report generation	No	Yes	No	No	Yes
Reminder	No	No	No	Yes	No
Notification	Yes	No	No	No	Yes
Usability platform	Android app	Mobile-based	Web-based	Mobile-based	Android app

According to the comparison results, most existing applications include login, registration, appointment booking, and appointment management features. Despite the similarities, Clinic Appointment Scheduling System supports report generation feature. Besides, the reminder feature is only implemented in Square Appointments. Other than that, Mr. Doc: A Doctor Appointment Application also apply notification feature. The study of these existing appointment scheduling applications offers a wide range of features that are suitable for the proposed application. The proposed application is developed with the goal of including login, registration, appointment booking, appointment management, report generation, and notification features.

3. Methodology

3.1 Project Planning

In the project, object-oriented model [9] is selected as the methodology to develop The PKU System. Object-oriented model is divided into several phases which are planning phase, analysis phase, design phase, implementation phase and testing phase. Table 2 shows the activity and deliverables of each phase that need to be produced during the project development.

Table 2: Software Development Activities and Their Deliverables

Phase	Activity	Deliverable
Planning	<ul style="list-style-type: none"> Discuss project title and case study Identify problem, objectives, and project scope Determine project schedule 	<ul style="list-style-type: none"> Project proposal Gantt chart
Analysis	<ul style="list-style-type: none"> Review and analyse existing system Interview stakeholder Analyse the requirements Determine the functional and non-functional requirement Specify software and hardware used in the system 	<ul style="list-style-type: none"> Software Requirement Specification Document UML diagrams Class diagrams Requirement Traceability Matrix To-be model
Design	<ul style="list-style-type: none"> Design system architecture Design user interface Design database 	<ul style="list-style-type: none"> Software Design Document System architecture Database design User interface design
Implementation	<ul style="list-style-type: none"> Implement the design into a prototype 	<ul style="list-style-type: none"> Programming code System prototype
Testing	<ul style="list-style-type: none"> Test the system functionality 	<ul style="list-style-type: none"> Test cases and results Final system product

3.2 Requirement Analysis

Requirement analysis is the process of determining user expectations for a developed or proposed product [10]. It includes all the tasks that are carried out to identify the needs of various stakeholders. System requirements usually contain functional and non-functional requirements, user requirements and system requirements. Table 3 summarizes the functional modules provided in the system.

Table 3: System Functional Module

No.	Module	Function	User
1.	Login module	Allow administrator and student to login to the system	Administrator, Student
2.	Registration module	Allow student to register to the system	Student
3.	Appointment booking module	Allow student to book appointment	Student
4.	Appointment management module	Allow administrator to accept, reject, or reschedule the appointment Allow student to cancel the appointment	Administrator Student
5.	Report generation module	To generate report	Administrator
6.	Notification module	To inform student any update about the appointment	Student

Functional requirement captures the system's intended behaviour such as services, tasks, or functions that the system is expected to perform [11]. Table 4 shows the functional requirements of the proposed system.

Table 4: Functional Requirements

No	Module	Description
1.	Login module	<ul style="list-style-type: none"> The system should allow user to login into the system using registered user id and password. The system should only allow authorized user with a valid user id and password to log in. The system should notify the user if there is any invalid input. The system should redirect user to the homepage upon successful login.
2.	Registration module	<ul style="list-style-type: none"> The system should allow student to register account with their personal information such as username, matrix number, and contact number.
3.	Appointment booking module	<ul style="list-style-type: none"> The system should allow students to select date from the calendar. The system should allow students to select the available time slot. The system should allow students to book appointment based on the time slot provided.
4.	Appointment management module	<ul style="list-style-type: none"> The system should allow administrator to accept or reject the appointment request. The system should allow administrator to view and reschedule the appointments booked by students. The system should allow students to cancel their appointments.
5.	Report generation module	<ul style="list-style-type: none"> The system should allow administrator to generate report.
6.	Notification module	<ul style="list-style-type: none"> The system should send notification to students when there is any update about their appointments.

Non-functional requirement is the specification that describes the system's operation capabilities as well as the constraints that improve system functionality [12]. Table 5 shows the non-functional requirements of the developed system.

Table 5: Non-functional Requirements

No	Requirements	Description
1.	Performance	The system should be able to operate 24 hours a day. The system should be able to respond to user action within 3 seconds.
2.	Operational	The system should be able to install and operate in Android system.
3.	Security	The system should be able to protect the user personal information and their appointment information.
4.	Usability	The system should be user friendly and easy to use.

User requirements define what the user does with the system, such as which activities users must be able to perform [13]. Table 6 shows the user requirements of the developed system.

