

The Development of e-Appointment System for UTHM Health Centre using Laravel Framework

Siti NurFalahah Mohd Ridoi¹, Nur Ariffin Mohd Zin^{1*}

¹ *Fakulti Sains Komputer dan Teknologi Maklumat,*

Universiti Tun Hussein Onn Malaysia, Parit Raja, Batu Pahat, 86400, MALAYSIA

*Corresponding Author: ariffin@uthm.edu.my

DOI: <https://doi.org/10.30880/aitcs.2024.05.02.063>

Article Info

Received: 16 July 2024

Accepted: 30 October 2024

Available online: 30 November
2024

Keywords

e-Appointment System for UTHM
Health Centre, Web technology,
Prototyping model, PKU.

Abstract

Pusat Kesihatan Universiti (PKU) at Universiti Tun Hussein Onn Malaysia (UTHM) is the primary medical facility for students and staff, offering comprehensive healthcare services. PKU faces challenges in appointment management, accessibility, and communication. To address these, an e-Appointment System has been developed using PHP (Laravel framework) and MySQL, following a prototyping software development model. The system includes modules for login, profile, appointment booking, automation and notification, booking records, feedback, and an administrator dashboard. User Acceptance Testing shows positive feedback, with over 90% satisfaction among patients and 80% among administrators. The system improves healthcare management efficiency at UTHM, enhancing the well-being and productivity of the university community.

1. Introduction

The World Health Organization (WHO) defines health as a state of complete physical, mental, and social well-being, emphasizing that healthcare encompasses a broad spectrum of services aimed at preserving this well-being [1]. In university settings, such as Universiti Tun Hussein Onn Malaysia (UTHM), healthcare services are essential for supporting the academic and personal goals of students, faculty, and staff [2]. Pusat Kesihatan Universiti (PKU) at UTHM is dedicated to providing comprehensive healthcare services, focusing on modern, quality facilities, disease prevention, and health education [3][4].

Recognizing the dynamic and busy lives of university students, this project proposes the development of an "e-Appointment System for UTHM Health Centre." The envisioned system aims to provide a user-friendly online platform for scheduling medical appointments, addressing several challenges faced by PKU and its community. Traditional appointment management systems at PKU face inefficiencies, accessibility issues, and communication gaps, which the proposed system seeks to overcome [3].

The primary objectives of this project are to design a web-based appointment booking system using an object-oriented approach, develop the system using web technology, and conduct alpha and beta testing to validate its functionality. The system is designed to offer several key features across various modules, each enhancing the overall healthcare experience. These include secure user authentication and profiles, appointment scheduling, real-time availability management, automated notifications, booking records and history, feedback and rating, and a comprehensive administrator dashboard.

By implementing this e-Appointment System, PKU aims to optimize healthcare resource allocation, reduce administrative burdens, and cater to the digital expectations of students. This system represents a transformative tool that will reshape healthcare access and set the stage for future innovations at PKU, enhancing the overall well-being of the university community [2][5][6].

2. Related Work

This section reviews existing concepts and systems relevant to the e-Appointment System for UTHM Health Centre, focusing on healthcare services, online appointment booking systems, and web development frameworks.

2.1 Healthcare Services

Primary health care (PHC) emphasizes universal access to health services, comprising inclusive health services, policies addressing broader health factors, and empowering individuals to manage their health [7]. University healthcare services significantly impact students' well-being and academic success, offering essential services like psychological counseling, healthy lifestyle advice, and regular health checkups [8]. PKU at UTHM is committed to providing these vital services, with the proposed e-Appointment System aiming to optimize healthcare processes and enhance service delivery [2][3].

2.2 Online Appointment Booking System

An online appointment booking system allows patients to schedule medical appointments electronically, reducing workload and preventing duplicate reservations [9]. Such systems provide a convenient way for patients to select doctors, check availability, and book appointments. This project utilizes the Laravel PHP framework, chosen for its robust features, scalability, and developer-friendly nature, to develop the proposed system [10][11].

2.3 Study of existing system

Existing systems like Qmed Asia, BookDoc, and MySejahtera offer valuable insights into effective online appointment booking solutions. Qmed Asia provides comprehensive digital health modules, BookDoc connects patients with healthcare professionals across Asia, and MySejahtera assists the Malaysian government in managing public health [12][13][14]. These systems highlight the importance of user-friendly interfaces, real-time availability, and efficient scheduling, which will be integrated into the e-Appointment System for UTHM Health Centre. The comparison between the existing systems is shown in **Table 1**.

Table 1 System's Comparison

Features/System	QMed Asia	BookDoc Application	MySejahtera Application	e-Appointment System for UTHM Health Centre
Login	√ Phone number/email √ TAC	√ email √ Apple √ Facebook √ Biometric	√ Phone number √ email	√ Matric Number √ Staff ID √ Admin ID
User Authentication	√	√	√	√
Patient Profile	√	√	√	√
Appointment Scheduling	√ Calendar date picker √ Slot availability using color indicator √ Shows unavailable date and time	√ Calendar date picker X Slot availability using color indicator X Shows unavailable date and time	X Calendar date picker √ Slot availability using color indicator X Shows unavailable date and time	√ Calendar date picker √ Slot availability using dropdown menus √ Shows unavailable date and time √ Different services with different time frames
Real-time availability	√	√	√	√
Booking Records and History	√ History (Showing all appointments including the upcoming and past bookings in a single section in Booking Records)	√ Upcoming √ History	√ Upcoming √ Arrived √ Cancelled √ No show	√ Upcoming √ Completed √ Cancelled √ Tab Views

Booking Flexibility	√ Cancel √ Reschedule	√ Cancel √ Reschedule	√ Cancel √ Reschedule	√ Cancel X Reschedule
List of healthcare treatments and services	√ Categorized by type of services √ List	√ Categorized by type of services √ List	√ Categorized by type of services √ List	√ Categorized by type of services √ Dropdown menu
Automation and Notification	√ Email √ Notification	√ Email √ Notification	√ Email √ Notification	√ Email √ Notification
Feedback and Ratings	√ Feedback X Rating	√ Feedback X Rating	√ Feedback X Rating	√ Feedback √ Rating

In summary, the analysis of existing systems like Qmed Asia, BookDoc, and MySejahtera offers critical insights that inform the development of the e-Appointment System for UTHM Health Centre. Each system provides unique features and functionalities, such as secure user authentication, detailed service information, and real-time availability management. By leveraging the strengths and addressing the limitations observed in these systems, the e-Appointment System aims to deliver a user-friendly, efficient, and secure platform tailored to the needs of UTHM's community, ultimately enhancing healthcare service delivery at PKU.

3. Methodology

This section outlines the methodology used for developing the e-Appointment System, focusing on the Prototyping Model.

3.1 Prototyping Model

The Prototyping Model involves creating an early model of the final product to guide development. This iterative approach integrates analysis, design, and implementation phases, allowing for continuous refinement based on user feedback [15]. Given the system's healthcare context and the need for continuous user feedback, the Prototyping Model is an ideal choice. It supports early identification of design issues and ensures the final system meets user needs. The development process includes six phases: Planning, Analysis, Design, Implementation, System Prototype, and Testing, each with specific tasks and outputs [16][15]. **Table 2** depicts the software development activities related to the task during the software development.

Table 2 Software development activities and their task

Phase	Task	Output
Planning	<ul style="list-style-type: none"> Propose the project Develop a Gantt chart 	<ul style="list-style-type: none"> Project proposal Develop Gantt chart
Analysis	<ul style="list-style-type: none"> Analyse functional and non-functional requirements Gather initial user requirements Identify potential challenges and constraints Conduct a feasibility study 	<ul style="list-style-type: none"> Functional and non-functional requirements User requirement analysis Software and hardware requirements UML diagrams Requirement traceability matrix
Design	<ul style="list-style-type: none"> Design the logical structure of the proposed system Develop a database schema for the system 	<ul style="list-style-type: none"> Architecture design Database design Interface design
Implementation	<ul style="list-style-type: none"> Develop the proposed system Test the proposed system 	<ul style="list-style-type: none"> Program code Test Cases
System Prototype	<ul style="list-style-type: none"> Create a working prototype of the system that includes core functionalities Gather feedback from stakeholders on the prototype Make changes to the system based on user feedback 	<ul style="list-style-type: none"> A working prototype of the system A list of feedback from stakeholders A revised system prototype that incorporates user feedback
Testing	<ul style="list-style-type: none"> Conduct system testing to identify and address any issues Finalize the system prototype 	<ul style="list-style-type: none"> System Testing Report Final System Prototype

3.2 Analysis

The System Requirement Analysis phase in software development is crucial for identifying, documenting, and analyzing the needs of users and stakeholders. The goal is to define what the system should achieve and set the foundation for subsequent development phases. The e-Appointment System for UTHM Health Centre includes functional modules, each catering to specific user categories, such as students, staff, PKU's staff, and administrators. The core features encompass secure login, profile management, appointment booking, automation and notifications, booking records, feedback, and an administrator dashboard.

