MINISTRY OF EDUCATION
IMAM ABDULRAHMAN BIN
FAISAL UNIVERSITY
COLLEGE OF COMPUTER
SCIENCE
& INFORMATION
TECHNOLOGY

وزارة التعليم
جامعة الإمام
عبد الرحمن بن فيصل
كلية علوم الحاسب
وتقنية المعلومات

جامعة الإمام عبد الرحمن بن فيصل
IMAM ABDULRAHMAN BIN FAISAL UNIVERSITY

CIS 512 Software Quality Assurance

Assignment 1

Due Date: 24th November 2024.

Doctor Appointments Booking System Software Quality Assurance Test Design Report

Student
Raghad Aljassim
Solaf Mohammed Alhotailah
Aishah Saleh AlKatheer
Rayhanah Jassim Aldolaim
Batool alsaffar
Hajer Abdullah Alsaleh
Reem Zaki Almahfoud
Zainab Saeed Alfandi

Contents

List of Tables

List of Figures

1. Introduction

This paper describes the test plan for Doctors Appointments App, which is compliant with IEEE standards. The purpose of the test plan is to offer a thorough structure for testing procedures to guarantee the app's functionality, reliability, security, and usability. It contains information about test items, what features to test and not to test, the testing approach, pass/fail criteria, the testing process, environmental needs, change management, and plan approvals.

1.1.Objectives

Establishing a methodical and organized approach to testing the Doctors Appointments App is the aim of this test strategy. The strategy guarantees:

1. The app's key features have all been extensively tested.
2. To verify functionality, security, performance, and user experience, testing methodologies are established.
3. an awareness of the pass/fail criteria.
4. Verifying that the system satisfies all requirements and is prepared for deployment is the final objective.

1.2.Testing Strategy

Our approach is aimed at confirming every aspect of the app's performance, dependability, and safety by utilizing structured testing techniques and levels.

1- Testing Stages:

- **Unit Testing**: To make sure they work properly when used alone, each app module—such as appointment scheduling, user login, and notifications—will be tested separately.
- **Acceptance Testing**: This last phase makes sure the application meets user needs and functions as planned in practical situations.
- **Integration testing:** involves examining how different modules work together, for as how the doctor's calendar and patient appointment scheduling synchronize.
- **System Testing**: To make sure all functional and non-functional requirements from the SRS are satisfied, a thorough system evaluation will be carried out.

2- Methodology for Testing:

- **Testing manually** by a person without the use of automated tools,This will include non-repetitive tests such as security tests, recovery tests, and user acceptance tests (UAT).
- **Testing automatically**, Automated tools will be used to automate repetitive tasks like regression and load testing.

3- Evaluation Environments:

- **Environment for development**

Testing will occur in the development environment using artificial data to isolate and test components through unit and integration testing.

- **Testing environment for user acceptance**

In a UAT environment mirroring the production configuration, real doctors and patients will be involved as users to confirm system readiness.

1.3.Scope

This test plan's scope comprises:

- Functional testing of functions like calendar synchronization, account management, notifications, and appointment scheduling.
- verification of privacy compliance and data security for sensitive data, such as medical records.
- confirmation of the system's functionality under various loads.
- Evaluation of the user interface's usability and accessibility.
- Assessment of recovery and error-handling systems.

1.4.Reference Material

The test plan relies on the following references:

- Doctors Appointments App **Software Requirements Specification (SRS)**.
- Doctors Appointments App **Software Design Specification (SDS)**.
- Relevant IEEE standards for software testing and quality assurance.

1.5.Definitions and Acronyms

- User Acceptance Testing, or UAT
- Software Requirements (SRS) Details
- Software Design Specification, or SDS

- Electrical and Electronics Engineering Institute (IEEE)
- Mock Data: Test-related data that has been simulated
- Regression testing is the process of retesting to make sure modifications haven't impacted functioning.
- Load testing: Evaluating how well an application performs when used frequently

2. Test Items

Doctor appointments booking system has many items and requirements that need to be considered and tested, and they are:

1. Requirements Specification:

Table 1: Functional requirements

User	Functionality	ID
Patients	Sign up	**FR1**
	Login	**FR2**
	Reset password	**FR3**
	Search appointment	**FR4**
	Book appointment	**FR5**
	View appointment	**FR6**
	Cancel appointment	**FR7**
	Receive reminders	**FR8**
Doctors	Sign up	**FR9**
	Login	**FR10**
	Reset password	**FR11**
	View schedules	**FR12**
	View patients' details	**FR13**
Admin	View users' data	**FR14**
	Accept new doctor`s registration	**FR15**
	Generate reports	**FR16**
System	Handle schedule conflicts	**FR17**
	Send notifications and reminders	**FR18**