Table 6: User Requirements

No.	User Requirements
1.	All users must be able to enter a valid user id and password to login to the system.
2.	Students should be able to register account.
3.	Students should be able to see the available time slot in a selected date.
4.	Students should be able to book appointment based on the time slot provided.
5.	Students should be able to cancel their appointments.
6.	Students should be able to get a notification when their appointments are accepted, rejected, or rescheduled.
7.	All users should be able to view their profile details.
8.	Administrator should be able to accept or reject the appointment request.
9.	Administrator should be able to view appointments booked by the students.
10.	Administrator should be able to reschedule the appointments.
11.	Administrator should be able to generate report.

3.3 System Analysis

Use case diagram is used to define the system behaviour and represent system's requirements. The interaction between the system and its actors are shown in the use case diagram [14]. There are two actors involved in the use case, which is administrator and students. Besides that, there are a total of 6 use cases representing system functional module in this project. The use cases are registration, login, book appointment, manage appointment, generate report, and notification. The administrator is related to 3 use cases, while the student is related to 5 use cases. Figure 1 shows the use case diagram of The PKU System proposed in this project.

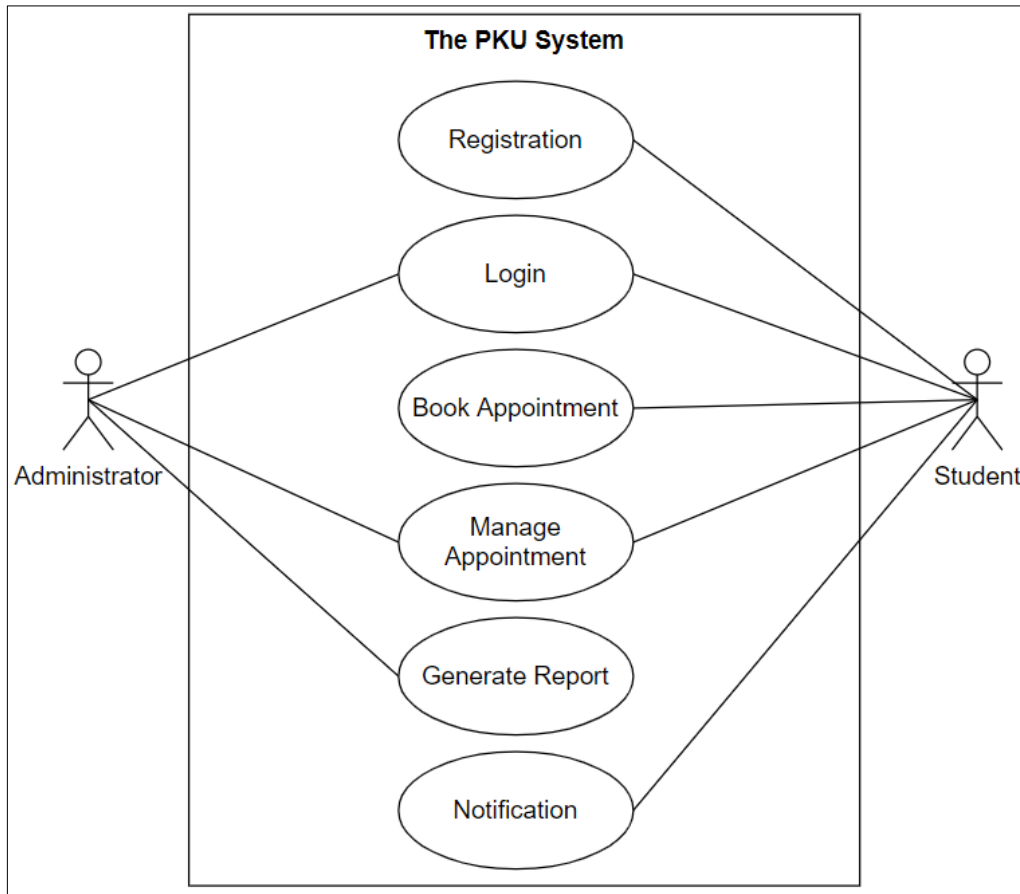


Figure 1: Use Case Diagram

Class diagram describes and visualizes the objects in a system based on their attributes, operations, constraints, and relationships among them [14]. It is used in analysis and design of the static view of the proposed system. Figure 2 displays the class diagram for The PKU System.