Functional requirements, as outlined in **Table 3**, include secure login procedures, profile management, user-friendly appointment booking, automation of notifications, comprehensive booking records, user feedback, and an efficient administrator dashboard. Specific functionalities are detailed for each module.

Table 3 Functional requirements

No.	Module	Functional requirements
1.	Login	<ul style="list-style-type: none"> Allow users to log in using valid credentials. Provide password recovery options.
2.	Profile	<ul style="list-style-type: none"> Allow users to view and edit profiles. Store and manage user profiles securely.
3.	Appointment Booking	<ul style="list-style-type: none"> Display user-friendly interface for scheduling appointments. Provide calendar view.
4.	Automation & Notification	<ul style="list-style-type: none"> Automate email notifications for booking confirmation. Include essential details.
5.	Booking Records and History	<ul style="list-style-type: none"> Allow users to view complete booking history. Categorize and display status.
6.	Feedback & Ratings	<ul style="list-style-type: none"> Enable users to provide feedback and ratings.
7.	Administrator Dashboard	<ul style="list-style-type: none"> Efficiently manage appointments, users, medical staff, opening and closing hours of medical services, time slots and announcements.

Table 4 outlines non-functional requirements, emphasizing security, usability, performance, reliability, and scalability.

Table 4 Non-functional requirements

No.	Module	Non-functional requirements
1.	Security	<ul style="list-style-type: none"> Force users to log in before booking. Implement robust encryption methods.
2.	Usability	<ul style="list-style-type: none"> Ensure an intuitive user interface. Provide accessibility features.
3.	Performance	Maintain low latency for quick response times.
4.	Reliability	Ensure high system reliability with minimal downtime.
5.	Scalability	Design a scalable system architecture for an increasing user base.

The e-Appointment System for UTHM Health Centre is a sophisticated solution streamlining healthcare appointments. It features secure login, profile management, user-friendly appointment booking, automated notifications, and a comprehensive booking history. The Administrator Dashboard enhances administrative efficiency. Non-functional aspects prioritize security, usability, performance, reliability, and scalability for a transformative healthcare booking platform.

The To-Be Model (**Appendix A**) for the e-Appointment System for UTHM Health Centre sets the stage for a transformative healthcare booking platform, embodying cutting-edge functionalities and user-centric design.

Figure 1 presents the use case diagram for UTHM student and staff as patients, and PKU's staff as administrators. The visual representation encapsulates the system's activities, depicting the comprehensive functionalities and user interactions within the e-Appointment System for UTHM Health Centre.

There are four use cases for patients including login, view profile, book appointment that includes automation and notification, and view appointment booking history that includes feedback and ratings. As for administrators, there are seven use cases involve login, manage user accounts, manage appointments that includes view feedback and rating, manage users, manage medical staff, manage medical services, manage time slot, and manage announcement.

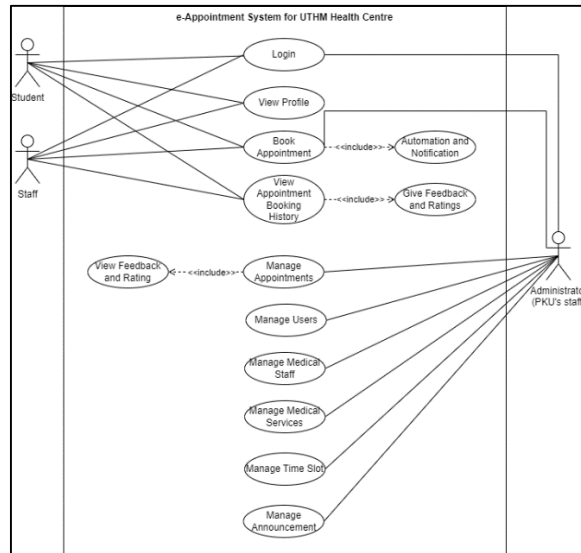


Figure 1: Use Case Diagram for e-Appointment System for UTHM Health Centre

Next, the static structure of the proposed e-Appointment System is illustrated through a class diagram in **Figure 2**, featuring 11 key classes: User, Administrator, Appointment, FeedbackRating, MedicalStaff, Service, TimeSlot, Department, Unit, Announcement, and Notification. The User class includes roles for student, staff, and administrator, connecting with Appointment and FeedbackRating for login, profile management, booking, and feedback. The Administrator class manages system operations, including appointments and announcements. The Appointment class handles booking and feedback, interacting with TimeSlots, MedicalStaff, and Service classes. FeedbackRating captures patient evaluations, while MedicalStaff and Service manage healthcare details. The TimeSlot class ensures scheduling efficiency, and the Announcement class facilitates communication. This diagram supports project phases and effective healthcare service delivery.

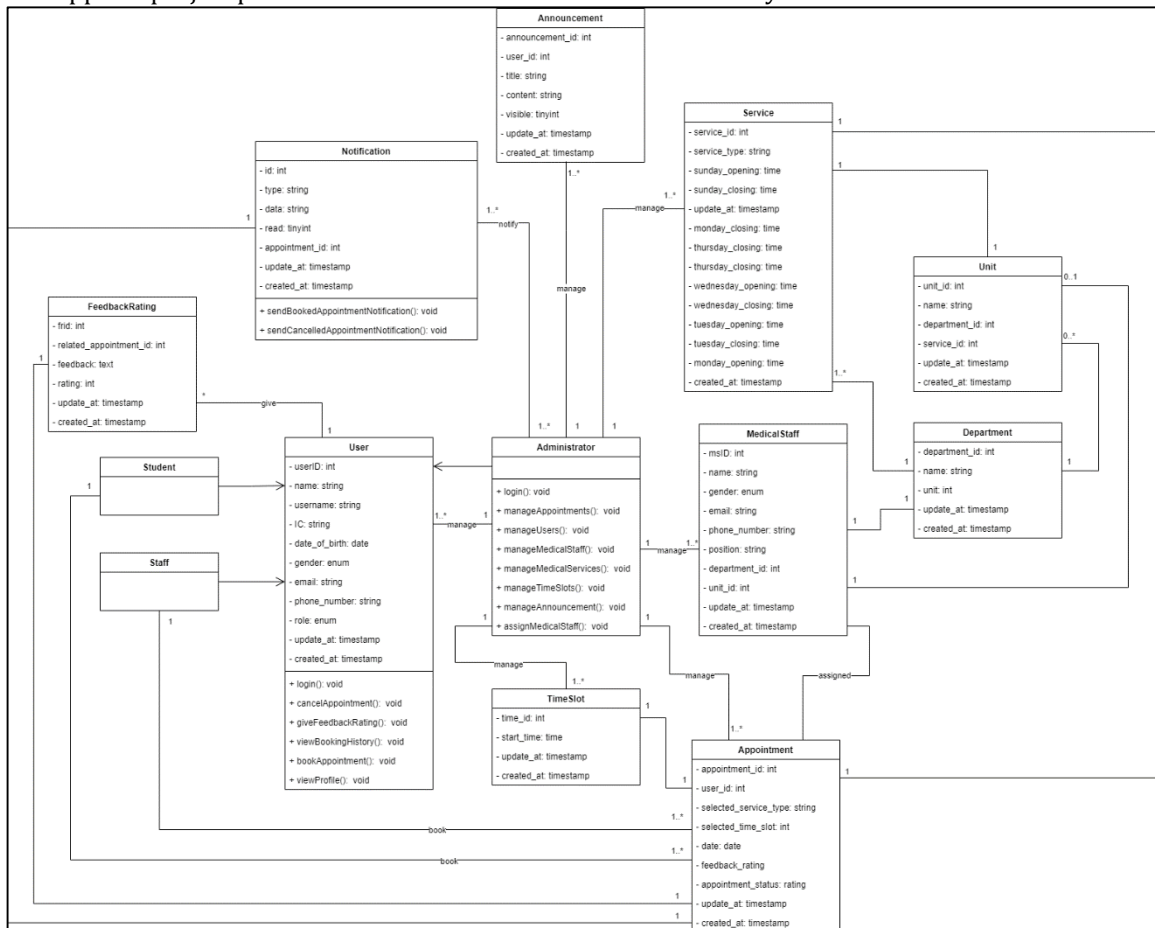


Figure 2: Class Diagram for e-Appointment System for UTHM Health Centre

3.3 Design

The schema table, as depicted in the class diagram (refer **Figure 2**), encapsulates essential attributes and relationships that contribute to the overall functionality and coherence of the system's data architecture. The database schema extracted from the Class Diagram is listed below:

- i. users (userID, name, username, password, IC, date_of_birth, gender, email, phone_number, role, remember_token, created_at, updated_at)
- ii. appointments (appointment_id, user_id, msID, selected_service_type, selected_time_slot, date, feedback_rating, appointment_status, created_at, updated_at)
- iii. feedback_rating (fr_id, related_appointment_id, feedback, rating, created_at, updated_at)
- iv. departments (department_id, name, unit, created_at, updated_at)
- v. services (service_id, service_type, department_id, sunday_opening, sunday_closing, monday_opening, monday_closing, tuesday_opening, tuesday_closing, wednesday_opening, wednesday_closing, thursday_opening, thursday_closing, created_at, updated_at)
- vi. units (unit_id, name, department_id, service_id, created_at, updated_at)
- vii. time_slots (time_id, start_time, created_at, updated_at)
- viii. medical_staff (msID, name, gender, email, phone_number, position, department_id, unit_id, created_at, updated_at)
- ix. announcements (announcement_id, user_id, title, content, visible, created_at, updated_at)
- x. notifications (id, type, data, read, appointment_id, created_at, updated_at)

The e-Appointment System at UTHM Health Centre emphasizes user-friendly design to enhance functionality and aesthetics. The prototype focuses on creating intuitive and accessible interfaces for patients, administrators, and medical staff. By prioritizing simplicity and clarity, the system ensures easy navigation across modules like appointment booking, profile management, and feedback submission. Responsive design elements provide a seamless experience on various devices. Usability testing and user feedback are integral to the design process, ensuring the final product meets user expectations and improves healthcare service efficiency at UTHM Health Centre. The prototype processes are shown in **Figure 3(a)** and **Figure 3(b)**.

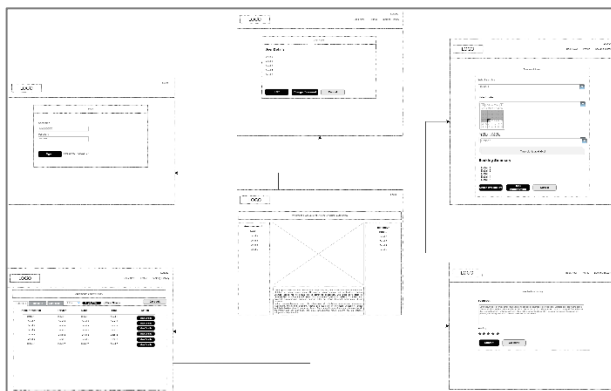


Figure 3(a): Prototype Design (Patients)

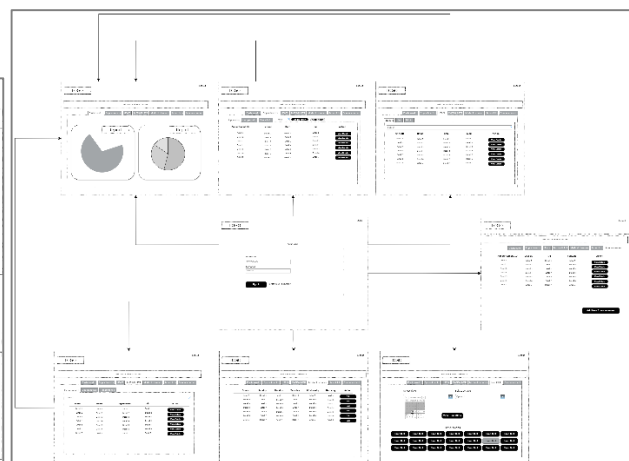


Figure 3(b): Prototype Design (Administrators)

4. Result and Discussion

This section mainly explains about user interface and code segment of the e-Appointment System for UTHM Health Centre. Along in this section, is the result of test cases as well.

4.1 Implementation

The e-Appointment System features a user-friendly interface designed to ensure ease of use and accessibility, allowing students and staff to effortlessly navigate and manage their healthcare appointments. The system was implemented using PHP and MySQL, with Visual Studio Code for coding. It includes modules for login, profile management, appointment booking, notifications, booking history, feedback, and an administrator dashboard. Laravel's authentication system handles login and password recovery. The profile module allows secure data management, while the booking module provides an intuitive interface for scheduling. Notifications are automated for confirmations and updates. The history module tracks appointments, and the feedback module collects user experiences for review. The administrator dashboard efficiently manages appointments, user accounts, and feedback.

In Login Module, users will be able to login and access their accounts by entering their username and password. The interfaces for login page are shown in **Figure 4(a)** for patients, and **Figure 4(b)** for administrators. Through login, patients will be able to book and cancel appointments, manage profile, and view their booking history. For administrators, they will be able to access the administrator's dashboard and manage the system. **Figure 4(c)** shows the code segment for patients to log in to the system while **Figure 4(d)** shows the code segment for administrators' login page.

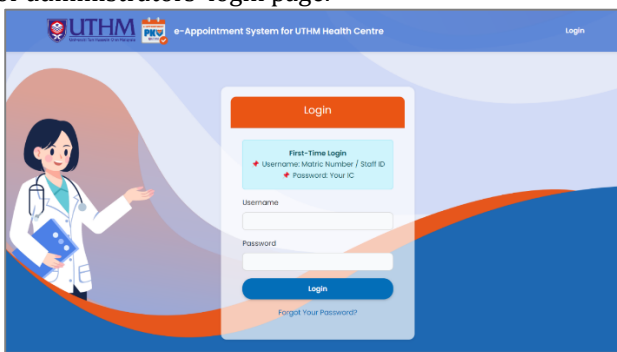


Figure 4(a): Patient login interface

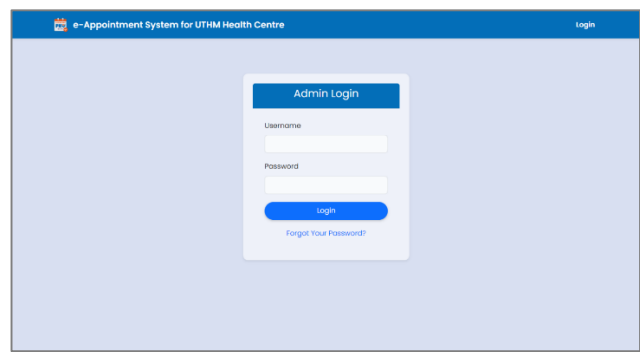


Figure 4(b): Administrator login interface

```

52 public function login(Request $request)
53 {
54     $credentials = $request->only('username', 'password');
55
56     if (Auth::attempt($credentials)) {
57         $user = Auth::user();
58
59         if ($user->role == 'student' || $user->role == 'staff') {
60             return redirect()->intended('/home');
61         } else {
62             Auth::logout();
63             return redirect()->back()->with('status', 'Invalid username or password');
64         }
65     }
66
67     return redirect()->back()->with('status', 'Invalid username or password');
68 }
    
```

Figure 4(c): Code Segment for Login – Patients

```

103 public function adminLogin(Request $request)
104 {
105
106     $credentials = $request->only('username', 'password');
107     $credentials['role'] = 'admin';
108
109     if (Auth::attempt($credentials)) {
110         return redirect()->route('admin.dashboard');
111     }
112
113     return redirect()->back()->with('status', 'Invalid admin credentials');
114 }
    
```

Figure 4(d): Code Segment for Login – Administrators

Users can manage their profile by viewing, editing their details and change their passwords. **Figure 5(a)** and **Figure 5(b)** display the profile interfaces for patients and administrators respectively, while **Figure 5(c)** shows the code segment for displaying the profile page.