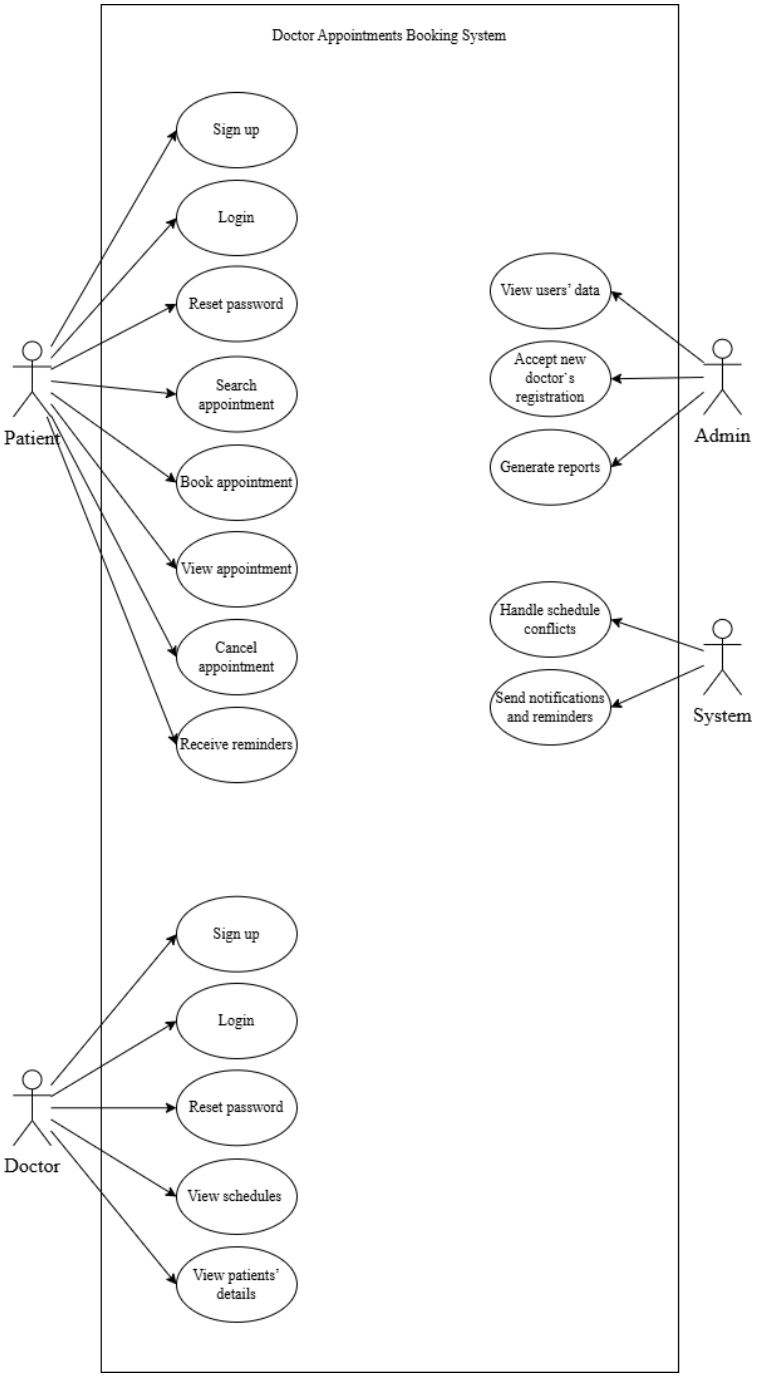

Figure 1: Use case diagram

2. **Design Specification:**
    a. Database for storing appointments, patients, and doctor data.
    b. Graphic user interfaces for patients, doctors, and admins.
    c. Backend APIs to handle appointment management and notifications.
3. **User Guide:**
    a. Clear step-by-step instructions for patients to use the system smoothly.
    b. A complete guidance for doctors to update and view patient details and appointments.
4. **Operations Guide:**

Admins` tasks Instructions including user and database management and generating reports.

5. **Installation Guide:**
    a. Uploading the software on cloud.
    b. Configuring database connections and setting user roles.
6. **Features:**
    a. Availability: System stability of 99.9%.
    b. Response Time: Appointment booking response in under 2 seconds.
    c. Security and privacy: Data encryption, OTP, and 2-factor authentication.
7. **Process of Defect Removal:**
    a. Prioritize bugs:

Table 2: Bugs prioritizing

High	Security issues
	Appointments conflicts
	Slow response times
	Unable to handle users number increasing
Medium	UI design
	Delayed data updates
	Delayed notification sending
	Sending reminders for appointment has been already canceled
Low	Typos

    b. Use a bug tracking system and other testing techniques for fixes and retests.

8. **Verification and Validation Plans:**
    a. Ensuring compliance with healthcare regulations and other laws and legals.
    b. Ensuring the usability with test users (doctors and patients).

2.1.Programming Modules

Table 3: Program modules

NO	Module	Items
1	Patient module	• Test registration, login, and reset password, • Test search, book, view, and cancel the appointments • Test receiving notifications for upcoming appointments.
2	Doctor module	• Test sign up, log in, reset password, • Test viewing the schedules and patients details.
3	Admin module	• Test managing users and system data. • Test receiving and accepting new doctors' requests. • Testing generating reports.
4	System module	• Test managing schedule conflicts. • Test sending reminders.

2.2.Job Control Procedures

1. **Appointment Scheduling**: Ensuring the process of search / viewing / booking / canceling an appointment and receiving reminders are logical properly.
2. **API Calls**: Validating backend logic and the process of handling concurrent booking attempts properly.

2.3.User Procedures

Testing all patients, doctors and admins documentations by answering the following questions:

1. Does the document of patients describe how to book, view, cancel an appointment easily?
2. Does the document of doctors include clear instructions about requesting registration and viewing the schedules and patients data?
3. Does the document of admins is enough to help them with viewing and managing users data and generating reports processes?

2.4.Operator Procedures

1. Ensuring that the help line is available.
2. Monitoring the system and fixing defects like double bookings.

3. Features to Be Tested

3.1 Registration

Feature: User registers.

Test Objectives: To ensure that users can register.

Test scenarios:

1. Ensure that the account registration page is available and clear.
2. Ensure that the user can register if the national ID is not already present in the system.
3. Ensure that registration information is saved in the database.

Test Pass/Fail criteria: All tests must be passed without error and the changes happened correctly.

3.2 Login

Feature: User login.

Test Objectives: To ensure that users can login for authorized users.

