CHAPTER 6. SYSTEM TEST

6.1. Test Methods

After analyzing and developing this hospital application, the author conducted testing using the Blackbox testing method. Black-box testing is a method used to evaluate the functionality of an application without delving into its internal code structure. This approach focuses solely on examining the input and output of the software, treating it as a black box where the internal workings are unknown to the tester. By simulating user interactions and analyzing the system's responses, black-box testing aims to identify bugs, errors, and discrepancies between expected and actual outcomes, ensuring the software meets specified requirements and functions correctly from an end-user perspective.

6.2. Test Environment

Black-box testing environments for mobile applications, as highlighted in various studies, entail evaluating the functionality and performance of the app without requiring knowledge of its internal structure or codebase. For instance, in a research "Info Hospital: Web/Mobile Application based Health Care System" demonstrated this approach in the context of a web or mobile healthcare system called "Info Hospital," focusing on assessing the application's behavior and outputs [1]. Similarly, in a research "Design and development of mobile healthcare application prototype using flutter" explored black-box testing within the realm of healthcare applications developed using Flutter, emphasizing the importance of ensuring the software functions correctly from an end-user perspective without relying on insights into its underlying implementation details [16]. These studies underscore the significance of black-box testing in verifying the reliability, usability, and adherence to requirements of mobile applications across diverse domains.

6.2.1. Unit Test Tools

Testing in mobile app development is a crucial phase where developers evaluate the functionality, usability, and performance of the app across various devices and operating systems. It ensures that the app meets the requirements, functions as intended, and provides a seamless user experience. Table 6.1 is a table of the results of unit tests using the black-box testing method.

Test Scenario	Test Case	Test Steps	Result	Status
User Login from Login Page	User login using a valid patient account	1. The user is on the login page. 2. The user fills with valid email. 3. The user fills with valid password. 4. The user presses the Login button.	User enters the main page of the patient application.	Passed

	User login using a valid doctor account	1. The user is on the login page. 2. The user fills with valid email. 3. The user fills with valid password. 4. The user presses the Login button.	User enters the main page of the doctor application.	Passed
	User login using an invalid account	1. The user is on the login page. 2. The user fills with invalid email. 3. The user fills with invalid password. 4. The user presses the Login button.	User cannot login and get notification “Error. Login failed, please check your credentials”.	Passed
	User login with empty mandatory field	1. User is on the login page. 2. User leaves the mandatory fields empty. 3. User clicks the login button.	User cannot login and get notification “Error. Login failed, please check your credentials”.	Passed
User registered an account in Register Page	User registers an account by selecting the role of patient.	1. User is on the register page. 2. User fills out all fields. 3. User selects the patient role. 4. User presses the register button.	The user successfully registered an account and received the notification “Registration Success. Please login to continue”.	Passed
	User registers an account by selecting the role of doctor.	1. User is on the register page. 2. User fills out all fields. 3. User selects the doctor role. 4. User presses the register button.	The user successfully registered an account and received the notification “Registration Success. Please login to continue”.	Passed
	User registers an account by leaving mandatory fields empty.	1. User is on the register page. 2. User leaves the mandatory fields empty. 3. User clicks the login button.	User cannot register and get notification “Error. Register failed, please check your credentials”.	Passed
The user made an appointment on the consultation menu page.	The user fills in the section for selecting a doctor, date, and type of consultation.	1. The user is on the consultation menu. 2. The user selects a doctor along with their specialization. 3. The user chooses the day and time for the consultation. 4. The user selects the type of consultation. 5. The user presses the “make appointment” button.	The user successfully created an appointment, and the entered data will appear on the “MyAppointment” page.	Passed
The user made an appointment on the Medical Check-up menu page.	The user fills in the section for selecting date and package.	1. The user is on the medical check-up menu page. 2. The user selects the date for the medical check-up. 3. The user chooses the type of medical check-up package. 4. The user presses the “Make Appointment” button.	The user successfully created an appointment, and the entered data will appear on the “MyAppointment” page.	Passed
User is viewing the list of doctor schedules.	The user views the list of doctor schedules from the home page of the application, and	1. The user is on the main page of the application. 2. The user presses the “All” button next to the list of several duty doctors.	The page immediately navigates to the “Doctor List” page.	Passed

	when they press the "All" button, they can see the entire list of doctors in the "Doctor List" menu.			
	The user sees the list of doctor schedules from the "Doctor List" menu.	1. The user is on the main page of the application. 2. The user presses the "Doctor List" menu button.	The user can scroll and view the doctors' names and working hours.	Passed
Calls Ambulance	The user called an ambulance for an emergency situation.	1. The user is on the Ambulance page. 2. The user presses the "Contact via Whatsapp" button. 3. The user is in the WhatsApp application.	The user calls the ambulance.	Passed
Appointment History	The user pressed the "finish" button to complete the appointment.	1. The user is on the appointment detail page. 2. The user presses the finish button.	The completed appointment will appear on the "History Appointment" page.	Passed
	User presses the "Consult Again" button.	1. The user is on the history appointment page. 2. The user presses the "consult again" button.	Users can be directed straight to the consultation menu page to schedule a follow-up consultation.	Passed

Table 6. 1 System Testing Results by the Blackbox Method

6.2.2. Unit Test Case Analysis

Unit test case analysis is a critical phase of the testing process that lays the groundwork for effective and thorough testing of software components. It helps ensure that testing efforts are focused on the most important aspects of the software and that potential issues are identified and addressed early in the development lifecycle. Table 6.2 is the result of unit test case analysis for Hospital Mobile Applications.

No	Test Item	Yes	No
1.	User can successfully log in with valid credentials.	✓	
2.	User cannot log in with invalid credentials.	✓	
3.	User is logged out after explicitly logging out.	✓	
4.	User can successfully book an appointment with a doctor.	✓	
5.	Patient appointment time selection starts from today and onwards.	✓	
6.	User cannot book an appointment in the past.	✓	
7.	User can book overlapping appointments.	✓	

8.	User can view their own patient profile.	✓	
9.	Patient information is correctly displayed and formatted.	✓	
10.	Doctor information is correctly displayed and formatted.	✓	
11.	User can see the availability schedule of a doctor.	✓	
12.	Notifications are delivered for upcoming appointments.	✓	
13.	Users can make ambulance calls.	✓	
14.	Users can make a consultation again from the appointment history.	✓	
15.	User can view appointment details and join the meeting once the appointment has been accepted by the doctor.	✓	
16.	The status appointment changes to "finish" when the appointment is completed.	✓	
17.	The appointments that have been completed will appear on the Appointment history page.	✓	
18.	User can modify their personal data on the profile page.		✓
19.	User can choose consultation appointment date		✓
20.	User can cancel the appointment		✓

Table 6. 2 Results of Unit Test Case Analysis

6.3. Test Conclusion

After conducting Blackbox testing and unit test case analysis on the hospital application, it is evident that the application performs well in terms of basic functionality, user experience, and data integrity. The Blackbox testing method successfully verified various features such as account registration, user login, appointment creation, and viewing doctor schedules, resulting in positive outcomes. Users could interact with the application seamlessly, navigating through its features with ease. However, the unit test case analysis identified areas for improvement, including addressing security vulnerabilities related to session management, enabling users to update their patient information, ensuring doctor profile viewing functionality, and resolving issues with notification persistence across app restarts. Overall, while the application demonstrates strengths in functionality and usability, addressing these identified areas for improvement will enhance its security, usability, and overall user engagement, thus providing a more seamless healthcare experience.