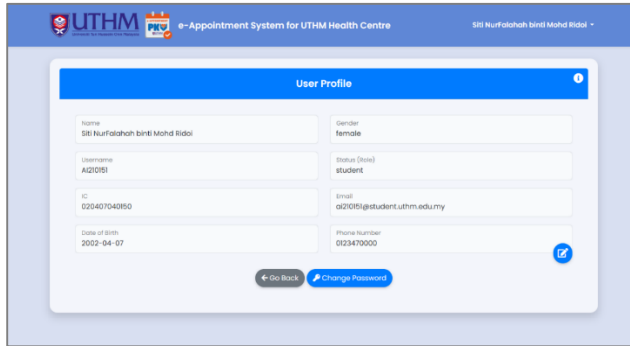


Figure 5(a): Patient profile interface

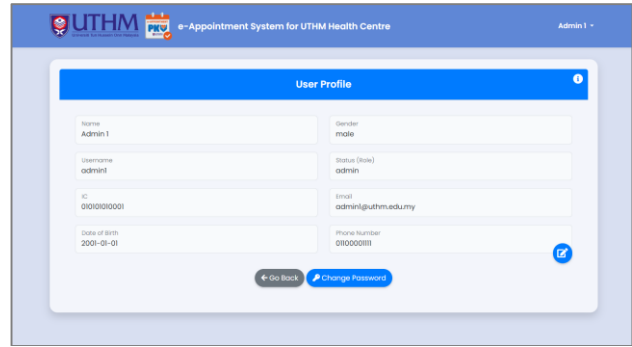


Figure 5(b): Administrator profile interface

```
public function showProfile()
{
    $user = auth()->user();
    $editable = true;
    return view('profile', compact('user', 'editable'));
}
```

Figure 5(c): Code segment for Profile

Patients will be able to book appointments at PKU by selecting service, date, and time slots, then check the availability of the selected time slot, as can be seen in **Figure 6(a)**. **Figure 6(b)** shows the code segment for appointment booking.

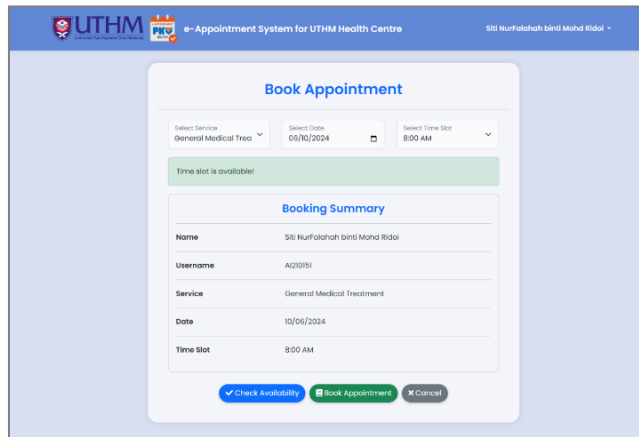


Figure 6(a): Appointment booking interface

```
public function create()
{
    $services = Service::all();
    $disableDate = now()->hour >= 19;
    $bookedTimeSlots = Appointment::where('date', now()->toDateString()->pluck('selected_time_slot');
    $cancelledTimeSlots = [];
    $serviceTimeRestrictions = [];
    foreach ($services as $service) {
        $serviceTimeRestrictions[$service->service_type] = [
            '0' => $service->sunday_closing,
            '1' => $service->monday_closing,
            '2' => $service->tuesday_closing,
            '3' => $service->>wednesday_closing,
            '4' => $service->thursday_closing,
        ];
    }
    $timeSlots = TimeSlot::all();
    return view('appointments.create', compact('services', 'timeSlots', 'disableDate', 'bookedTimeSlots', 'cancelledTimeSlots', 'serviceTimeRestrictions'));
}
```

Figure 6(b): Code segment for Appointment Booking

As for the Booking Records and History, appointments will be sorted into tab view which categorizes into Upcoming, Completed and Cancelled, as shown in **Figure 7(a)**, while **Figure 7(b)** displays its code segment.

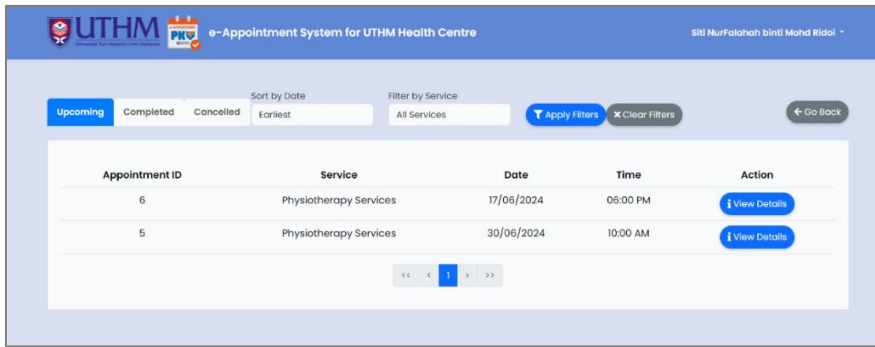


Figure 7(a): Booking Records and History interface

```

public function bookingHistory(Request $request)
{
    if (!auth()->check()) {
        return redirect()->route('login')->with('error', 'Please log in to view booking history.');
```

```

    }

    $user = auth()->user();
    $currentDate = now()->toDateString();
    $currentTime = now()->toTimeString();

    $upcomingAppointmentsQuery = Appointment::where('user_id', $user->userID)
        ->where('appointment_status', '!=', 'cancelled')
        ->where(function($query) use ($currentDate, $currentTime) {
            $query->where('date', '>', $currentDate)
            ->where(function($query) use ($currentDate, $currentTime) {
                $query->where('date', '=', $currentDate)
                ->whereHas('timeSlot', function($query) use ($currentTime) {
                    $query->where('start_time', '>', $currentTime);
                });
            });
        });

    $upcomingAppointments = $upcomingAppointmentsQuery->paginate(10, ['*', 'upcoming']);

    $completedAppointmentsQuery = Appointment::where('user_id', $user->userID)
        ->where(function($query) use ($currentDate, $currentTime) {
            $query->where('appointment_status', 'completed')
            ->where(function($query) use ($currentDate, $currentTime) {
                $query->where('date', '<', $currentDate)
                ->where(function($query) use ($currentDate, $currentTime) {
                    $query->where('date', '=', $currentDate)
                    ->whereHas('timeSlot', function($query) use ($currentTime) {
                        $query->where('start_time', '<=', $currentTime);
                    });
                });
            });
        });

    $completedAppointments = $completedAppointmentsQuery->paginate(10, ['*', 'completed']);

    $cancelledAppointmentsQuery = Appointment::where('user_id', $user->userID)
        ->where('appointment_status', 'cancelled');

    $cancelledAppointments = $cancelledAppointmentsQuery->paginate(10, ['*', 'cancelled']);

    $services = Service::all();

    return view('appointments.booking-history', compact('upcomingAppointments', 'completedAppointments', 'cancelledAppointments', 'services', 'filterDate', 'filterService'));
}
    
```

Figure 7(b): Code segment for Booking Records and History

Patients are also able to give their feedback and rating after completing an appointment. This ensures the satisfactory of the patients in getting healthcare services at PKU can be monitored. Figure 8(a) shows the feedback and rating form, along with its code segment in Figure 8(b).

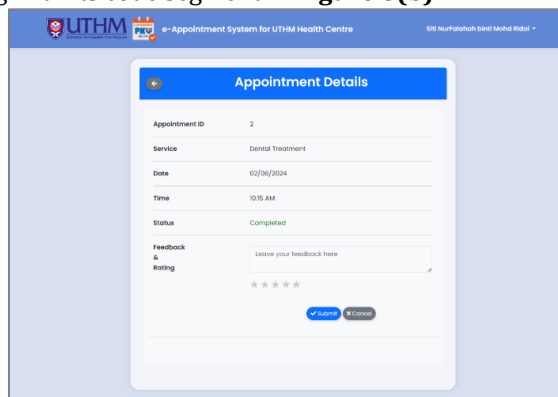


Figure 8(a): Feedback and rating interface

```

public function store(Request $request)
{
    Log::info('Store method called');
    Log::info('Request Data: ', $request->all());

    $request->validate([
        'appointment_id' => 'required|exists:appointments,appointment_id',
        'feedback' => 'nullable|string',
        'rating' => 'nullable|integer|min:1|max:5',
    ]);

    try {
        $appointmentId = $request->input('appointment_id');
        $feedbackRating = FeedbackRating::updateOrCreate(
            [
                'related_appointment_id' => $appointmentId,
            ],
            [
                'feedback' => $request->input('feedback'),
                'rating' => $request->input('rating')
            ]
        );

        // Update the feedback_rating column in the appointments table
        $appointment = Appointment::find($appointmentId);
        $appointment->feedback_rating = $feedbackRating->fr_id;
        $appointment->save();

        Log::info('Feedback and rating submitted successfully');
        Log::info('Updated feedback_rating column in appointments table with fr_id: ' . $feedbackRating->fr_id);

        return redirect()->route('booking-details', ['appointmentId' => $appointmentId])->with('message', 'Feedback and rating submitted successfully!');
    } catch (\Exception $e) {
        Log::error('Failed to submit feedback and rating: ' . $e->getMessage());
        return redirect()->route('booking-details', ['appointmentId' => $request->input('appointment_id')])->with('error', 'Failed to submit feedback and rating. Please try again.');
```

Figure 8(b) Code segment for Feedback and Rating

As for administrators, module Dashboard allows them to manage the appointment bookings and related services within the system. The interface for dashboard is shown in Figure 9(a) along with its code segment in Figure 9(b). The dashboard display the statistical report of appointments by services and over time, to provide informational insights for administrators of the appointment trend in PKU.