Test scenarios:

1. Ensure that allowed users can log in using their ID and password.
2. Ensure that the ID number is in the system and, if it isn't, show an error message.
3. Ensure that if the password is entered incorrectly, the system will display an error message.
4. Ensure the system locks when a user tries to enter the password more than is permitted.

Test Pass/Fail criteria: All tests must be passed without error.

3.3 Patient module: Search, book, view, and cancel the appointments

Feature: User search, book, view, and cancel the appointments.

Test Objectives: To ensure that users can search, book, view, and cancel the appointments.

Test scenarios:

1. Ensure that the user can search for appointments and get the correct results.
2. Ensure that the user can book and that the system booked the appointment, and it no longer exists.
3. Ensure appointments are viewed correctly and clearly to the user.
4. Ensure the cancelled appointment is correct and reappeared for booking.

Test Pass/Fail criteria: All tests must be passed without error and the changes happened correctly.

3.4 Patient module: Receiving notifications for upcoming appointments

Feature: User receiving notifications for upcoming appointments.

Test Objectives: To ensure that users can receive notifications for upcoming appointments.

Test scenarios:

1. Ensure that the notification reaches the user in a timely manner.
2. Ensure that the notification appears correctly and clearly.
3. Ensure that the user can view the notification details.

Test Pass/Fail criteria: All tests must be passed without error.

3.5 Patient module: Viewing the schedules and patient's details

Feature: User viewing the schedules and patient's details.

Test Objectives: To ensure that users can view the schedules and patient's details.

Test scenarios:

1. Ensure that the authorized user can view the patient schedule.
2. Ensure that any changes to patient schedules are also made to the user at the correct time.
3. Ensure that the user can view patient information details.
4. Ensure that patient information is displayed clearly and correctly.

Test Pass/Fail criteria: All tests must be passed without error and the changes happened correctly.

3.6 Managing users and system data.

Feature: Admin manages users and system data.

Test Objectives: To ensure that admin manages users and system data.

Test scenarios:

1. Ensure that the form is available for new doctors to register.
2. Ensure that the form reaches the admin.
3. Ensure that the admin can accept the doctor and if he accepts, he can add his data to the database.
4. Ensure that the admin can reject the doctor, and if he rejects him, he cannot enter the system.

Test Pass/Fail criteria: All tests must be passed without error and the changes happened correctly.

3.7 Managing schedule conflicts

Feature: The system manages schedule conflicts.

Test Objectives: To ensure that system manages schedule conflicts.

Test scenarios:

1. Ensure that the system gives a reservation for only one patient and that the reservation is not for more than one patient.
2. Ensure that changes and updates are implemented.

Test Pass/Fail criteria: All tests must be passed without error and the changes happened correctly.

4. Features Not to Be Tested

- **Module:** Patient and doctor module.

  Feature not to be tested: Test reset password (for patients and doctors)

  Reason: If the password reset feature has been implemented and tested in previous versions, we may not need to conduct comprehensive testing again. We can verify that the basic functionality works, but it might not be essential to allocate significant resources to it unless there are major changes.

- **Module:** Admin module.

  Feature not to be tested: Test receiving and accepting new doctors' requests.

  Reason: If the system for handling doctor requests is already well-established and has been tested in earlier versions, we might not need to test it again extensively. Instead, we can focus on new features rather than reviewing established and reliable processes.

- **Module:** Admin module.

  Feature not to be tested: Test generating reports.

  Reason: We can postpone extensive testing if report generation is a low priority for the initial launch and relies on stable existing data sources. Instead, we should concentrate our initial testing efforts on the more critical features that affect user interaction.

- **Module:** System module.

  Feature not to be tested: Test sending reminders.

  Reason: If reminders are managed by a third-party service that has proven reliable in previous projects, we might not need to conduct comprehensive testing. We can verify the basic functionality, but in-depth testing may not be as essential.

5. Approach

The testing approach will be conducted to our project Doctor Appointments Booking System will cover a variety of testing to achieve quality assurance. Each test has objectives to achieve, methods, criteria fit to the project. It includes:

5.1. Component Testing

Objective: this test conducted to verify individual components or modules to check if they function as required, and detect any defects in early stage in development process

Method and Criteria:

- Unit testing will be conducted to verify individual functionality
  create test cases to check:
  - Dealing with input errors, such as entering a number in the name field
  - Ensure that there are available appointments and that the appointment duration calculation is correct
- Intergration testing will be conducted to test interaction between components
  - Correct and clear display of the appointment schedule and availability of doctors
- System testing will be conducted to test entire system
  - Performing a security test to check for vulnerabilities that may lead to the leakage of patient and medical staff data

- **Criteria's:**test indivduial components and how they interact with each other

Inputs: Module code, unit test cases.

Outputs: Unit test reports, code coverage reports.

5.2. Integration Testing

Objective: this test is conducted to verify the interaction between components and how they exchange data. It aims to detect any defects or issues when components are combined to work together.

Method and Criteria:

- Big bang integration test will be conducted to test all component as a whole
- Incremental integration testing will be conducted starting from individual components to integrate all the components.

- **Criteria's:**

– Verifying the system's performance under difficult conditions, such as reservations at peak times.
– Verify the compatibility of interface components when exchanging data.

Inputs: Module code, integration test cases

Outputs: Integration test reports, and interface documentation updates.

5.3.Conversion Testing

Objective: this test will be conducted to measure the effectiveness of the system in converting from old format to new one.

Method and Criteria:

- Test cases will be conducted to verify the data conversion process is correct.
  Criteria's:
  – Ensure that all data elements and historical data in the system are not damaged or damaged during the transfer process

Inputs: Data conversion scripts, historical data, conversion test cases

Outputs: Conversion test reports, data validation reports.