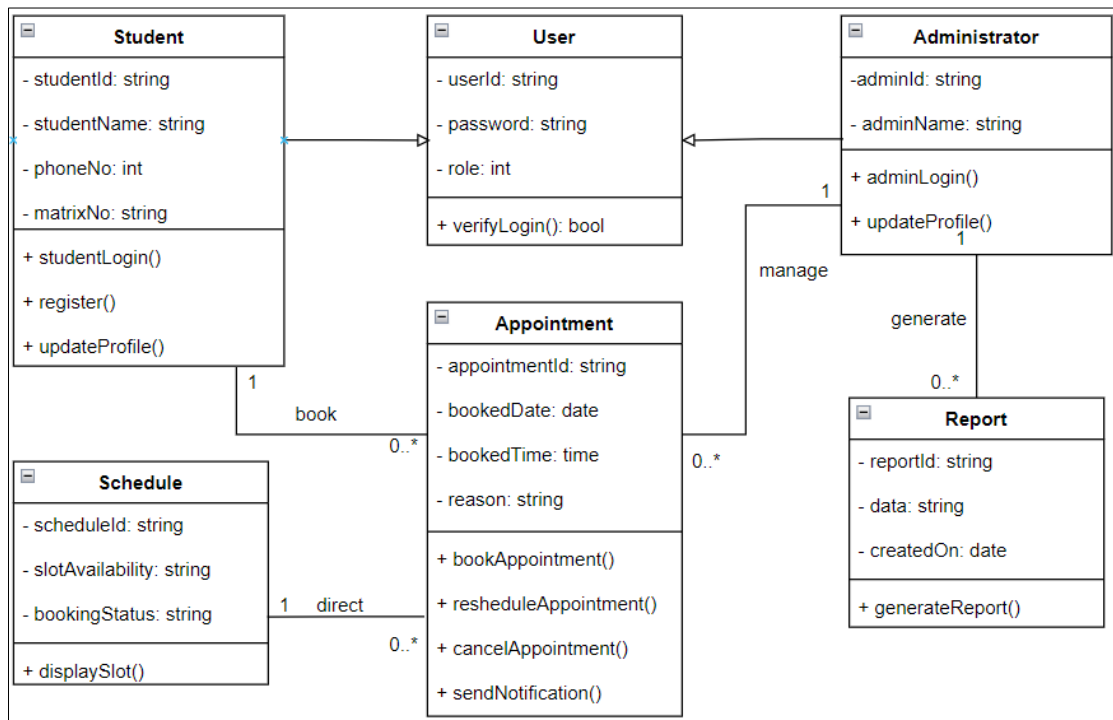


Figure 2: Class Diagram

To-be model shows the proposed future processes of appointment management process in the health center after the implementation of the new system in this project. Figure 3 shows the to-be model of the proposed process of appointments management.