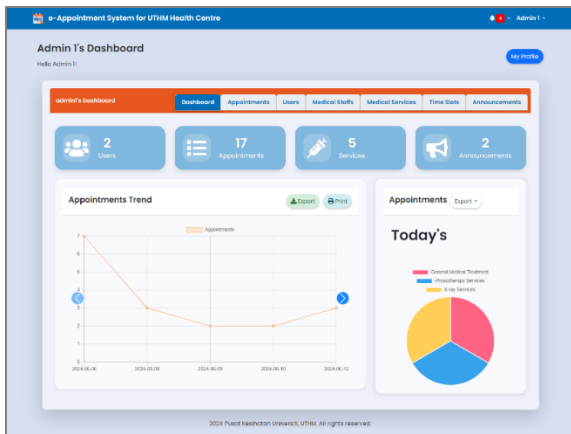


Figure 9(a): Dashboard interface

```

public function index()
{
    $totalUsers = User::count();
    $totalAppointments = Appointment::count();
    $totalServices = Service::count();
    $totalAnnouncements = Announcement::count();

    $appointmentsByService = Appointment::select('selected_service_type as service', DB::raw('count(*) as count'))
        ->groupBy('selected_service_type')
        ->get();

    $appointmentsByStatus = Appointment::select('appointment_status as status', DB::raw('count(*) as count'))
        ->groupBy('appointment_status')
        ->get();

    $appointmentsByMonth = Appointment::select(DB::raw('DATE_FORMAT(created_at, "MY-%m") as month'), DB::raw('count(*) as count'))
        ->groupBy(DB::raw('DATE_FORMAT(created_at, "MY-%m")'))
        ->get();

    $servicesByAppointment = Service::withCount('appointments')
        ->orderBy('appointments_count', 'desc')
        ->get();

    $appointmentsTrend = Appointment::select(DB::raw('DATE(created_at) as date'), DB::raw('count(*) as count'))
        ->groupBy(DB::raw('DATE(created_at)'))
        ->orderBy('date', 'asc')
        ->get();

    $appointmentsByServiceToday = Appointment::whereDate('created_at', Carbon::today())
        ->select('selected_service_type', DB::raw('count(*) as count'))
        ->groupBy('selected_service_type')
        ->pluck('count', 'selected_service_type');

    return view('admin.dashboard', compact(
        'totalUsers',
        'totalAppointments',
        'totalServices',
        'totalAnnouncements',
        'appointmentsByService',
        'appointmentsByStatus',
        'appointmentsByMonth',
        'servicesByAppointment',
        'appointmentsTrend',
        'appointmentsByServiceToday'
    ));
}
```

Figure 9(b): Code segment for Dashboard

Administrators can manage appointments, users, medical staff, medical services, time slots and announcements within the system. Each of this management is sorted into tab views for better visibility and navigability throughout the system. Figure 10(a) and Figure 10(b) display the interface and the code segment of manage appointments respectively, that shows the appointments are categorized into Upcoming, Completed and Cancelled tabs. Administrator can assign the medical staff to each appointment in this module.

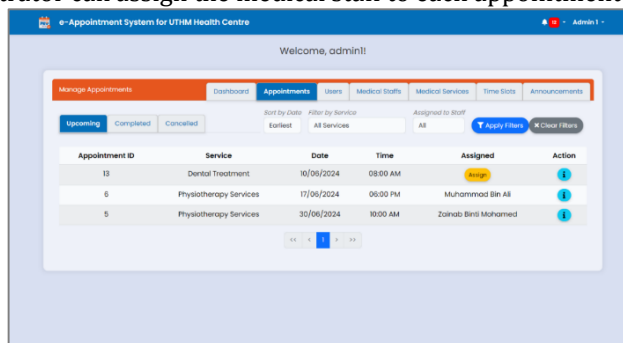


Figure 10(a): Manage appointment interface

```

public function manageAppointments(Request $request)
{
    if (!auth()->check()) {
        return redirect()->route('login')->with('error', 'Please log in to manage appointments.');
```

```

Appointment::where('appointment_status', '!=', 'cancelled')
->where('date', '<', now()->toDateString())
->orWhere(function($query) {
    $query->where('date', '=', now()->toDateString())
        ->whereHas('timeSlot', function($query) {
            $query->where('start_time', '<', now()->toTimeString());
        });
});
->update(['appointment_status' => 'completed']);

$upcomingAppointments = $upcomingAppointmentsQuery->paginate(10, ['*'], 'upcoming');
$completedAppointments = $completedAppointmentsQuery->paginate(10, ['*'], 'completed');
$cancelledAppointments = $cancelledAppointmentsQuery->paginate(10, ['*'], 'cancelled');

$services = Service::all();

return view('admin.manageAppointments', compact('upcomingAppointments', 'completedAppointments', 'cancelledAppointments', 'filterDate', 'filterService', 'filterAssigned', 'services'));
    
```

Figure 10(b): Code segment for manage appointments

Followed by manage users module, administrators can manage the users of the system by viewing, editing, and deleting users, along with adding a new user. Users are sorted into tab views of roles which are admin, student and staff. Figure 11(a) and Figure 11(b) shows the interface and the code segment of this module respectively.

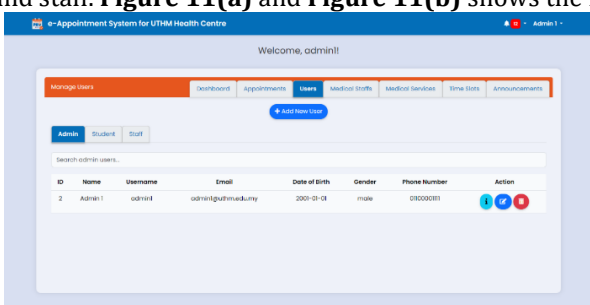


Figure 11(a): Manage users interface

```

public function manageUsers()
{
    $adminUsers = User::where('role', 'admin')->get();
    $studentUsers = User::where('role', 'student')->get();
    $staffUsers = User::where('role', 'staff')->get();

    $tab = "admin";
    return view('admin.manageUsers', compact('adminUsers', 'studentUsers', 'staffUsers', 'tab'));
}
    
```

Figure 11(b): Code segment for manage users

Then, manage medical staff module allows administrators to manage the medical staff within PKU. The medical staff are sorted into tab views based on their departments. Administrators can view, edit, and delete their details, along with adding a new medical staff. The interface is shown in Figure 12(a) while the code segment for this module is shown in Figure 12(b).

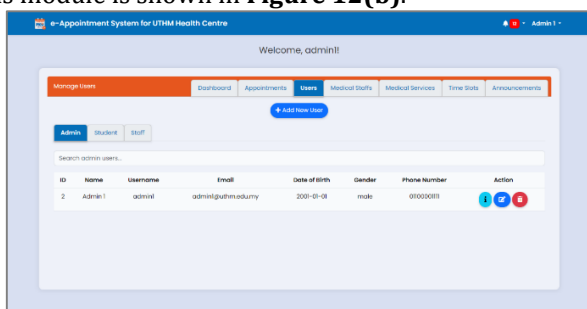


Figure 12(a): Manage medical staff interface

```

public function manageMedStaff()
{
    $departments = Department::with('medicalStaff')->get();

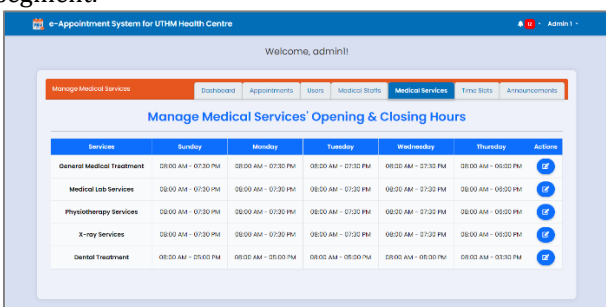
    return view('admin.manageMedStaff', compact('departments'));
}

2 references | 0 overrides
public function viewMedStaffDetails($medStaff)
{
    $medStaff = MedicalStaff::findOrFail($medStaff);

    return view('admin.medStaffDetails', compact('medStaff'));
}
    
```

Figure 12(b): Code segment for manage medical staff

Manage medical services module allows administrators to manage the opening and closing hours of the medical services within PKU. Shown in Figure 13(a) is the interface for this module and Figure 13(b) is the code segment.



```

public function manageMedServices()
{
    $services = Service::all();

    return view('admin.manageMedServices', compact('services'));
}
    
```

Figure 13(a): Manage medical services interface

Figure 13(b): Code segment for manage medical services

Next, is manage time slot module that allows administrators to view the booked and available time slots, by selecting a date and a service. The system will display list of all the time slots of the selected day and service, with colour indicators of red for the booked time slot and green for the available ones. This module's interface is shown in **Figure 14(a)** while its code segment is shown in **Figure 14(b)**.