5.4.Job Stream Testing

Objective: this testing to ensure that the application operates in the production environment.

Method and Criteria:

- Create test cases design to all possible scenario's
- Use defect tracking system to track defect and if there are defects assign them to developer to solve them
  Criteria's:
  – Verifying that the system is working properly and in the correct sequence

Inputs: Production job scripts, job scheduling documentation.

Outputs: Job stream test reports, production job schedules

5.5.Interface Testing

Objective: This test focuses on ensuring that the application works efficiently and effectively and that interactions and data exchange work correctly with the external system and its interfaces

Method and Criteria:

- User interface test will be conducted to test the interactions are operate correctly with backend system.
- Test cases will be conducted to verify communcation between the application and external system.
- **Cerifera's:**
  - Verifying that data was transmitted, received and processed correctly

Inputs: Interface specifications, interface test cases.

Outputs: user interface test report, updated interface documentation.

5.6.Security Testing

Objective: this test will be conducted to verify the functionality of application system control and audit feature are conducted.

Method and Criteria:

- Test cases will be conucted to verify authentication, authorization, data privacy, access control.
- **Criteria's:**
  - Ensure that protection features work as intended and reduce vulnerabilities.

Inputs: Security requirements, security test cases

Outputs: Security test reports, security audit logs

5.7.Recovery Testing

Objective: This test is performed to ensure the system's ability to recover after a failure or disaster. It verifies backups and its ability to restore application features as they were before.

Method and Criteria:

- Test case will be conducted like in case if system failure and how it will recover from disaster.

Criteria's:

- Ensure that data recovery and system procedures are working efficiently and effectively**.**

Inputs: Disaster recovery plan, recovery test cases.

Outputs: Recovery test reports, recovery documentation updates.

5.8.Performance Testing

Objective: this test conducts to evaluate system performance criteria's like speed, response, system robustness under pressure.

Method and Criteria:

- Test cases will be conducted to check speed, availability, stimulate how the system works under pressure.
  - **Criteria's:**
  - Achieving specified performance criteria successfully.

Inputs: Performance requirements, performance test cases.

Outputs: Performance test reports, performance optimization recommendations.

5.9.Regression Testing

Objective: this test is conducted to verify modifications applied don't have negative side effect on functionality was already tested in the application or system.

Method and Criteria:

- Test cases to check modifications doesn't effect on existing features and modules.
- **Criteria's:**
- Ensure there are no new bugs or defects in the system.

Inputs: Previous test cases, code changes.

Outputs: Regression test reports, updated test cases

5.10. Acceptance Testing

Objective: this test is conducted to verify the acceptance criteria is deployed and it meets specified requirements and ask customer to accept it or not.

Method and Criteria:

- User Acceptance test (UAT) conducted to customer to check if it meets their needs.
- Test cases will depend on defined customer criteria.
  - **Criteria's**
  - Meet customer needs and goals.

Inputs: Customer acceptance criteria, user acceptance test cases.

Outputs: Acceptance test reports, customer acceptance documentation.

5.11. Beta Testing

Objective: this test is conducted by limited users using a pre-release version of system to verify the compliance with business functional requirements and to gather feedback, report issue or defects before releasing the certified version.

Method and Criteria:

- Give access to end user for pre-release version.
- **Critia's:**
- Successfully identify and report system defects and faults.

Inputs: Beta test plan, pre-release version of the system.

Outputs: Beta test reports, defect reports.

6. Pass / Fail Criteria

1. Booking Functionality

Pass:

- The user can choose a time, date, and doctor to successfully schedule an appointment.
- Appointment confirmation display immediately after successful booking.
- Booking conflicts are avoided by the system.

Fail:

- The user can’t choose a time, date, and doctor to schedule an appointment.

- No confirmation message after booking.
- Booking conflicts occur (e.g., double-booking).

2. Availability of Doctors and Appointment

Pass:

- Doctor's schedules are accurate and show the real time availability.
- User can select doctors based on clinics, location, and availability.

Fail:

- Doctor's schedules are inaccurate and don't show the real time availability.
- system show inaccurate timeslots.
- user can’t select based on criteria.

3. Confirmation

Pass:

- Confirmation is sent to user after booking an appointment.

Fail:

- No confirmation is sent to user after booking an appointment.

4. Secure payment system

Pass:

- User can successfully make their payment through secure payment system.

Fail:

- User make their payment, and add their payment information through unsecure payment system.

5. Booking Changes/Cancellations

Pass:

- User can successfully reschedule their appointment.
- User can successfully cancel their appointment.

Fail:

- User can't reschedule their appointment.
- User can't cancel their appointment.

6. User Interface and Usability

Pass:

- The system is compatible with a variety of devices and is mobile-friendly.
- The system doesn't lag and loads rapidly.

- Clear booking process, and instructions.

Fail:

- The system isn't compatible with a variety of devices.
- The system isn't mobile-friendly.
- Complicated booking process, and instructions.
- The system crashes or lag during the appointment process.

6.1.Suspension Criteria

Suspension of testing activities may occur if any of the following conditions are met:

1. Critical System Failures
    - If the appointment system fail to work or crashes entirely.
2. Vulnerabilities in Security
    - If the system unsecure and vulnerable to add personal information and payment details.
3. Major data problem
    - if the system displays inaccurate information, such as inaccurate doctor or appointment times.
4. Major functions failure
    - If the major features such as scheduling or verifying appointments are fail to function properly.
5. Bad User Interface
    - if the system stops functioning (for example, buttons stop working, pages don't load).

6.2.Resumption Criteria

1. Solve Critical System Failures

- Condition: Solve Critical System Failures to making sure that major functions are function properly.
- Test Items to Repeat:
    - Booking appointment and confirmation process.
    - Cancellation and rescheduling features.

