SCHOOL OF TECHNOLOGY

BEACHELOR OF SCIENCE IN SOFTWARE DEVELOPMENT

UNIT CODE: BSD 3201

ADMISSION: 21/04706

NAME: DUNCAN MUNENE MURIUKI

NAME OF THE SUPERVISOR: WINNIE OKONG'O

PROJECT TITLE: FASTPRO RESOURCE SCHEDULING AND OPTIMIZATION WEB APP

TEST PLAN

Contents

TEST PLAN

1. Introduction

The requirements tested in this document include those of the FastPro Resource Scheduling and Optimization Web Application. They include the total test plan the main test procedures that are supposed to be followed with a purpose of achieving an efficient and reliable system.

1.1 Goals and Objectives

The primary goals of the testing process are:

- For the fact that the FastPro Resource Scheduling and Optimization Web Application is to perform certain tasks and has some required characteristics.
- To detect any faults or errors with the system and to make formal accounts of them and to fix them.
- To confirm the functionality and ease of use of the system in its interface for FastPro clients, such as the project managers and the resource schedulers.
- To provide high system reliability, performance and stability prior to system delivery.

1.2 Statement of Scope

The scope of this testing effort encompasses all key functionalities of the FastPro Resource Scheduling and Optimization Web Application, including:

- Resource Scheduling: Resource scheduling creation, resource assignment, and resource schedule updating based on dynamic inputs.
- Optimization Engine: Affirming that the optimization algorithm is the best structure to allocate resources between different tasks to maximize efficiency and avoid idle time.
- Reporting and Analytics: Creating and compiling complex reports and charts from the database, the extent of resource utilization, time costing of project schedules, and the completion percentage.
- User Management: Controlling user accounts and giving role and permission in such a way that unauthorized users cannot access or change any data.

The following aspects are excluded from the scope of this test plan:

- Performance under extreme load conditions (stress testing): This might be addressed at another phase of the performance test.
- Security testing (penetration testing): Security considerations will be addressed during a security concern review process employing the security Concern Review tools and security specialists.

1.3 Major Constraints

- Project Timeline: There is always a deadline to meet since testing has to be done within the project period in order to go-live.
- Resource Availability: The testing will be done by limited testers; the availability of testing resources may affect the time line.
- Software Access: The testers will be using the beta copy of the application, meaning that some or all of the features of the application could be slowed, or not implemented fully or at all as in the Release to Manufacturing (RTM) copy.

2.0 Test Plan for FastPro Resource Scheduling and Optimization Web Application

This part provides information on the general testing vision and major project management considerations to yield proper testing for the FastPro Resource Scheduling and Optimization Web Application.

2.1 Software (SCI) to be Tested

The software to be tested is hence the FastPro Resource Scheduling and Optimization Web Application, which is version 1.0.0. Whenever any excluded functionalities are there like security testing, they are described in Section 1.2.

2.2 Testing Strategy

In this case, the levels of testing that will be used includes: Successful testing will demonstrate that the whole system is functional and satisfies all functional and non-functional requirements.

2.2.1 Unit Testing

- Strategy: Specific software modules will be united, and their ability to work according to their descriptions and provided specifications will be checked.
- Selection Criteria: This type of testing will be done on each part of the critical system modules the resource scheduling module, the optimization engine, and report features. As such, rigorous algorithms including, resource optimization, will be preferred.

2.2.2 Integration Testing

- Strategy: While developing the modules, an aspect of progressive integration will be observed, to ensure that the modules' interactions especially where they are very interfaced, like the scheduling and optimization processes, are properly handled.
- Order of Integration: The first level of integration will involve key applications where resources will be allocated and scheduling done, followed by secondary, and more elaborate joint working involving reporting and analytical functionalities.

2.2.3 System Testing

- Strategy: Finally, there will be validation of the whole integrated system to determine whether the functional and non-functional requirements of usability, performance, and accuracy of the optimization engine are met.
- Order of Validation: First, testing will focus on those important functionalities within the application that will make a huge impact to the application success such as scheduling of resources and their allocation, then the application's occasional functionalities like reporting and analytical functionalities.

2.The second phase is also known as User Acceptance Testing (UAT).

Strategy: Some the testing processes will engage the end-users or the project managers and the resource planners with the specific tests on usability, realistic operations, and design experience. To satisfy the operational requirements, actual user feedback will be integrated into the system.

2.3. capitals that ought to be tested Testing Resources and Staffing

The test execution will be conducted by a separate testing team possessing knowledge about the different approaches to software testing.

Some of defects that often hale from the optimization algorithms or scheduling logics will be helped in fixing by the developers after testing.

Project manager is responsible for testing function, resource management as well as the realization of the test plan and the schedule.

2.4 Test Work Products

sample of test plans and procedures.

- The scenarios for the testing levels, namely, scheduling, optimization testing, and report testing scenarios.
- Abecded as TLU, test data are the data that are applied when tests run or are being conducted.

Test results logs and reports.

- Bugs report containing list of defects and their status on when they were found and if they were fixed.

2.5 Test Record Keeping

- All test plans, cases, results, and logs will be saved in a central shared document repository that any project stakeholder may access.
- The test results will be recorded in order to have a reference in the future, and for comparing the current and future performance.

