CLEVELAND STATE UNIVERSITY

1964

EEC 623 SOFTWARE QUALITY ASSURANCE

TEST PLAN

OPEN CLINIC

ANITHA THATISETTY (2845429)

NIKITA KONDAPALLI (2837105)

HARSHKUMAR GOHIL (2862904)

Testing Open Clinic: A Comprehensive Test Plan

This test plan outlines the approach and strategies that will be used to test the Open Clinic software version 0.8.0. Open Clinic is a patient management system designed for healthcare facilities, and its source code is available on GitHub. Features including patient registration, medical record management, appointment scheduling, and invoicing are all included in the software. Together, we will ensure the program is extensively tested and adheres to the requirements.

The test plan details the software that will be tested and the functions that will be examined. Additionally, it describes the testing plan, which comprises system and unit testing. This test plan also includes a detailed timetable for unit and system testing with responsibilities assigned, procedures for storing and analyzing test results, testing tools and environment, and test work items that will be created. With this plan's help, the project's testing part will be successfully completed within the allotted time frame and will be well structured and recorded.

1. Software to be tested

The software to be tested is the Open Clinic version 0.8.0, an open-source patient management system designed for healthcare facilities. Its source code is easily accessible on GitHub and was created in Java. Healthcare providers may manage patient medical data, appointments, and billing with Open Clinic. System testing, unit testing, and automated testing are just a few of the testing methods and techniques used to evaluate software. To ensure the program is completely examined, the project team will use a range of testing techniques and technologies, including JUnit, Selenium, TestNG, and Apache JMeter. Testing will be done across several platforms to ensure the software's interoperability with different systems.

2. Functions to be tested

The functions to be tested are outlined in the project plan as follows

As per the project plan, the Open Clinic software will be tested for its major functions, including patient registration, medical record management, appointment scheduling, and billing. These functions form the software's core, and their proper functioning is crucial for its overall performance.

I. The first function to be tested is patient registration.

Registration allows healthcare providers to store patient information for a long period. This function must be tested for accuracy and completeness, ensuring patient information is correctly captured and stored in the system. The testing team can develop test cases that evaluate the registration process, including capturing patient demographic data, medical history, and insurance information. The team can use automated testing tools like JUnit and TestNG to perform unit tests and ensure that the registration process works as expected.

II. The second function to be tested is medical record management.

Healthcare professionals can handle patient medical records thanks to medical record management. The testing team can create test cases to gauge how well this function manages patient diagnosis, prescriptions, lab findings, and imaging investigations, among other functions. To carry out functional tests and make sure that medical records are being recorded and stored properly, the team can use automated testing tools like Selenium.

III. The third function to be tested is appointment scheduling, The ability to plan appointments for patients, monitor appointment progress, and remind patients is provided through appointment scheduling. The scheduling process, including the ability to set up appointments, keep track of appointment progress, and send reminders, may be evaluated by the testing team by developing test cases. The team can run UI tests and ensure appointment scheduling functions as intended using automated testing tools like Selenium.

IV. The fourth function to be tested is billing.

By invoicing, healthcare professionals can create invoices for their services and manage patient billing data. The testing group can create test cases to assess the billing procedure, including the capacity to control billing data and produce bills. To perform load testing and ensure that the billing process can manage high volumes of data and transactions, the team can use automated testing tools like Apache JMeter.

Finally, testing of the Open Clinic software will include testing of the key features that make up the platform's core. To ensure the program operates as intended, the testing team will create test cases using automated testing tools and conduct functional, unit, UI, and load testing. The project plan offers a detailed time-line for testing activities and includes room for flexibility to account for last-minute modifications. The software will be high quality, trustworthy, and suit the expectations of healthcare professionals if the testing strategy is completed.

3. Testing strategy

The testing strategy for the Open Clinic software project will involve a combination of unit testing and system testing.

3.1 Unit Testing

Test specific program elements or modules, such as methods, classes, or functions. Unit testing aims to detect bugs or other faults in the code as early in the development process as feasible and to ensure that each component of software works as intended when used separately. While deciding which components to employ for unit testing, factors such as the component's significance, complexity, and frequency of use will be considered. For example, unit testing will be given more priority for parts that are both crucial to the operation of the software and often used.

The team members will share responsibilities for unit testing, one going primarily in charge of creating and running unit tests using JUnit and TestNG. To ensure that the unit tests are thorough and cover all essential components of the software, the other two will examine the test results. Creating test cases for each method or function in the product will be the strategy for designing unit test cases, emphasizing evaluating all potential input combinations and edge cases.