2. Solve the security and Vulnerabilities problems

- Condition: Ensuring that the system is free from security issues, and address and solve all vulnerabilities problems.

- Test Items to Repeat:
    - Checking the security of adding payment details.
    - privacy of personal information.

3. Correct data integrity issues

- Condition: Data issues have fixed , ensuring that all data is accurate and consistent.
- Test Items to Repeat:
    - Verify of the doctors availability
    - Verify the appointment schedules and times.
    - Verify the accuracy of data (doctors name, user details).

4. Fix the bad user inter

- Condition: The user interfaces is function well with no broken buttons, or problems.
- Test Items to Repeat:
    - Verify the responsiveness of all buttons, links, and forms.
    - Verify that the system works with various devices and mobiles.

6.3.Approval Criteria

To approve test results and proceed with the Doctor appointment System, the following conditions must be met:

1. All Critical Test Cases Passed
    - Condition: All critical test cases must passes successfully without errors.
    - Approval requirement: Test cases that cover the system's essential functions ought to run and pass without any problems.

2. Standards for Security Are Fulfilled
    - Condition: All security tests must passes successfully without errors.
    - Approval requirement: Security audits must demonstrate that no vulnerabilities exist.

3. Functionality of the Interface and User Experience
    - Condition: The system must be compatible with a variety of devices and be mobile-friendly.
    - Approval requirement: UI auditors confirm that the system elements are function and easy to use

4. Data Integrity Verified
    - Condition: The system must guarantee precise data storage and retrieval free from mistakes or corruption.
    - Approval requirement: Data consistency across various system components and data integrity should be verified by test results.

7. Testing Process

7.1.Test Deliverables

Throughout the testing procedure, the following deliverables will be produced:

Test Plans: Comprehensive documents that specify the goals, resources, scope, and methodology of the testing.

Test Cases: Detailed test cases that cover functional, usability, security, and performance aspects of every feature.

Testing reports: Are summaries of testing operations that include defect records and pass/fail outcomes.

Defect tracking: A record of faults found, together with information on their severity, status, and resolution.

7.2.Testing Tasks

Among the testing tasks will be:

Test Planning: Specify the goals, timetable, and extent of the testing.

Test Case Design: Create test cases in accordance with specifications and requirements.

Test execution: Run both automated and manual tests while documenting the findings.

Reporting Defects: Record any flaws discovered during testing and rank them according to their seriousness.

Regression Testing: Retest application components to make sure no new problems arise after bugs have been addressed.

User Acceptance Testing (UAT): Assist real users (patients and doctors) in testing the application to ensure it satisfies user needs.

7.3.Responsibilities

The testing responsibilities are distributed as follows:

Testing Team: the team will conduct the test cases, document results, and report problems.

Project Manager: is responsible for monitoring test process, making sure that deadlines and quality requirements are met.

Developers: they're responsible for resolving the issues and making the required adjustments.

Stakeholders: They offer suggestions during UAT and give their approval for the finished product.

7.4.Resources

The resources needed for the testing process are:

Testing Instruments: Regression and performance testing are conducted using automated testing tools, while usability evaluations are conducted manually.

The test environment is a specialized setting that replicates the production system in order to provide precise testing outcomes.

Documentation: includes requirement specifications, design documents, and user guides.

Development Team: Is the group of developers in charge of producing test cases.

7.5.Schedule

- Test Planning and Design [Start Date – 15/11/2024, End Date – 20/11/2024]
- Test Case Development [Start Date – 21/11/2024, End Date – 26/11/2024]
- Test Execution (Functional, Usability, Security) [Start Date – 27/11/2024, End Date – 3/12/2024]
- Performance Testing [Start Date – 4/12/2024, End Date – 5/12/2024]
- Regression Testing and UAT [Start Date – 6/12/2024, End Date – 7/12/2024]
- Final Reporting and Review [Start Date –7/12/2024, End Date – 7/12/2024]

8. Environmental Requirements

8.1.Hardware

Laptops and Desktops:

• Contemporary computers equipped with a strong processor, ample memory, and quick storage for seamless operation of testing tools.

• Able to work with Windows, macOS, and Linux operating systems.

Mobile Devices:

- Smartphones for iOS and Android to test app compatibility across platforms.

Server:

- A powerful server to handle backend processes and store testing data.

Network:

- A reliable and fast internet connection for real-time testing and internal communication.
- Secure local network for controlled testing environments.

8.2.Software

Operating Systems:

- Windows 10 or macOS for testers' devices.
- Ubuntu or CentOS for the server environment.

Database:

- MySQL for storing user data, appointments, and logs.

Testing Tools:

- **Postman:** To validate APIs for secure and reliable data transfer.
- **Selenium:** To automate repetitive UI tests and ensure compatibility.
- **JMeter**: To test system performance under load and identify issues.
-

Version Control:

- GitHub for managing test cases and tracking code changes.

8.3.Security

- **Data Encryption:** Use HTTPS to secure data during transmission.
- **Access Control:** Implement strong passwords and two-factor authentication.
- **Backup and Recovery:** Perform daily backups to secure test data.
- **Environment Isolation:** Maintain separate testing environments to avoid affecting live systems.

8.4.Tools

- **Bug Tracking:** Jira for documenting and managing issues found during testing.
- **Automation:** Selenium to reduce manual effort in repetitive tests.
- **Performance:** JMeter for stress testing and analyzing system capacity.
- **API Testing:** Postman for checking API connections.
- **Communication:** Slack or Microsoft Teams for testers to coordinate and share results.