2.6 Test Metrics

The following test metrics will be tracked and reported:

- Number of test case executions.
- Total number of defects detected with reference to their level of intensity.
- Defect resolution rate.
- Test coverage percent, it checks whether all resources and all kinds of optimization and report generation are exercised well through tests.

2.7 Testing Tools and Environment

A simulation of the production environment to be tested will be created; allocated resources for scheduling and optimization will also be created in anticipation.

Regression testing will be done using automated testing tools (if applicable); otherwise test cases will be automated to increase their efficiency in areas that may involve massively repeated actions.

The other area where manual testing will be used is where testing cannot be fully automated due to its type such as User Acceptance testing and Performance testing.

2.8 Test Schedule

A test plan will now include a specific time frame for unit, integration, system, and possibly, UAT testing. This schedule will be incorporated with the project master schedule based on the available resource, other dependencies and project timeframe. This testing schedule will guarantee that before deployment all functionalities have gone through their rigorous test.

3.0 Test Procedure for FastPro Resource Scheduling and Optimization Web Application

This subtopic aims to describe the actual test process as well as the test cases of the FastPro Resource Scheduling and Optimization Web Application at various testing levels.

3.1 Software (SCI) to be Tested

The software in question under this testing project is the FastPro Resource Scheduling and Optimization Web Application up to Version 1.0.0. Other unavailable features (, such as the security testing) are described in Section 1.2.

3.2 Testing Procedure

A structured approach will be followed for testing at each level of the application:

3.2.1 Unit Testing

Procedure:

- There will be unit tests of the application to create specific use cases for each crucial module with a focus on scheduling, optimization, and reporting abilities.
- Positive and negative test cases will be created along with boundary test cases and test cases based upon error conditions.

- For some of the test cases, especially at the unit level there will be the use of automated testing tools.

Example Unit Test Case:

- Component: Resource Scheduling Module
- Stubs/Drivers: We can use something like mocking framework to mimic those external dependences or some form of stubs.

Test Case:

- Input: Include a resource in the schedule and it will show particular time intervals.
- Expected Result: The resource is scheduled and shown on the calendar view as planned.
- Purpose: Ensure that the scheduling module is able to create and display resource in accordance to the input created by the user.

3.2.2 Integration Testing

Procedure:

- Components will be combined in a step-by-step manner, with primary functions, including resource management, appointment, and enhancement.
- Among integration test cases, data passing and communication between the integrated modules will be prioritized being tested, in case of when, for example, scheduling information has to be passed synchronously with optimization data.
- Any errors noted at the test integration level will be reported and explained to the programmers.

Example Integration Test Case:

Modules: Resource Scheduling is one of the components of the optimization engine.

Test Case:

- Allocate a resource and at the same time call for the optimization process for that resource.
- Make sure that the optimization engine correctly modifies schedule depending on the availability of the resource and optimization constraints.
- Purpose: Make sure that the optimization engine sits alongside the scheduling module correctly and that information is passed from one to another properly.

3.2.3 System Testing

Procedure:

- The whole system integrated will need to have the functional as well as the non-functional test to ensure compliance with the requirements such as usability, performance and optimization accuracy.
- System test cases will embrace main components such as resource scheduling, optimization processing, and reported capabilities.
- Test data will be generated based on the real application scenarios including resource available/ constrained/ maximized.

Example System Test Case:

Test Case:

- Using our example for the coursework, log in to the system as a resource manager.
- Start/End times of the new resource schedule, place and type of the resource.
- Call the optimization module to optimize the investments given availability and foreseeing constraints.
- It is important to ensure that the optimized schedule is well developed and all resources well assigned.
- Prepare a report on resources and the outcomes of resource management and resource utilization.
- Purpose: Imitate a full cycle of resource scheduling and optimization procedures and check system performance, correctness, and reporting.

3.2.4 High-Order Testing (to be addressed in separate documents):

- Recovery Testing: Process of system recovery from failure will be described in another test plan.
- Security Testing (if not excluded): Overall, a clear and specific security testing design will be provided in order to check the site's weakness.
- Stress Testing and Performance Testing: These will be included in another Performance testing plan which will encompass issues to do with a large number of schedules for resources and other optimization activities.
- Alpha/Beta Testing: This may be included depending on what procedures has been made in development process and the information required from the users.

3.Testing Resources and Staffing There is generally little supplied for testing in software development and very few companies have enough testers.

For more information on testing resources and staffing please see section two point three In terms of the testing team the following is important to note; The testing team must be experienced and must be tasked with the following duties; To test software products These responsibilities must be augmented by developer involvement in ways that include identification of defects.

3.4 Test Work Products

- Individual use cases for scheduling, optimization, and reporting of some selected unit tests.
- Simple end to end test cases.
- Full workflows-based system test cases.
- Test reports and the logs that detail the test execution, date and time and the ID of the tester, and result.
- Detailed bug reports stating problems at the time of bringing out the test.

3.5, Test Record Keeping und Test Log

All test documents which include test case identification and results, test logs will be kept in a document sharing system for ease of use and versions.

A comprehensive test log will be maintained to record:

- The time and date of test execution.
- Tester name.
- Test case ID and description.
- Test results (pass/fail).

If there were any discovered issues then the characteristics of the respective defects and the state of their fixation.

This structured fashion will make certain that this application has been tested from all fronts and the essential parts of the FastPro Resource Scheduling and Optimization Web Application will be thoroughly checked, in order to perfect the system.