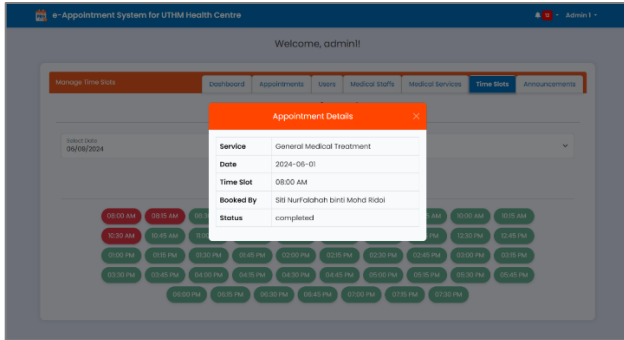


Figure 14(a): Manage time slot interface

```
public function manageTimeSlots(Request $request)
{
    $services = Service::all();
    $timeSlots = [];
    $bookedTimeSlots = [];

    if ($request->has('date') && $request->has('service')) {
        $selectedDate = $request->input('date');
        $selectedService = $request->input('service');

        $bookedTimeSlots = Appointment::where('date', $selectedDate)
            ->where('selected_service_type', $selectedService)
            ->pluck('selected_time_slot')
            ->toArray();

        $timeSlots = TimeSlot::all();
    }

    return view('admin.manageTimeSlots', compact('services', 'timeSlots', 'bookedTimeSlots'));
}
```

Figure 14(b): code segment for Manage time slot

Lastly, is manage announcement module. This module allows administrators to create, edit and delete announcements, along with setting the visibility of each announcement, that will be displayed on the home page. The interface for this module is shown in **Figure 15(a)** and the code segment in **Figure 15(b)**.

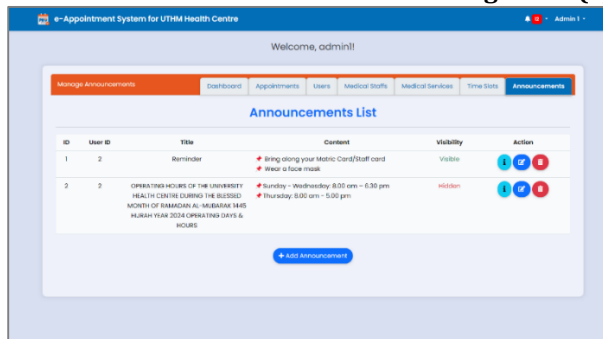


Figure 15(a): Manage announcement interface

```
public function manageAnnouncements()
{
    $announcements = Announcement::all();

    return view('admin.manageAnnouncements', ['announcements' => $announcements]);
}
```

Figure 15(b): code segment for Manage announcement

4.2 Testing Phase

Testing is essential for ensuring the stability and reliability of the e-Appointment System for UTHM Health Centre. Functional testing focuses on validating the functional requirements of a system to ensure its proper operation. The test case results for all modules in are detailed in **Table 5**. The test cases were executed successfully, with the actual results matching the expected outcome, indicating that the system functioned as intended during testing.

Table 5 Test Cases

Test Case ID	SRS ID	Description	STATUS
TC_100	REQ_100	Login	Fail/Pass
TEST_100_001	FR01-01, FR01-03, CR01-01, CR01-02	Patient log in to the website by entering username and password.	PASS
TEST_100_002	FR01-02	Patient log in to the website without entering the username or password.	PASS
TEST_100_003	FR01-02, CR01-01, CR01-02	Patient log in to the website by entering the wrong username or password.	PASS
TEST_100_004	FR01-01, FR01-03, CR01-02	Administrator log in to the website by entering username and password.	PASS
TEST_100_005	FR01-02	Administrator log in to the website without entering the username or password.	PASS
TEST_100_006	FR01-02, CR01-01, CR01-02	Administrator log in to the website by entering the wrong username or password.	PASS
Test Case ID	SRS ID	Description	STATUS
TC_200	REQ_200	Profile	Fail/Pass
TEST_200_001	FR02-01	User clicks 'View Profile' button.	PASS
TEST_200_002	FR02-03	User clicks edit icon on phone number details.	PASS
TEST_200_003	FR02-03, FR02-04, CR02-01	User edits the phone number and clicks save icon.	PASS
TEST_200_004	FR02-02	User clicks 'Change Password' button.	PASS
TEST_200_005	FR02-02, CR02-01	User enters a wrong current password.	PASS
TEST_200_006	FR02-02, CR02-01	User enters password that is less than 8 characters.	PASS
TEST_200_007	CR02-01	User enters the password that do not match for the confirmation password.	PASS
TEST_200_008	CR02-01	User enters same password for the new password.	PASS
TEST_200_009	FR02-02, FR02-04	User enters correct current password, password that is 8 characters and above and enter correct password for confirmation password.	PASS
TC_300	REQ_300	Appointment Booking	Fail/Pass
TEST_300_001	FR03-01, FR03-02	User clicks 'Book Appointment' button.	PASS
TEST_300_002	FR03-01, FR03-11, FR03-12, FR03-14, CR03-01, CR03-02	User selects a Friday or Saturday for the date.	PASS
TEST_300_003	FR03-01, FR03-14, CR03-01, CR03-02	User clicks 'Check Availability' button without selecting a service or a date or a time slot.	PASS
TEST_300_004	FR03-01, FR03-02, FR03-03, FR03-04, FR03-05, FR03-06, FR03-07, FR03-13, FR03-14, CR03-01, CR03-02	User clicks 'Check Availability' button after selecting a service, a date and a time slot.	PASS
TEST_300_005	FR03-01, FR03-02, FR03-08, FR03-09, FR03-10, CR04-01	User clicks 'Book Appointment' button.	PASS
TC_400	REQ_400	Booking Records and History	Fail/Pass
TEST_400_001	FR04-01, FR04-02, FR04-03, FR04-10, CR04-01	User clicks 'Booking History' button.	PASS
TEST_400_002	FR04-02, FR04-04, FR04-10, CR04-01	User clicks the Upcoming tab.	PASS
TEST_400_003	FR04-02, FR04-05, FR04-10, CR04-01	User clicks the Completed tab.	PASS
TEST_400_004	FR04-02, FR04-06, FR04-10, CR04-01	User clicks the Cancelled tab.	PASS
TEST_400_005	FR04-10, CR04-01	User selects the filters and apply the filters.	PASS