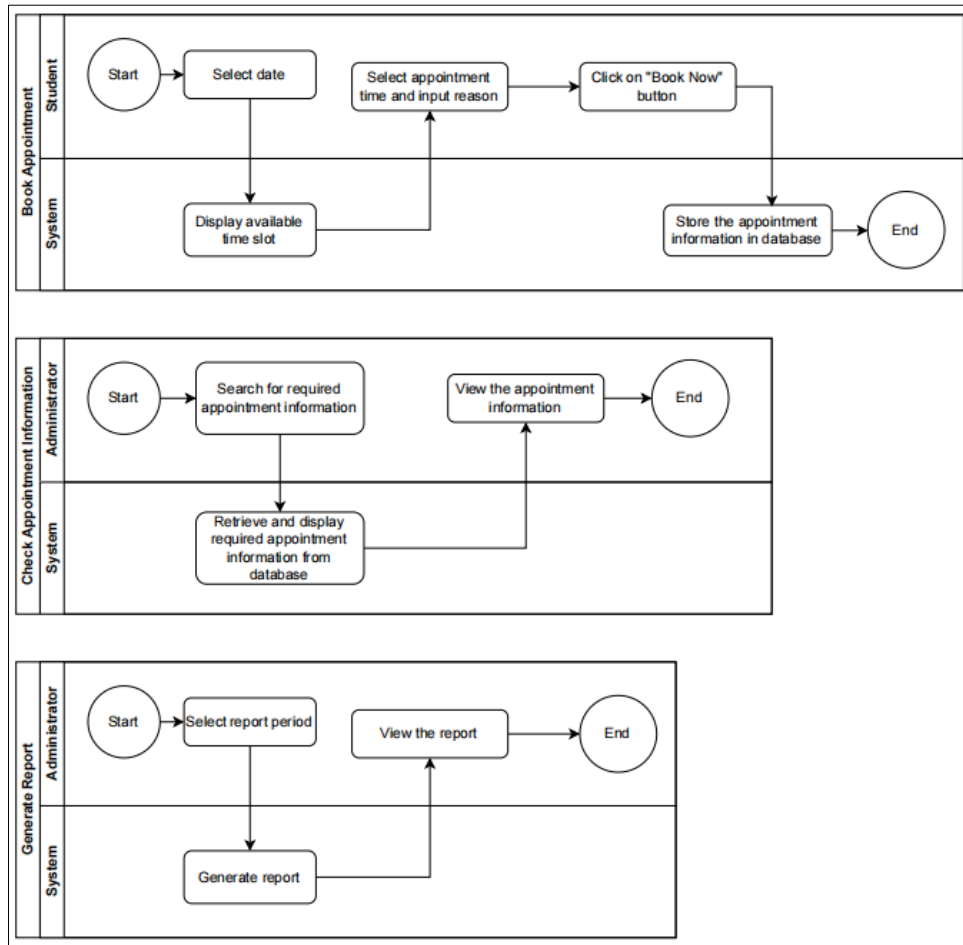


Figure 3: To-be Model

3.4 System Design

System architecture is a conceptual model that describes the structure, behavior, environment, and relationships between system components [15]. MVP architecture pattern is the architecture model implemented in this project. Figure 4 shows the MVP architecture of The PKU System.

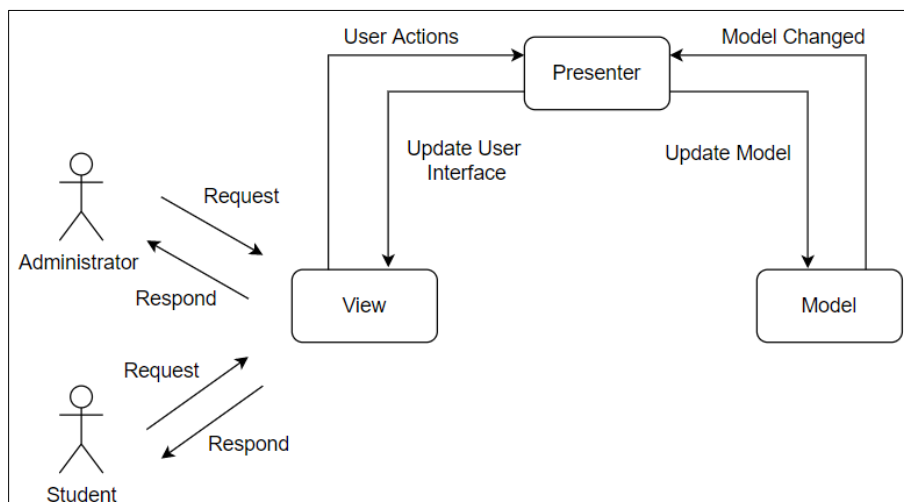


Figure 4: System Architecture

The data schema describes the data organization and tables relationships in the database. The database schema for the database is listed as follows:

- i. Administrator (admin_id, admin_name, admin_password)
- ii. Student (student_id, student_name, phone_no, matrix_no, student_password)
- iii. Appointment (app_id, book_date, book_time, reason, student_id, admin_id, schedule_id)
- iv. Schedule (schedule_id, slot_availability, book_status)
- v. Report (report_id, data, created_on, admin_id)

User interface is the front-end of system view where the user interacts with the system. Figure 5 and Figure 6 show some of the user interfaces for the Administrator and the students.