3.2 System Testing

System testing will be done to test the complete software system as a whole, including the integration of different components and their interactions. System testing is used to verify that the software complies with functional, non-functional, and business requirements and to find any flaws or faults in the way the product behaves in actual use.

Functional, performance, security, and usability tests are the different system tests that will be carried out. The Open Clinic software's numerous functions, including patient registration, medical record management, appointment scheduling, and invoicing, will be done through functional testing. Using Apache JMeter, performance testing will examine how the software performs under various workloads and stressful circumstances. Software's capacity to prevent unwanted access, maintain data confidentiality and integrity, and adhere to industry norms and regulations will all be tested as part of security testing. To determine whether the software is user-friendly and accessible to all users, usability testing will examine the program's user interface and user experience

.

The team members will share the responsibility for system testing. The team is wholly responsible for creating and carrying out system tests with the help of Selenium and other testing frameworks. Two of them will analyse the system tests to ensure they cover all crucial program functionalities and check for any errors or issues. As part of the system test case creation strategy, test cases are developed for each key software feature, focusing on assessing all potential scenarios and edge situations.

Unit testing and system testing will both be used in the testing plan for the Open Clinic software project. Together, the team members will ensure that the software is extensively tested and that any flaws or faults are found and fixed. The testing strategy will be based on each software component's criticality, complexity, and frequency of usage, emphasizing testing all feasible scenarios and edge situations. The team will develop and execute test cases using many testing frameworks and technologies, such as Apache JMeter, Selenium, JUnit, and TestNG.

4. Test work products

Test drivers, test cases, and test reports are among the testing work items that will be generated for this project. Test drivers will be developed to execute and automate the testing procedure for the key features of the Open Clinic software. These test drivers can also automate regression testing, saving time and money.

The functional decomposition of the software will serve as the basis for the test case design. These test cases will cover the main features of the Open Clinic software, such as patient registration, medical record management, appointment scheduling, and invoicing. Each test case will be created to cover a particular situation or use case, including instructions, anticipated outcomes, and actual outcomes. All functionality will be completely and properly examined by using these test cases as a reference during the testing process.

Data files will be created in addition to the test cases and test drivers. The input and output data utilized during testing will be stored in these data files. To ensure that the software is thoroughly tested, the data files will cover various scenarios and use cases.

Test reports will then be generated to record the testing procedure and outcomes. These reports will provide information on the testing procedure, including the testing environment, test outcomes, and any problems or flaws found during testing. These reports will document the testing procedure and be used to inform stakeholders of the outcomes of the testing.

This project's testing work outputs will be crucial in maintaining the Open Clinic software's quality. The team may ensure that all functionalities are tested thoroughly, precisely, and effectively by creating thorough test drivers, test cases, data files, and reports. These work products will also be helpful tools for the software development team and serve as a manual for future testing initiatives.

5. Test record keeping

Mechanisms for recording and analysing test results must be present in any project, including software testing. This enables the testing team to monitor the project's development, spot any problems, and ultimately guarantee that the software complies with the necessary standards. The project plan for the Open Clinic software testing project will include a section on test record management.

Task 5 is to record test results and report the findings, as stated in the project plan. This entails documenting the outcomes of the testing done during Task 4, during which various testing methods and methodologies will be utilized to examine the essential components of the software. The test record-keeping procedure entails writing down the outcomes of the testing phase and archiving them in a way that makes them accessible for analysis and retrieval.

The project plan specifies several tools and technologies the testing team will employ to make this process easier, including JUnit, Selenium, TestNG, and Apache JMeter. Thanks to these tools, the team will be able to perform automated unit testing, web application testing, and functional and scalability testing.

After the testing, the team will examine the data and provide reports describing the outcomes in detail. The reports will be set up to give an overview of the testing done, the findings, and any problems found. The reports will also include suggestions for resolving any problems found and enhancing the product's functionality.

Maintaining test records will also entail setting up a repository for the test findings and associated paperwork. The team can quickly access the test findings and documentation at any time and share them with other stakeholders.

The testing team will design precise rules for recording and preserving test results and related documentation to guarantee the effectiveness of the test record-keeping procedure. In doing so, it will be necessary to establish the reports' format, list the data that must be included, and lay out the steps for saving and retrieving the reports.

Last but not least, the testing team will regularly examine the test findings and associated documentation to make sure they are correct and current. To guarantee that the test record-keeping procedure continues efficiently for the length of the project, any problems discovered during the review phase will be rectified immediately.