8.5.Publications

- **User Manuals:** Step-by-step guides for patients, doctors, and admins on using the app.
- **Technical Documents:** API specifications and system design details.
- **Test Reports:** Records of tests, results, and any problems found.

8.6.Risks and Assumptions

Risks:

- Delays in getting required devices for compatibility testing.
- Network problems affecting real-time testing.
- Testers need additional time to learn new tools.

Assumptions:

- All tools and devices will be available before testing starts.
- The testing environment will closely reflect real-world conditions.
- Testers are familiar with the tools required for the process.

9. Change Management Procedures

1- **Change Request Submission**
- Utilize a change request template containing:
- Change description
- Justification for the change
- Expected benefits

Log and track all requests systematically.

2- Approval Process
- Involve key stakeholders (e.g., test manager, project manager).
- Evaluate the impact on objectives before approval.

3- Checking and Evaluation
- Track metrics and gather team feedback.
- Adjust changes as needed if issues arise.

4- Documentation and Reporting
- Maintain a comprehensive change request log.
- Regularly report the status and impact of changes to stakeholders.

5- **Review and Continuous Improvement**
- Periodically review the process to address inefficiencies.
- Encourage team feedback for enhancements.

10.Plan Approvals

This plan is approved by:

Project Supervisor: **Ms. Ghada Alraqib**

Date: **24 / 11 / 2024**

Signature:

MINISTRY OF EDUCATION
IMAM ABDULRAHMAN BIN
FAISAL UNIVERSITY
COLLEGE OF COMPUTER
SCIENCE
& INFORMATION
TECHNOLOGY

وزارة التعليم
جامعة الإمام
عبد الرحمن بن فيصل
كلية علوم الحاسب
وتقنية المعلومات

جامعة الإمام عبد الرحمن بن فيصل
IMAM ABDULRAHMAN BIN FAISAL UNIVERSITY

CIS 512 Software Quality Assurance

Assignment 2

Due Date: 8th December 2024.

Doctor Appointments Booking System Software Quality Assurance Test Design Report

Student
Raghad Aljassim
Solaf Mohammed Alhotailah
Aishah Saleh AlKatheer
Rayhanah Jassim Aldolaim
Batool alsaffar
Hajer Abdullah Alsaleh
Reem Zaki Almahfoud
Zainab Saeed Alfandi

Contents

1. Introduction

Document overview

This report on the Doctor Appointment App test presents a thorough analysis of the testing methods and results for the application. It encompasses comprehensive details regarding the testing methods used, test cases carried out, results achieved, and any problems or bugs found throughout the testing process. The purpose of this report is to verify that the app's functionality, usability, and reliability fulfill the necessary criteria for effective appointment scheduling and management.

This document is the software test report of the xxx testing phase of the XXX software development project. It contains the results of tests, which were executed during the testing phase xxx.

Abbreviations and Glossary

1..1 Abbreviations

Abbreviation	**Definition**
Test ID	**Unique identifier assigned to each test case.**
NOK	The test failed or the result did not meet expectations.
POK	Partially OK – The test partially passed, with some issues.
NR	Not Run – The test case has not been executed yet.
NC	Not Completed The test started but wasn't finished due to interruptions or errors.
GUI	Graphical User Interface.

1.2 Glossary

Acronym	**Definition**
Bug ID	A unique identifier assigned to a reported defect or issue.
Test Platform	The hardware and software environment in which tests are executed.

Test Sheet	A document or system entry used to record test steps, results, and outcomes.
Test Case	A set of actions and conditions used to verify a specific feature or functionality.

References

1.3 Project References

#	Document Identifier	Document Title
[R1]	1	IEEE Software Test Plan.
[R2]	2	Test Cases checklist

1.4 Standard and regulatory References

#	Document Identifier	Document Title
[STD1]	1	IEEE Standard 1233 – 1998, IEEE Guide for Developing System Requirements Specifications.

1. The test results are documented with clear explanations of both the expected and actual outcomes.
2. Test cases are labeled with the specific feature being tested and assigned a unique number for easy reference.
3. Each test case is categorized based on the system module or functionality it covers and includes a priority level to indicate its importance.
4. Issues are tracked with a unique ID, associated test case ID, and cross-referenced with project management tools for resolution.
5. A standard template is followed for all test cases, including sections for test ID, objective, prerequisites, steps, expected results, actual results, and status (pass/fail).

2. Overview of Test Results

2.1. Test Log

The doctor appointment software (version 1.0) was tested on a dedicated test platform consisting of various testing tools (e.g., Selenium, Postman, JMeter) located in Quality Assurance Department, Eastern Province, from the 2024/11/15 to the 2024/12/11. The tests outlined in the software test plan for this phase were carried out.

Testers where:

- Rayhanah Jassim Aldolaim
- Batool Jaffar Alsaffar
- Zainab Saeed Alfandi
- Hajer Abdullah Alsaleh
- Solaf Mohammed Alhotailah
- Aishah Saleh AlKatheer
- Reem Almahfoud
- Raghed Ibrahim Aljassim

2.2. Rational for Decision

After executing a test, the decision is defined according to the following rules:

- OK: The test is set to the "OK" state when all steps are in the "OK" state. The real result is compliant with the expected result.
- NOK: The test is set to the "NOK" state when all steps of the test are set to the "NOK" state or when the result of a step differs from the expected result.
- Partial OK: The test sheet is set to a "Partial OK" state when at least one step of the test is set to a "NOK" state or when the result of a step is partially compliant with the expected result. à Keep it or remove it. Source of inconsistencies: criteria to set if the result is Partial OK may be qualitative
- NOT RUN: The default state of a test has not yet been executed.
- NOT COMPLETED: The test is set to the "Not Completed" state when at least one step of the test is set to the "Not Run" state.