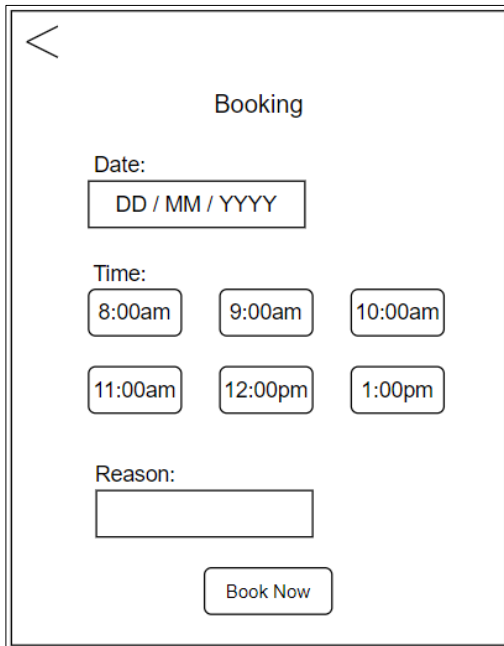


Figure 5: Booking Appointment Page

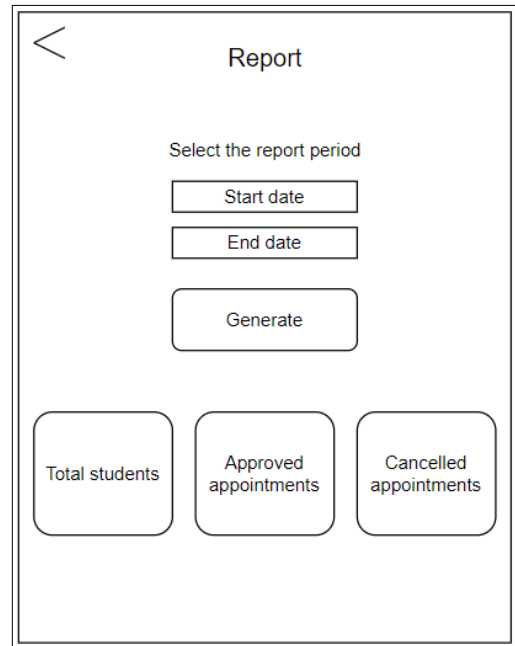


Figure 6: Generate Report Page

4. Results and Discussion

4.1 Implementation

This section shows some examples of the modules of The PKU System which developed using Android Studio platform. In this project the language that was used is Java programming language. The database for this application is using Firebase. The main interfaces and their code segments are shown and discussed.

Login module is the main page when user entering the system. Students need to insert their matric number and password, while administrator needs to insert admin id and password. After that, they need to select their role to allow the system redirects them to their respective homepage. The login process is successful when the data inserted does match with the data in database. Figure 7 shows the interface for the users to login into The PKU System. Figure 8 shows the code segment for login page.

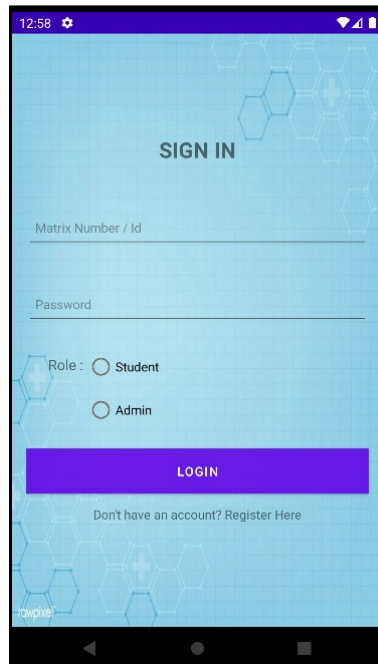


Figure 7: Login Page

```

package com.example.thepku;

import ...

public class Login extends AppCompatActivity {
    EditText mid, mpassword;
    Button mbtnlogin;
    TextView mregistertext;
    RadioGroup rgRole;
    FirebaseFirestore fStore;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_login);
        mid = findViewById(R.id.id);
        mpassword = findViewById(R.id.password);
        mbtnlogin = findViewById(R.id.btnLogin);
        mregistertext = findViewById(R.id.registerText);
        rgRole = findViewById(R.id.rgRole);
        fStore = FirebaseFirestore.getInstance();

        mbtnlogin.setOnClickListener(new View.OnClickListener() {
            public void onClick(View v) {

                String id = mid.getText().toString().trim();
                String password = mpassword.getText().toString().trim();
                int checkedId = rgRole.getCheckedRadioButtonId();
            }
        });
    }
}

```

Figure 8: Code Segment for Login Page

Registration module is only for new students to register account in the system. They need to insert data such as their username, matrix number, phone number and password. If any of the field is empty upon submission, the system will request the student to fill in the empty field. Figure 9 shows the interface of registration page. Figure 10 shows the code segment for registration page.



Figure 9: Registration Page

```

package com.example.thepku;

import ...

public class Register extends AppCompatActivity {
    EditText muserName,matrix,mphone,mpass1,mpass2;
    Button mbtnRegister;
    TextView mloginText;
    FirebaseFirestore fStore;
    String userid;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_register);

        muserName = findViewById(R.id.userName);
        matrix = findViewById(R.id.matrixnum);
        mphone=findViewById(R.id.phone);
        mpass1 = findViewById(R.id.pass1);
        mpass2 = findViewById(R.id.pass2);
        mbtnRegister = findViewById(R.id.btnRegister);
        mloginText = findViewById(R.id.loginText);
        fStore = FirebaseFirestore.getInstance();
    }
}

```

Figure 10: Code Segment for Registration Page

Appointment booking module is only for students to book their appointments. Students need to select date and time and input appointment reason to complete the booking process. Figure 11 shows the interface of appointment booking page. Figure 12 shows the code segment for appointment booking page.

