Quality Assurance Plan

actiTIME - Time Tracking Software

<3rd November 2023>

IS4102 Advanced Software Quality Assurance

Assignment II

19020572 | G.K.G Perera

1 Introduction

1.1 PURPOSE

The primary purpose of this QA plan is to formalize the evaluation process and ensure the quality and reliability of the HR functionality within actiTIME. This plan will lay out the comprehensive testing procedures, objectives, roles and responsibilities, and timelines that will guide in achieving the desired level of quality for the HR feature of the actiTIME application.

1.2 PROJECT OVERVIEW

actiTIME is a feature-rich web-based tool designed to make project management and time tracking easier for companies and organizations. This flexible program gives customers the ability to track work hours, manage projects efficiently, and obtain insightful information on the performance of their teams. Its feature set includes comprehensive reporting choices, task management tools, project management functionality, and strong time-tracking capabilities.

2 Scope

2.1 IN-SCOPE

This Quality Assurance Plan includes the following in-scope items:

i. Verification of the HR portal's accessibility using a valid username and password.
ii. Verification of the ability to access employee profiles.
iii. Verification of the functionality to examine attendance and leave reports.
iv. Verification of the process for assessing timesheets for approval or rejection.

2.2 OUT-OF-SCOPE

The following features are out of scope for this Quality Assurance Plan:

i. Managing work assignments feature.
ii. Lock time tracking feature.
iii. Financial and performance reports.
iv. Creation and management of tasks.
v. Customer and project management feature.
vi. New user (employee) addition feature.

3 Testing Strategy

3.1 PRODUCT/APPLICATION/SOLUTION RISKS

Risks	Criticality	Mitigation Strategy
Security vulnerabilities	High	Update and patch the infrastructure and application regularly.
Others acquiring unauthorized access	High	Use techniques like multi-factor authentication to strengthen user authentication.
System interruptions or downtime	High	Continuously oversee the well-being of the system and establish redundancy measures for vital components.
Poor performance under load	High	Enhance system performance and do comprehensive load testing.
Violations of compliance	High	Review and update HR rules and procedures regularly to ensure compliance.
Audit trail lacking necessary information	High	Retain detailed logs of all user actions and system changes.
Devices/Browser Incompatibility	Medium	During development, ensure cross-browser and cross-device compatibility.
Delays in timesheet approval	Medium	Implement timely alerts and provide transparent approval workflows.
User Interface Issues	Low	Do usability testing regularly to find and fix UI bugs. Apply user feedback to make changes.
Test Documentation Errors	Low	Establish strict review procedures for test documentation to identify and fix mistakes at an early stage of the testing cycle. Employ uniform templates.

3.2 LEVEL OF TESTING

Test Type	Description
Functional Testing	**Ensures that the application operates correctly by following the specifications and requirements.**
1. Unit Testing	Validates the software's smallest units or components in isolation, which is often done by developers throughout development.
2. Integration Testing	Verifies the interactions and data flow between connected components to ensure they function properly.
3. System Testing	The entire system is evaluated to ensure that it meets all requirements, including functional and non-functional features.
4. Acceptance Testing	Check to see if the system has met the acceptance criteria and is ready for release.
Regression Testing	**Checks that recent code modifications have not had a negative impact on existing functionality.**
1. Selective Regression Testing	Optimizes test coverage by focusing on specific regions impacted by code changes.
2. Retesting	Rechecks previously discovered and corrected issues to ensure that they have been repaired.
Non-functional Testing	**Focuses on aspects other than functionality, including performance, security, and user experience.**
1. Performance Testing	Determines how the system works under various situations and stress levels.
2. Security Testing	Identifies vulnerabilities and shortcomings in the software's safety features.
3. Compliance Testing	Checks if the program adheres to specific norms or standards.

4. Test Approach

4.1 Test Design Approach

A detailed examination of the HR process requirements is part of the QA plan for testing the HR application to comprehend expected behavior. To assist with test case design, test models and diagrams representing HR workflows and data interactions will be produced.
Test cases that are in line with actual HR procedures will be ensured by collaboration with HR specialists, end users, and domain experts. Realistic HR data will be used for thorough testing, and a variety of scenarios, including edge cases, will be covered by testing methodologies such as analytical, model-based, and consultative.

Test design techniques include:

i. *Boundary Value Analysis:*
   - Test HR portal accessibility with valid credentials, including minimum and maximum input values.

ii. *Equivalence Partitioning:*
   - Group HR input for timesheet actions into valid and invalid categories.

iii. *Decision Table Testing:*
   - Confirm report examination functionality with different access scenarios.

iv. *State Transition Testing:*
   - Validate the timesheet approval process for correct state transitions.

v. *Error Guessing:*
   - Identify issues in various HR portal features using error guessing, considering common error-prone scenarios.