TEST_400_006	FR04-10, CR04-01	User clicks the 'Clear Filters' button.	PASS
TEST_400_007	FR04-04, FR04-05, FR04-06, FR04-07, FR04-11	User clicks the 'View Details' button.	PASS
TEST_400_008	FR04-07, FR04-08, FR04-09	User clicks the 'Cancel Appointment' button of the upcoming appointment on the upcoming appointment details page.	PASS
TC_500	REQ_500	Feedback and Rating	Fail/Pass
TEST_500_001	FR04-11	The system clicks 'Give Feedback and Rating' button on the completed appointment details page.	PASS
TEST_500_002	FR04-11	The user clicks the 'Submit' button without entering the feedback or select a rating.	PASS
TEST_500_003	FR04-11	User clicks the 'Submit' button after entering the feedback and selecting a rating.	PASS
TC_600	REQ_600	Manage Appointment	Fail/Pass
TEST_600_001	FR05-01	Administrator clicks the Appointments tab.	PASS
TEST_600_002	FR05-01	Administrator clicks the Upcoming tab.	PASS
TEST_600_003	FR05-01, CR05-01	Administrator clicks the Completed tab.	PASS
Test Case ID	SRS ID	Description	STATUS
TEST_600_004	FR05-01, CR05-01	Administrator clicks the Cancelled tab.	PASS
TEST_600_005	FR05-01	Administrator selects the filters and apply the filters.	PASS
TEST_600_006	FR05-01	Administrator clicks the 'Clear Filters' button.	PASS
TEST_600_007	FR05-01	Administrator clicks the 'View Details' button.	PASS
TEST_600_008	FR05-02, FR05-03, CR05-01, CR05-02	Administrator clicks the 'Assign' button on the upcoming appointment details page.	PASS
TC_700	REQ_700	Manage Users	Fail/Pass
TEST_700_001	FR07-01	Administrator clicks the Users tab.	PASS
TEST_700_002	FR07-01	Administrator clicks any of the user role tab.	PASS
TEST_700_003	FR07-02,	Administrator searches for a user that is under the correct user tab.	PASS
TEST_700_004	FR07-02	Administrator searches for a user that is not under the correct user tab.	PASS
TEST_700_005	CR07-02	Administrator clicks info icon.	PASS
TEST_700_006	FR07-03, CR07-02	Administrator clicks edit icon.	PASS
TEST_700_007	CR07-01, CR07-02	Administrator clicks delete icon.	PASS
TEST_700_008	FR07-04, CR07-02	Administrators click add new staff button.	PASS
TC_800	REQ_800	Manage Medical Staff	Fail/Pass
TEST_800_001	FR08-01	Administrator clicks the Medical Staff tab.	PASS
TEST_800_002	FR08-01	Administrator clicks any of the department tab.	PASS
TEST_800_003	CR08-01	Administrator searches for a medical staff that is under the correct department tab.	PASS
TEST_800_004	CR08-01	Administrator searches for a medical staff that is not under the correct department tab.	PASS
TEST_800_005	FR08-06	Administrator clicks info icon.	PASS
TEST_800_006	FR08-03	Administrator clicks edit icon.	PASS
TEST_800_007	FR08-05	Administrator clicks delete icon.	PASS
TEST_800_008	FR08-04	Administrator clicks add new medical staff button.	PASS
TEST_800_009	FR08-02, CR08-02	Administrator updates the status of the medical staff.	PASS
TC_900	REQ_900	Manage Medical Services	Fail/Pass
TEST_900_001	FR09-01, FR09-02	Administrator clicks the Medical Services tab.	PASS
TEST_900_002	FR09-02	Administrator clicks edit icon.	PASS
TEST_900_003	CR09-01	Administrator clicks save icon.	PASS

TC_1000	REQ_1000	Manage Time Slot	Fail/Pass
TEST_1000_001	FR10-01	Administrator clicks the Time Slots tab.	PASS
TEST_1000_002	FR10-01, CR10-01	Administrator clicks 'View Time Slots' button without selecting a date or a service.	PASS
TEST_1000_003	FR10-01, FR10-02, CR10-01	Administrator clicks 'View Time Slots' button after selecting a date or a service.	PASS
TEST_1000_004	FR10-02, FR10-03	Administrator clicks a booked time slot.	PASS
TC_1100	REQ_1100	Manage Announcement	Fail/Pass
TEST_1100_001	FR11-01, FR11-02, FR11-03	Administrator clicks the Announcements tab.	PASS
TEST_1100_002	FR11-03	Administrator clicks info icon.	PASS
TEST_1100_003	FR11-02, CR11-01	Administrator clicks add new announcement button.	PASS
TEST_1100_004	FR11-03, CR11-01, CR12-02	Administrator clicks edit icon.	PASS
TEST_1100_005	FR11-04	Administrator clicks delete icon.	PASS
TEST_1100_006	FR11-05, CR11-01, CR11-02	Administrator sets the announcement visibility to 'Show'.	PASS
TEST_1100_007	FR11-04, CR11-01, CR11-02	Administrator sets the announcement visibility to 'Hide'.	PASS

A total of 70 test cases had been conducted to test e-Appointment System for UTHM Health Centre. The system has passed successfully, as shown in **Table 6**, of the overall result of the test cases.

Table 6 Overall Result of Test Cases

Test Case ID	Total Test Cases	Total Success	Total Fail
TC_100	6	6	0
TC_200	9	9	0
TC_300	5	5	0
TC_400	8	8	0
TC_500	3	3	0
TC_600	8	8	0
TC_700	8	8	0
TC_800	9	9	0
TC_900	3	3	0
TC_1000	4	4	0
TC_1100	7	7	0
Total	70	70	0

The user acceptance testing (UAT) for the e-Appointment System at UTHM Health Centre was conducted to evaluate the system's usability, functionality, and overall effectiveness. Feedback was collected from both patients and administrators, providing valuable insights into the system's performance and highlighting areas for improvement. The UAT is delivered via a questionnaire form that is created through Google Forms. The Technology Acceptance Model (TAM) is used, with questionnaires designed around four constructs: Perceived Ease of Use (PEOU), Perceived Usefulness (PU), Attribute of Usability (AU), and User Satisfaction (US).

The user acceptance test for the e-Appointment System at UTHM Health Centre yielded highly positive results from both patients and administrators. Patients rated the system 93% "very good" and 7% "good," praising its user-friendly interface, ease of logging in, profile management, and booking processes, with 14 out of 15 recommending it, shown in **Table 7**. Administrators also expressed high satisfaction, with 71% rating it "good" and 29% "very good," appreciating the ease of managing appointments, user accounts, and time slots, along with a user-friendly dashboard, shown in **Table 8**. Both groups suggested future improvements like a mobile application and better calendar integration. Overall, the system is effective, reliable, and well-received proven by the positive feedback from patients and administrators.

Table 7 User Acceptance Testing – Patients

Question	1	2	3	4
Perceived Ease of Use (PEOU) Questions				
How would you rate the ease of logging into the system?			1	14
How would you rate the ease of viewing and editing your profile details?			1	14
How would you rate the ease of booking an appointment with this website?			1	14
How would you rate the simplicity of submitting feedback and ratings for a completed appointment?			2	13
Perceived of Usefulness (PU) Questions				
How would you rate the usefulness of having your profile information easily accessible and editable?			2	13
How would you rate the usefulness of the system in providing clear availability of appointment slots?			1	14
How would you rate the usefulness of the booking history feature in helping you keep track of your appointments?			1	14
How would you rate the usefulness of confirmation emails for upcoming appointments?			1	14
Attribute of Usability (AU)				
How would you rate the clarity and helpfulness of error messages (e.g., for incorrect login or incomplete form submissions)?			2	13
How would you rate the responsiveness of the system when navigating through different sections?			2	13
User Satisfaction (US)				
How would you rate your overall experience of using the e-Appointment System?			1	14
How would you rate the likelihood of recommending this system to other patients at UTHM Health Centre?			2	13
Comments and suggestions for future improvements :				
1.	-			
2.	It would be great to have this website as an application!			
3.	amazing system!			
4.	-			
5.	None			
6.	develop this as an application			
7.	It would be great if the system could sync with Google Calendar or Outlook.			
8.	All is okay			
9.	-			
10.	improve the mobile view			
11.	Good system!			
12.	N/A			
13.	-			
14.	N/A			
15.	Okay			

Table 8 *User Acceptance Testing – Administrators*

Question	1	2	3	4
Perceived Ease of Use (PEOU) Questions				
How would you rate the ease of accessing and managing appointments, including viewing details and assigning appointments?			4	3
How would you rate the ease of managing user accounts (searching, editing, deleting)?			6	1
How would you rate the ease of managing user accounts (searching, editing, deleting)?			5	2
Perceived of Usefulness (PU) Questions				
How would you rate the usefulness of the administrator dashboard, which displays all the important figures (statistics) for managing the health center’s appointments and users?			6	1
How would you rate the usefulness of the system in managing medical staff profiles and department assignments?			6	1
How would you rate the usefulness of the system in creating, editing, and managing announcements for the health center?			5	2
How would you rate the usefulness of the filter options for finding specific appointments or users?			7	
Attribute of Usability (AU)				
How would you rate the clarity and helpfulness of the instructions and labels within the admin interface?			7	
How would you rate the system's performance in terms of speed and reliability while performing administrative tasks?			7	
User Satisfaction (US)				
How would you rate your overall experience of using the administrative functions of the e-Appointment System?			6	1
How would you rate the completeness of features available for effectively managing appointments and users at UTHM Health Centre?			7	
Comments and suggestions for future improvements :				
1.	Try to develop application.			
2.	Improve system responsiveness.			
3.	-			
4.	Good			
5.	Warna oren kurang menarik, boleh ditukar warna eg fb, tiktok atau media sosial yang mempunyai pengikut ramai			
6.	The basic system is good, but still need software upgrade and update in the future.			
7.	brilliant idea to make it easier for people to book any appointments in the future			

5. Conclusion

The conclusion of the e-Appointment System for UTHM Health Centre encapsulates the project's objectives, achievements, and areas for improvement. The system successfully addressed challenges in managing healthcare appointments by improving accessibility, efficiency, and communication while streamlining operations at PKU. Key benefits include online booking convenience, efficient time management, automated notifications, real-time availability, and a platform for feedback. However, the system faced initial setup and training challenges, occasional technical issues, and user adaptability concerns. Recommendations for future enhancements include providing comprehensive user training and support, ensuring system scalability, regular maintenance, integrating user feedback, and developing a mobile application. Overall, the system has significantly improved healthcare services at PKU, with potential for further advancements to enhance user experience.