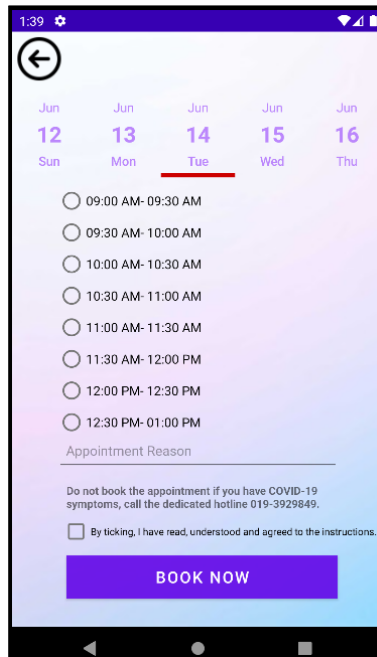


Figure 11: Appointment Booking Page

```

package com.example.thepku;

import ...

public class Stu_booking extends AppCompatActivity {
    TextView t1;
    Button btnbook;
    EditText txtreason;
    RadioGroup rgtime;
    RadioButton r1,r2,r3,r4,r5,r6,r7,r8;
    FirebaseFirestore fStore;
    String time;
    String appdate;
    CheckBox checkBox;
    ImageButton backbtn;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_stu_booking);
        t1=(TextView) findViewById(R.id.textView4);
        rgtime=findViewById(R.id.rgTime);
        txtreason=findViewById(R.id.reason);
        btnbook=findViewById(R.id.button);
        checkBox=findViewById(R.id.checkbox);
        fStore = FirebaseFirestore.getInstance();
        r1=findViewById(R.id.radio1);
        r2=findViewById(R.id.radio2);
    }
}

```

Figure 12: Code Segment for Appointment Booking Page

Appointment management module allows administrator to accept or reject the appointment request from student. Figure 13 shows the interface of appointment request detail page. Figure 14 shows the code segment for appointment request detail page.



Figure 13: Appointment Request Detail Page

```

package com.example.thepku;

import ...

public class Adm_reqdetail extends AppCompatActivity {

    FirebaseFirestore fStore;
    TextView textname, textmatrix, textdate, texttime, textreason, phone;
    Button button2, button3;
    String id;
    ImageButton backbtn;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_adm_reqdetail);

        textname = findViewById(R.id.textname);
        textmatrix = findViewById(R.id.textmatrix);
        textdate = findViewById(R.id.textdate);
        texttime = findViewById(R.id.texttime);
        textreason = findViewById(R.id.textreason);
        button2 = findViewById(R.id.button2);
        button3 = findViewById(R.id.button3);
        phone = findViewById(R.id.phone);
        backbtn = findViewById(R.id.imageButton);
    }
}

```

Figure 14: Code Segment for Appointment Request Detail Page

4.2 Testing

Testing phase is conducted after the application has been developed successfully to validate whether it meets the requirements or not. Testing also is important to ensure the application free from bugs before delivering it to the end user. There are two types of testing have been conducted in this project which are functional testing and user acceptance testing.

Functional testing is concerned with the functional requirements and ensuring that the application functions properly. The system should carry out the functions that satisfy the functional requirements defined in the previous chapter. Table 7 shows the result of the testing for all modules in the application are presented in this section.

Table 7: Test Cases Result of Functional Testing

Test Case ID	Description	Expected Output	Actual Output
TEST_100_001	The user enters valid user id and password.	Success login	As expected output
TEST_100_002	The user enters invalid user id and password.	Unsuccessful login	As expected output
TEST_100_003	Login without entering user id and password.	Unsuccessful login	As expected output
TEST_200_001	The student fills up every field.	Success registration	As expected output
TEST_200_002	The student creates account with no information.	Unsuccessful registration	As expected output
TEST_300_001	The student selects date from calendar.	The system displays available time slot.	As expected output
TEST_300_002	The student books appointment by selecting time slot and filling up appointment reason.	Success booking	As expected output
TEST_300_003	The student books appointment without selects time slot and fills up appointment reason.	Unsuccessful booking	As expected output
TEST_400_001	Administrator approves the appointment request.	The system display pop-up message "Appointment approved".	As expected output
TEST_400_002	Administrator rejects the appointment request.	The system display pop-up message "Appointment rejected".	As expected output
TEST_400_003	Administrator reschedules the approved appointment.	The system display pop-up message "Appointment reschedule successful".	As expected output
TEST_400_004	Student cancels the upcoming appointment.	The system display pop-up message "Appointment cancelled".	As expected output
TEST_500_001	Administrator generates and views appointment report.	Display information accurately	As expected output
TEST_600_001	Student views notification message.	Display notification list	As expected output

The user acceptance testing was conducted using Google Form. There are five respondents involved in this testing process: three respondents are the staff of UTHM health center, and another two respondents are UTHM students. The bar graph in Figure 14 shows the user satisfaction on the system functionality.