4.2 Execution Strategy

4.3.1 Entry Criteria

- *The entry criteria refer to the desirable conditions to start test execution.*
- *Entry criteria are flexible benchmarks. If they are not met, the test team will assess the risk, identify mitigation actions, and provide a recommendation.*

Entry Criteria	Conditions	Comments
Test environment(s) is available	✔	
Test data is available	✔	
The code has been merged successfully	✔	
Development has completed unit testing	✔	
Test cases and scripts are completed, reviewed, and approved by the Project Team		

3.2.2 Exit criteria

- *The exit criteria are the desirable conditions that need to be met to proceed with the implementation.*
- *Exit criteria are flexible benchmarks. If they are not met, the test team will assess the risk, identify mitigation actions, and provide a recommendation.*

Exit Criteria	Conditions	Comments
100% Test Scripts executed		
90% pass rate of Test Scripts		
No open Critical and High severity defects		
All remaining defects are either cancelled or documented as Change Requests for a future release		
All expected and actual results are captured and documented with the test script		
All test metrics collected based on reports from daily and Weekly Status reports		
All defects logged in -Defect Tracker/Spreadsheet		
Test environment cleanup completed and a new back up of the environment		

3.3 DEFECT MANAGEMENT

- *It is expected that the testers will execute all the scripts in each of the cycles described above.*
- *The defects will be tracked through the Defect Tracker or Spreadsheet.*
- *It is the tester's responsibility to open the defects, retest, and close them.*

Defects found during the Testing should be categorized as below:

Severity	Impact
1 (Critical)	▪ *Functionality is blocked and no testing can proceed.* ▪ *Application/program/feature is unusable in the current state*
2 (High)	▪ *Functionality is not usable and there is no workaround, but testing can proceed*
3 (Medium)	▪ *Functionality issues but there is a workaround for achieving the desired functionality*
4 (Low)	▪ *Unclear error message or cosmetic error which has minimum impact on product use.*

5. Test Team Structure

5.1 TEAM STRUCTURE

#	Role	Resource Count
1	QA Manager	
2	QA Leads	
3	Senior QA Engineers	
4	QA Engineers	

5.2 ROLES AND RESPONSIBILITIES

QA Manager:

i. 1. Specify the goals and general test plan for the HR process testing.
ii. 2. Distribute resources and plan out the test procedures.
iii. 3. Oversee risk assessment and mitigation plans to guarantee a seamless testing procedure.

QA Leads:

i. 1. Create thorough test strategies and plans following the demands of the HR process.
ii. 2. Monitor the creation of test cases and make sure that the testing goals are met.
iii. 3. Prioritize defect management, organize test execution, and promote team communication.

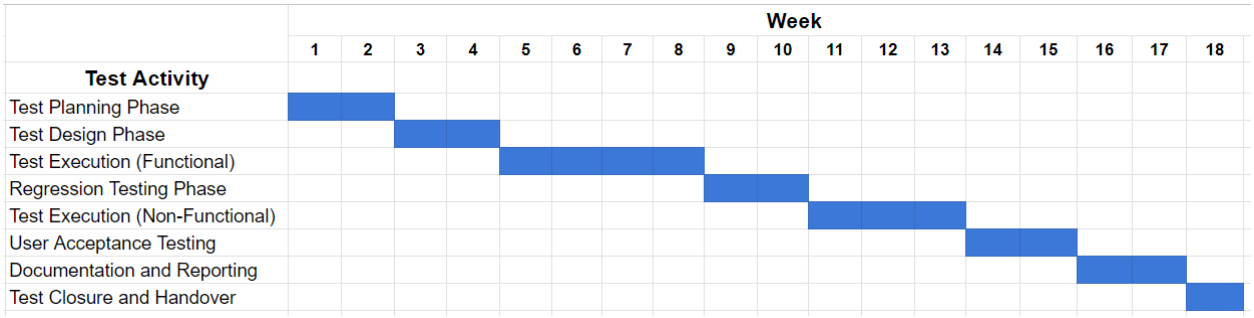
Senior QA Engineers:

i. 1. Based on the requirements, create, and maintain comprehensive test cases.
ii. 2. Carry out test cases, document errors, and automate as necessary.
iii. 3. Control test data and help to increase the effectiveness of testing.

QA Engineers:

i. Carry out test cases following test plans, document errors, and confirm fixes.
ii. Help manage test data while making sure it is pertinent to test cases.
iii. Participate actively in regression testing and keep up-to-date, correct test documentation.

6. Test Schedule



7. Test Reporting

7.1. Quality Matrices

iv. Test priority coverage: Ensure sufficient attention is given to high-priority test cases.
v. Test execution efficiency: Analyze how quickly and efficiently test cases are executed.
vi. Defect closure rate: Track the speed at which discovered vulnerabilities are fixed.
vii. Regression test coverage: Set regression testing coverage as a top priority to quickly identify problems.
viii. Test data relevance: Make that the test data is valid and applicable.
ix. Test environment stability: Evaluate the testing environment's stability and dependability.
x. Traceability of requirements: Verify that there are test cases for each requirement.

8. Test Environment Requirements

i. Software and Browser Compatibility: A stable test environment that mirrors the production setup, supporting common operating systems and web browsers.
ii. Hardware Resources: Sufficient hardware resources for testing, including server infrastructure if applicable.
iii. Data Sets: Access to diverse HR data sets (employee profiles, attendance, leave records) to simulate real-world scenarios.
iv. Security and Privacy: Adherence to data security and privacy regulations to protect sensitive HR data.
v. Testing Tools: Access to relevant testing tools for test case management, automation, performance, and security testing.

9. Dependencies and Assumptions

Dependencies

i. Test Item Availability:
- Access to a representative dataset of HR processes for realistic testing scenarios.
- Availability of required hardware and software configurations.

ii. Project Deadlines:
- Adherence to project timelines to ensure a well-structured QA process and on-time delivery of a reliable HR portal.

iii. Data Security:
- Assurance of data security and privacy compliance during testing, especially with sensitive HR data.

Assumptions

i. Assumption of entry criteria:
- Unit testing, code merging, and system development are complete, test data will be defined using actiTIME's demo data.

ii. Stable Environment:
- The HR login and leave management feature of actiTIME is assumed to be in a stable state for testing.

iii. Resource Availability:
- Assumption that necessary testing resources will be available throughout the testing process.