Test results are listed in 3.

2.3.Overall Assessment of Tests

2.3.1. Qualitative Overall Assessment of Test

- **User Experience (UX):** The system is easy to use and learn. There are no confusing steps or processes.
- **User Interfaces (UI):** The interfaces are simple and clear for all types of users. The colors, fonts and sizes are consistent.
- **Functionalities:** The system performs all the functions correctly as planned and expected.
- **Error Handling**: The system provides less helpful error messages and some of them aren`t understood by non-technical people.
- **Security:** The system`s security needs to be improved by using more powerful tools and techniques.
- **Compatibility:** The screen responsive needs to be implemented to fit all devices screens.
- **Timing:** There is some delay in some functionalities like retrieving data from database and receiving notification.

2.3.2. Quantitative Overall Assessment of Test

- 85.71% of tests OK
- 0% of tests NOK
- 14.29% of tests POK
- 0% of tests NR
- 0% of tests NC

2.3.3. Statistics About Bugs and Enhancements

- Total number: 22
- Number of Critical: 1
- Number of Major: 8
- Number of Minor: 13
- Percentage of Enhancements: 81.82%

2.4.Impact of Test Environment

Any software system, including a system for scheduling doctor's appointments, depends heavily on its test environment for correctness and dependability. Test environments can have a wide range of effects, particularly when the actual conditions differ from the intended ones. These discrepancies could result from the usage of hardware, simulators, or software testing tools that don't accurately represent the real-world situation.

Impact of Test Environment on Doctor Appointment System Table

Area	Expected Condition	Real Condition	Impact
Hardware Issues	In the test environment, hardware (servers, devices) operates dependably.	Slow performance or malfunctions (e.g., printer or device failures) are possible problems with real hardware.	Disruptions may arise from incompatibilities or malfunctions that occur during live use.
Software Testing Tools and Simulators	Test tools mimic the optimal operation of the system (e.g., booking, notifications).	Real-world interactions, like patient behavior and system strain, might not be faithfully simulated.	Performance problems or bugs might go unnoticed, and the system might act strangely when used in real life.
User Behavior and Interactions	Users adhere to ideal, preset workflows (e.g., scheduling, confirming appointments).	Unpredictable user behavior (such as incomplete forms, missed tasks, or interactions that are prone to errors) is possible.	The system might not respond appropriately to user activities in the actual world, which could result in mistakes or a bad user experience.
Security and Privacy	Vulnerability scans and other controlled settings are part of security testing.	It's possible that they don't accurately replicate real-world security risks like hacking and data breaches.	Undiscovered security flaws could result in data loss during live usage or unauthorized access.
Network Conditions	Perfect network circumstances are used in the test environment (e.g., no latency, consistent connectivity).	Slow speeds, disruptions, or instability are possible in real-world networks (e.g., slow internet or downtime).	Tests may not detect network problems such as sluggish or erratic connections because the test environment typically has a fast, reliable network.

3. Detailed Test Results

3.1 Registration Test

Test ID	**1**	**Comment**	**Decision**
Test description	This test aims to test patient registration in the application.	User account is created successfully.	OK
Verified Requirement	Username ID Password	Username= String ID=Number Password= Password	
Initial conditions	1. Fill in all fields. 2. The password must consist of a combination of 8 numbers, letters and special characters.	-	
Tests inputs	Username ID Password	Field information must be entered in the correct format.	
Data collection actions	User information will be verified in the database and saved.	-	
Tests outputs	The user will be registered in the application.	-	
Assumptions and constraints	1. A user cannot be registered if he/she has registered before. 2. Administrators and doctors cannot register.	-	
Expected results and criteria	Register your account successfully and go to the login page.	-	
Test procedure			

Step number	Operator actions	Expected result and evaluation criteria	Result
1	Fill in the fields with username, ID number and password.	Cheek user input.	OK
2	Click on the register button.	Database checking.	OK
3	Display user login page.	Waiting for user input.	OK
4	Fill in the required fields with the password and ID number.	Cheek user input.	OK
5	Click on the login button	Database checking.	OK
6	Display homepage.	The user gets access.	OK

3.2 Login Test

Test ID	2	Comment	Decision
Test description	Test aims to test user login.	User login to application successfully.	OK
Verified Requirement	ID Password	ID=Number Password= Password	
Initial conditions	2. Fill in all fields. 3. The inputs should be correct.	-	
Tests inputs	3. ID 4. Password	Field information must be entered in the correct format.	
Data collection actions	User information will be verified in the database.	-	
Tests outputs	The home page will be displayed.	-	
Assumptions and constraints	Only those with an account can log in.	-	
Expected results and criteria	Log in successfully and go to the home page.	-	
Test procedure			

Step number	Operator actions	Expected result and evaluation criteria	Result
1	Fill in the fields with ID number and password.	Cheek user input.	OK
2	Click on the login button	Database checking.	OK
3	Display homepage.	The user gets access.	OK

3.3 Search, book, view, and cancel the appointments Test

Test ID	3	Comment	Decision
Test description	The test aims to test that the user can search, book and view the appointment.	User search, book and view the appointment successfully.	OK
Verified Requirement	1. Appointment availability. 2. Confirm appointment.	Check user input.	NOK
Initial conditions	The user has authorized access.	-	
Tests inputs	Select day and time. Select a doctor's name.	The user should choose the day, time and the doctor's name.	
Data collection actions	The system records the time, day and doctor.	-	
Tests outputs	Appointment search, view and book.	-	
Assumptions and constraints	Appointment availability. Appointment booked successfully.	-	
Expected results and criteria	The appointment was booked successfully for the user and is not shown to other users.	-	
Test procedure			
Step number	**Operator actions**	**Expected result and evaluation criteria**	**Result**
1	The user enters the day or the name of the doctor.	Cheek user input.	OK