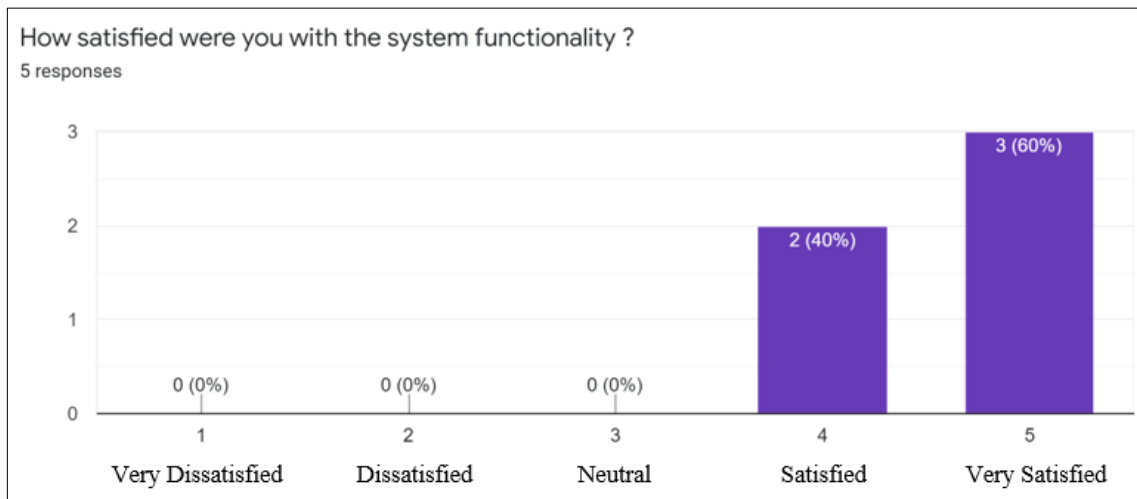


Figure 15: User Satisfaction on The System Functionality

Based on Figure 14, three respondents very satisfied while two respondents satisfied with the system functionality. This computerized management system can improve appointment management for both the health center and the students.

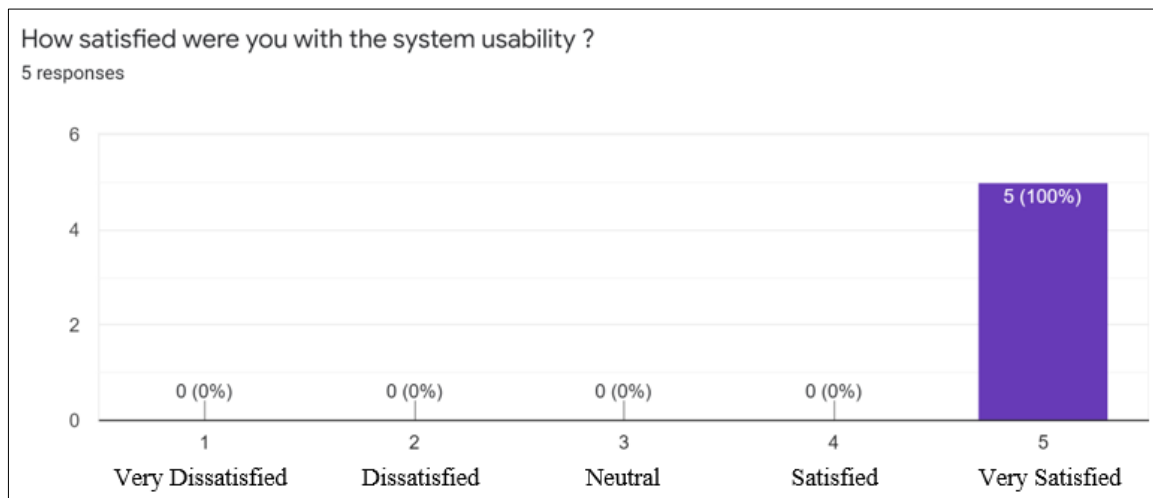


Figure 16: User Satisfaction on The System Usability

Figure 15 shows the result of the system usability testing. All the five respondents are very satisfied with the system usability as they agree that the system is user friendly and simple to learn and use.