Acknowledgement

The authors express their gratitude to the Faculty of Computer Science and Information Technology at Universiti Tun Hussein Onn Malaysia for the support and encouragement extended to them.

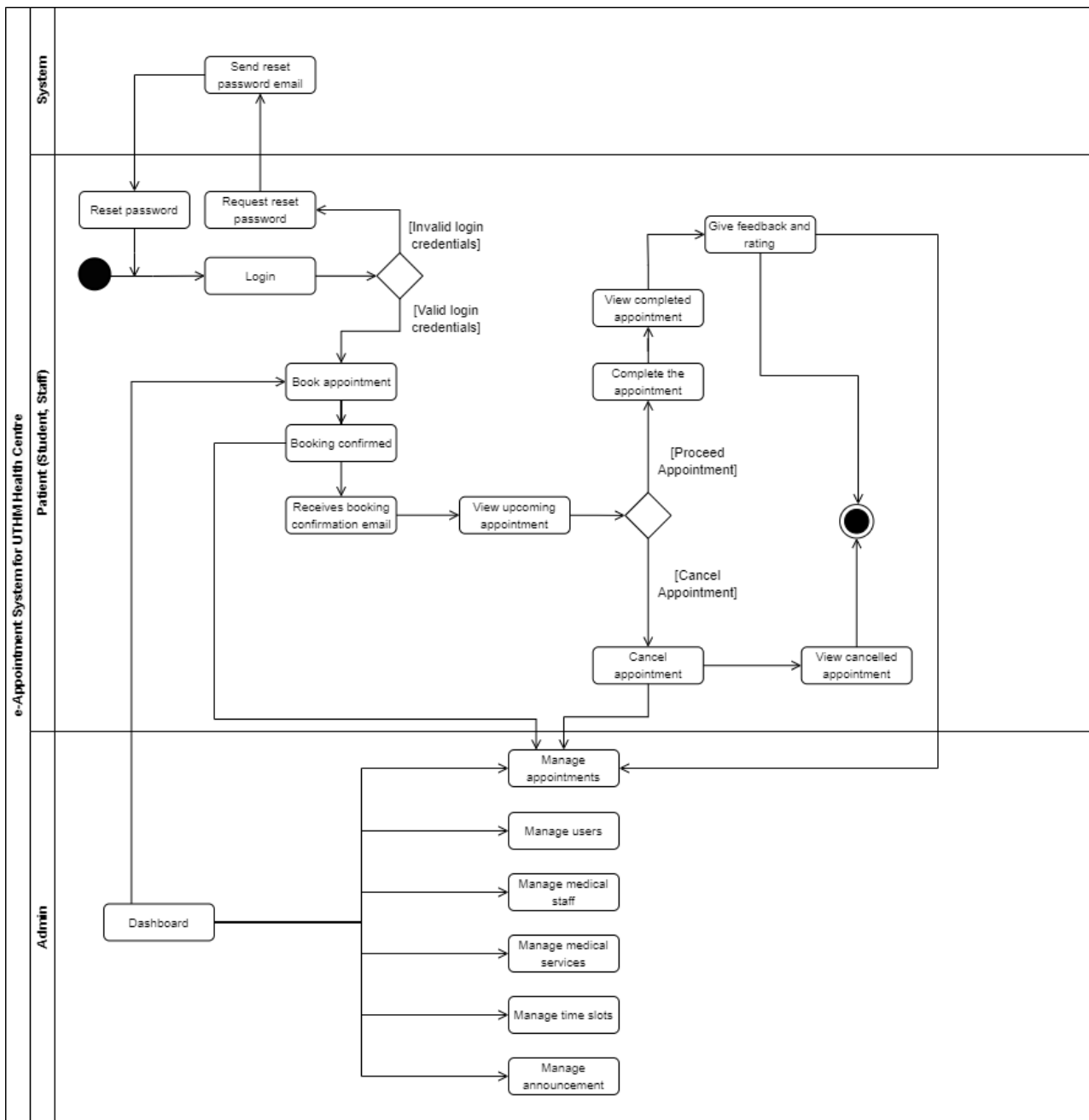
Conflict of Interest

The authors affirm that they have no conflicts of interest concerning the publication of the paper.

Author Contribution

*The authors acknowledge their contributions to the paper as follows: **study conception and design:** Siti NurFalahah binti Mohd Ridoi, Nur Ariffin bin Mohd Zin; **data collection:** Siti NurFalahah binti Mohd Ridoi; **analysis and interpretation of results:** Siti NurFalahah binti Mohd Ridoi, Nur Ariffin bin Mohd Zin; **draft manuscript preparation:** Siti NurFalahah binti Mohd Ridoi. All authors participated in the review of the results and approved the final version of the manuscript.*

Appendix A



References

- [1] WHO. (n.d.). Primary health care. World Health Organization. https://www.who.int/health-topics/primary-health-care#tab=tab_1
- [2] Ali, M. A. bin. (n.d.). Misi & Objektif. Misi & objektif. <https://health.uthm.edu.my/mengenai-pku/misi-objektif>
- [3] Ali, M. A. bin. (2019, October 14). Siapa Yang layak Menerima Perkhidmatan. Siapa yang layak. <https://health.uthm.edu.my/pekhidmatan/siapa-yang-layak>
- [4] PKU. (n.d.). Pusat Kesihatan Universiti | uthm. Pusat Kesihatan Universiti UTHM. <https://health.uthm.edu.my/>
- [5] MySejahtera Helpdesk. (2022, July 19). What is MySejahtera?. MySejahtera. <https://helpdesk.mysejahtera.malaysia.gov.my/en/support/solutions/articles/51000293086-what-is-mysejahtera->
- [6] Qmed Asia Team. (n.d.). Top healthcare company in Malaysia: Qmed Asia. Top Healthcare Company in Malaysia | Qmed Asia. <https://qmed.asia/our-story>
- [7] BookDoc Team. (2022, December 20). About Us. BookDoc. <https://www.bookdoc.com/about-us/>
- [8] AKSOY KARTCI, S., BUGDAYCI, S., KAVAKLI, B., ATICI, E., CIFCILI, S., & APAYDIN KAYA, C. (2020). The most common health care services needed by university students and employee. *Marmara Medical Journal*, 33(1), 27–34. <https://doi.org/10.5472/marumj.681967>
- [9] D. Bharadwaja, Ch. Bhavya Sri, G. Aswani, G. Sushma, & Ch. Prabhu Kiran. (2022). A doctor appointment booking system. *International Journal of Advanced Research in Science, Communication and Technology*, 75–79. <https://doi.org/10.48175/ijarsct-7601>
- [10] Pliannuom, S., Pinyopornpanish, K., Angkurawaranon, C., Pinyopornpanish, K., Wisetborisut, A., Likhitsathian, S., & Jiraporncharoen, W. (2021). Utilization of health care services and common disease diagnoses among university students: An analysis of 35,249 students from Thailand. *International Journal of Environmental Research and Public Health*, 18(13), 7148. <https://doi.org/10.3390/ijerph18137148>
- [11] Bagwan, K. I., & Ghule, S. D. (2019). A Modern Review on Laravel- PHP Framework. *Iconic Research and Engineering Journals*, 2(12), 1–3. <https://doi.org/https://www.irejournals.com/formatedpaper/1701266.pdf>
- [12] Floyd, C. (1984). A systematic look at prototyping. *Approaches to Prototyping*, 1–18. https://doi.org/10.1007/978-3-642-69796-8_1
- [13] Dennis, A., Wixom, B. H., & Tegarden, D. P. (2021). *Systems Analysis & Design: An object-oriented approach with UML*. John Wiley & Sons, Inc.