2	Available appointments are displayed for the user to choose from.	Cheek user input.	OK
3	Click on book an appointment.	The database is updated, and the system reserves the appointment, and it becomes unavailable to other users.	OK

3.4 Receiving notifications for upcoming appointments Test

Test ID	**4**	**Comment**	**Decision**
Test description	The test aims to test the arrival of notification of upcoming appointments to the user.	Upcoming appointment notifications are successfully received by the user.	OK
Verified Requirement	User has received notifications for upcoming appointments.	-	
Initial conditions	The user has booked an appointment and has authorized access.	-	
Tests inputs	The user clicks on book an appointment.	-	
Data collection actions	Save appointment notification information and details.	-	
Tests outputs	Send appointment notification to user.	-	
Assumptions and constraints	The user has booked an appointment.	-	
Expected results and criteria	The notification is sent to the user successfully.	-	

Test procedure			
Step number	**Operator actions**	**Expected result and evaluation criteria**	**Result**
1	The user has booked an appointment.	The appointment is checked in the system.	OK
2	The appointment notification and details are sent to the user.	The system sends the appointment notification and details to the user.	OK

3.5 Viewing the schedules and patient's details

Test ID	**5**	**Comment**	**Decision**
Test description	This test conducted to enable user to view the schedules and patient's details.	-	POK
Verified Requirement	User has access to see the schedules and patient's details.	-	
Initial conditions	The user should have signed in already.	-	
Tests inputs	The user clicks on schedules button to see the more details.	-	
Data collection actions	Capture screenshots of the displayed schedules and patient details.	-	
Tests outputs	schedules are displayed correctly with accurate information (time, date, patient name, etc..)	-	
Assumptions and constraints	1. The data is correct and up to date.	-	

	2. The system is configured correctly. 3. Data is displayed without delay of retrieving from database.		
Expected results and criteria	1. Schedules and patients' details are displayed accurately and completely. 2. The user interface is easy to navigate. 3. Fast response.	-	
Test procedure		-	
Step number	**Operator actions**	**Expected result and evaluation criteria**	**Result**
1	The user sign in.	Login successfully	OK
2	Users press the schedule button to view schedules and patient's details quickly and without delay.	It expected to retrieve quickly but Due to massive data, retrieving data will take some time and it will delay.	NOK

3.6 Managing users and system data

Test ID	**6**	**Comment**	**Decision**
Test description	The procedures for adding, editing, and removing user accounts as well as handling user-	The purpose of this test is to verify that the hospital management software can manage	OK

	related system data are covered in this test.	user accounts and system data.	
Verified Requirement	2. User Management Features 3. Integrity of System Data	-	
Initial conditions	To handle people and data, a user needs to have administrator privileges. To test, the system must be stable with current user accounts.	-	
Tests inputs	1. User information (password, role, and username) 2. Data fields to be updated or deleted (system data parameters)	-	
Data collection actions	Check the database for modifications following user or data management actions.	-	
Tests outputs	User accounts that are successfully created, updated, or deleted. Verification of the system's data consistency and integrity after operations.	-	
Assumptions and constraints	These tests can only be carried out by users with administrator access. The system is set up properly to	-	

	support user management features.		
Expected results and criteria	1. Users can be added to the system with valid data. 2. It is possible to successfully update current users. 3. It is possible to delete user accounts, and the system appropriately reflects the modifications. 4. Every activity preserves the integrity of the data, preventing any loss or corruption.	-	
Test procedure		**-**	
Step number	**Operator actions**	**Expected result and evaluation criteria**	**Result**
1	Fill in the user details for a new account.	User information is approved for creation.	OK
2	Press the "Add User" button.	A new user is added to the database. Enter the most recent information.	OK
3	To update, select an existing user.	Enter the most recent information.	OK
4	Click on the "Update User" button.	Databases are updated with the latest modifications.	OK
5	Select a user to delete.	Confirm deletion action.	OK
6	Click on the "Delete User" button.	User is removed from the database.	OK

3.7 Managing schedule conflicts

Test ID	**7**	**Comment**	**Decision**
Test description	The test is to ensure system manages and solve schedule conflicts.	Users solve conflicts successfully.	OK
Verified Requirement	Ensure there is not conflict schedules.	-	
Initial conditions	1. Users should sign in already. 2. Users should have access to view schedules and manage conflicts. 3. Test data are available.	-	
Tests inputs	1. User ID. 2. Users click on the more details button to check conflicts. 3. Various conflict resolution strategies (canceling, rescheduling).	-	
Data collection actions	Capture screenshot of conflict schedules and find solution to manage it.	-	
Tests outputs	1. Conflict schedules are solved correctly. 2. Multiple conflict strategies are provided (canceling, rescheduling).	-	
Assumptions and constraints	The system is running correctly. The data is accurate and up to date.	-	
Expected results and criteria	1. Conflicts are detected so the user can solve them according to the strategies provided.	-	

	2. Appropriate conflict strategies are provided. 3. Changes implemented successfully.		
Test procedure		-	
Step number	**Operator actions**	**Expected result and evaluation criteria**	**Result**
1	The user sign in	Login successfully	OK
2	Users press the schedule button.	The button will schedule	OK
3	User check schedule conflict.	Conflict schedule appears	OK
4	User decided which conflict strategy to implement.	Users implement strategy successfully	OK
5	User implements changes	Users implement strategy successfully	OK
6	Users save the change and update the schedule.	Changes are saved in database.	OK