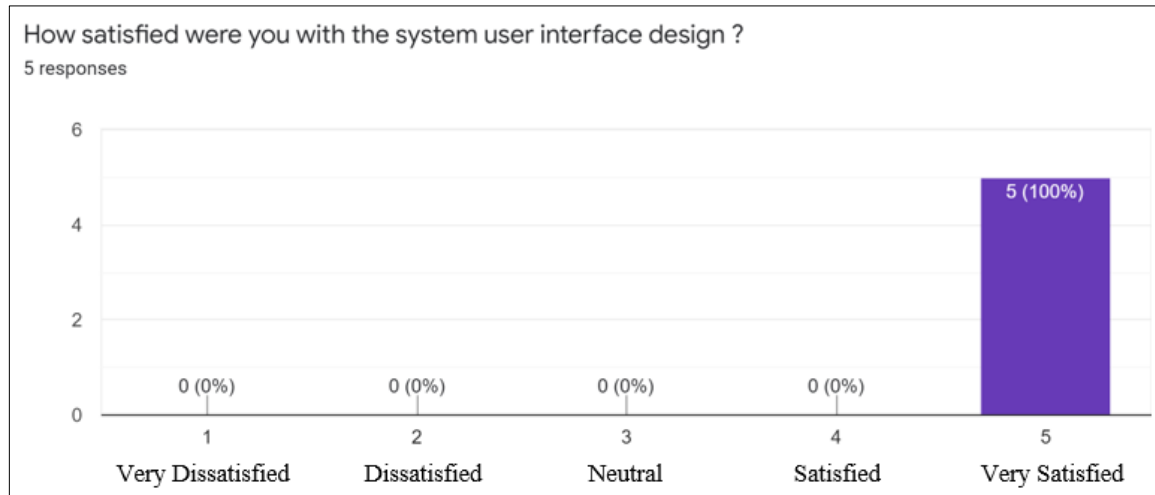


Figure 17: User Satisfaction on The System User Interface Design

Figure 16 shows the result of the system user interface design testing. All the five respondents are very satisfied with the system user interface design. The user interfaces are simple, consistent, and understandable.

5. Conclusion

As conclusion, The PKU System is developed successfully using Java programming language and connected to Firebase database. The application is tested with functional testing and user acceptance testing and proven to achieve the objectives, scopes and expected outcome when all test cases are as expected result. By using this system, UTHM Health Center management would have an online platform to manage appointments from the students. The database system could generate more clear and valuable documentation that records all the appointment details. This application would improve resource allocation and reduce the workloads of health center staff when serving the students.

Acknowledgment

The authors would like to thank the Faculty of Computer Science and Information Technology, Universiti Tun Hussein Onn Malaysia for its support.

References

- [1] Lea, D. & Bradbery, J. (2007). *Oxford Advanced Learner's Dictionary, 10th Edition*. Oxford: Oxford University Press.
- [2] Cao, W., Wan, Y., Tu, H., Shang, F., Liu, D., Tan, Z., ... & Xu, Y. (2011). A web-based appointment system to reduce waiting for outpatients: A retrospective study. *BMC health services research, 11*(1), 1-5.
- [3] Zhao, P., Yoo, I., Lavoie, J., Lavoie, B. J., & Simoes, E. (2017). Web-based medical appointment systems: A systematic review. *Journal of medical Internet research, 19*(4), e134.
- [4] HarperCollins. (2019). Rehabilitate. In *Collins English dictionary* (8th ed., p. 672).
- [5] Malik, S., Bibi, N., Khan, S., Sultana, R., & Rauf, S. A. (2017). Mr. Doc: A doctor appointment application system. *arXiv preprint arXiv:1701.08786*.
- [6] Saifuddin, A. B. (2018). Clinic Appointment Scheduling System (CASS). *University Sultan Zainal Abidin*.
- [7] Hussein, M. F. B. M. (2020). Appointment Consultation System. *University Sultan Zainal Abidin, 2010*.

- [8] *Square Appointments*. (2021). Retrieved on November 7, 2021, from <https://squareup.com/us/en/appointments>
- [9] Acheson, P. (2010, April). Methodology for object-oriented system architecture development. In *2010 IEEE International Systems Conference* (pp. 643-646). IEEE.
- [10] Grady, J. O. (2010). *System requirements analysis*. Elsevier.
- [11] Malan, R., & Bredemeyer, D. (2001). Functional requirements and use cases. *Bredemeyer Consulting*.
- [12] Chung, L., Nixon, B. A., Yu, E., & Mylopoulos, J. (2012). *Non-functional requirements in software engineering* (Vol. 5). Springer Science & Business Media.
- [13] Maguire, M., & Bevan, N. (2002, August). User requirements analysis. In *IFIP World Computer Congress, TC 13* (pp. 133-148). Springer, Boston, MA.
- [14] Pitone, D., & Pitman, N. (2005). *UML 2.0 in a Nutshell*. " O'Reilly Media, Inc."
- [15] Wang, C., Duan, W., Ma, J., & Wang, C. (2011, December). The research of Android System architecture and application programming. In *Proceedings of 2011 International Conference on Computer Science and Network Technology* (Vol. 2, pp. 785-790). IEEE.