In conclusion, the Open Clinic software testing project's test record-keeping is a crucial step. The testing team will be able to effectively track the project's progress, identify any issues, and make sure that the software complies with the necessary standards by using the tools and technologies specified in the project plan and establishing clear guidelines for documenting and storing test results and related documentation.

6. Testing tools and environment

To effectively test the Open Clinic software, as specified in the project plan, our team will use various testing methods and technologies. The test environment, which includes the tools, simulators, specialized hardware, test files, and other resources we'll utilize for testing, will be described in this section.

First, we will develop and run automated tests for the Open Clinic program utilizing JUnit, TestNG, and Selenium as our main testing frameworks. These testing frameworks, which allow a variety of testing techniques, including unit testing, functional testing, and integration testing, are frequently used to test Java applications. These frameworks will create and execute test cases for Open Clinic's basic functionality, such as patient registration, medical record management, appointment scheduling, and invoicing.

Second, we will assess the functionality and scalability of Open Clinic using Apache JMeter, a potent testing tool. Using Apache JMeter, we can simulate heavy user traffic, gauge server response times, and spot performance bottlenecks. This tool will test the system under load and ensure it can support many users without slowing down or crashing.

To ensure that our testing is exhaustive and precise, we will also use specialist hardware. For instance, we'll evaluate the Open Clinic user interface across various gadgets, including PCs, laptops, tablets, and cell phones. To ensure the program functions flawlessly across all platforms, we will test it on various operating systems, including Windows, macOS, and Linux.

Also, we will evaluate the program's performance and usefulness under numerous scenarios using a variety of simulators. For instance, we will simulate network delays and disruptions to make whether the application can manage unexpected faults and recover gracefully. Also, we will simulate various user roles and permissions to ensure that the application grants each user the appropriate level of access.

We will also use various test files and resources to ensure our testing is thorough. Test patient data will be used to evaluate the patient registration and medical record management functionalities. To test the invoicing feature, we will also use example billing data. Also, we will create and execute our test cases using various testing techniques like boundary value analysis, equivalence partitioning, and error guessing.

To successfully test the Open Clinic software program, our test environment will include a variety of tools, simulators, specialized hardware, test files, and other resources. To build and execute our test cases, we'll use a variety of testing frameworks, including JUnit, TestNG, and Selenium. Moreover, Apache JMeter will evaluate the program's functionality under load. To ensure that our testing is thorough and precise, we will also use specialist hardware and simulators.

7. Test schedule

A crucial part of our overall testing strategy is the test schedule for our Open Clinic software testing project. It describes the precise testing tasks that will be carried out, the timetables for each task, and the obligations of each team member. Using the test schedule, we can properly manage the testing process and produce high-quality software.

Unit and system testing will comprise the two primary sections of the test schedule. We will concentrate on testing certain software modules or components to make sure they are operating properly throughout the unit testing phase. The team, will work on this phase from April 3, 2023, through April 18, 2023.

We will create the unit test cases using the well-liked Java application testing frameworks JUnit and TestNG. Also, we will ensure that the test cases cover all of the essential software functionality. Then we will then run the unit tests and document the outcomes. Any flaws or problems found will be recorded for correction.

The system testing period starts on April 19, 2023, and lasts until May 10, 2023. Rather than evaluating individual components during this stage, we will concentrate on testing the software as a whole. We will combine manual and automated testing methods to ensure the program is working properly and satisfies all requirements.

Then we work on organizing the system testing procedures and ensuring that each test case is carried out. We carry out the automated tests using the popular web application testing tools Selenium and Apache JMeter. We will also conduct manual testing to ensure the program is user-friendly and satisfies all functional requirements.

We will record every test result and any problems or faults found throughout the testing procedure. We will use a collaborative solution like JIRA to keep track of these issues and ensure they are quickly resolved. In order to make sure we stick to our plan and deal with any problems that may emerge, we will also contact frequently with one another.

In conclusion, our test schedule is a crucial part of our testing strategy for the Open Clinic software. It describes the precise testing tasks that will be carried out, the timetables for each task, and the obligations of each team member. By adhering to this time-line and cooperating closely, we are convinced that we will create a high-quality software solution that satisfies all requirements.

Overall, the test plan offers an organized method for testing the Open Clinic software using various testing methods and methodologies. We will all work together to install the program, build test cases, and perform system testing. To guarantee that the project is finished within the chosen time-scale, the schedule provides flexibility and adjustments to be made as needed. The test plan offers a thorough method for testing the program, with unit testing, system testing, test work products, test record keeping, testing tools and environment, and a thorough testing schedule. The project plan and test plan are intended to ensure that the Open Clinic software is properly tested to offer a dependable and effective patient management solution for healthcare organizations.