Quality Assurance Plan

<Project actiTIME>

<2nd November 2023>

1 Introduction

1.1 Purpose

Use the Quality Assurance Plan document to describe the testing strategy and overall approach that will drive the testing of the project.

1.2 Project Overview

actiTIME is a time tracking software for cost effective projects. This software helps to manage a company at different levels such as team members, teams and projects. It also includes a report generation facility to make the decision making process easier.

2 Scope

2.1 In-Scope

Following features and functions should be available to HR

- *Log in to the application with a valid user and a valid password*
- *Visit Employee profiles*
- *Review leaves and attendance reports*
- *Approve or reject Employee timesheets*

2.2 Out-of-Scope

Following features will not be tested.

- *Billing & invoicing*
- *Integrations*
- *Customer care*
- *Task management*
- *Paid Services*

3 Testing Strategy

3.1 Product/Application/Solution Risks

Risks	Criticality	Mitigation Strategy
Immediately saves user profile changes without asking to confirm the changes.	Low	Add a confirmation button to prevent saving accidental changes.
Doesn't have a strong password policy.	High	Enforce a password policy that requires a complex password with special characters, numbers, etc.
Difficult to accept/reject a bulk of timesheets. Have to scroll up the page click the accept/reject button.	Medium	Add a button with functionality to accept/reject each timesheet separately.
In the user profile, the feature to copy scope from another user only shows the number of customers, projects, tasks they are assigned to. Need a more detailed scope before copying scope.	Medium	Provide a more detailed view of the customers, projects and tasks to be assigned before proceeding with the copy.
Insufficient workload management will lead to reduced company productivity and affect the operations.	Critical	Capacity planning, resource allocation.
When creating "Leave Time & Balances Report", the steps in the "Data Group" filter doesn't omit previously applied value from the next filter leading to the user selecting the same value for both.	Low	Develop the UI to eliminate selected filter values from the next filter.

3.2 Level of Testing

Test Type	Description
Unit Testing	Test individual component
Integrate Testing	Test integrated components
System Testing	Test the entire system
Acceptance Testing	Test the final system

3.2.1 Unit Testing

Breaking down a system into small testable portions and test each of them separately to ensure each module functions properly. This helps to detect errors at early stages by testing the newly implemented module.

3.2.2 Integrate Testing

Different individual models are grouped together and tested to ensure proper interactiveness and data flow between the modules.

3.3.3 System Testing

Fully integrated system is tested here to ensure the proper interactions between all modules. This also checks whether the system meets the requirement specification.

3.3.4 Acceptance Testing

Acceptance testing is performed by the end user to ensure the quality of the system and whether the system meets the business requirements. Alpha and beta testing are two types of acceptance testing.

4. Test Approach

4.1 Test Design Approach

Black box design strategy will be followed.
Following test designed techniques will be utilized.

- *Equivalence partitioning -* separate the inputs as valid and invalid partitions and write few test cases for each partition.
- *Decision table -* different input combinations will used to test system response.
- *Error guessing -* guessing error from the experience and intuition and prepare error scenarios and write test cases for each of them.

4.2 Execution Strategy

4.3.1 Entry Criteria

- *The entry criteria refer to the desirable conditions in order 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.*

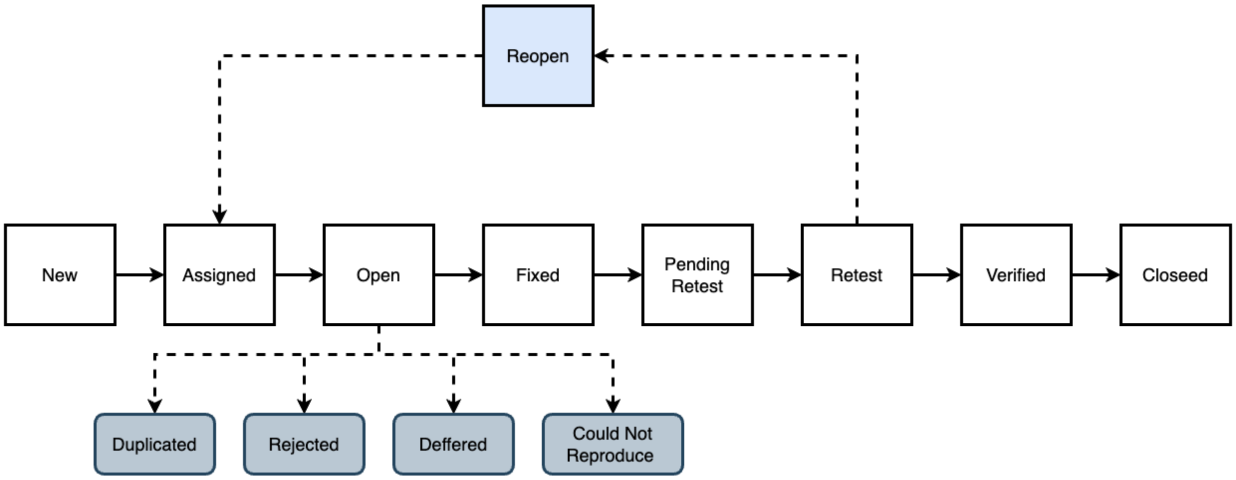
Entry Criteria	Conditions	Comments
Test environment(s) is available	✓	
Test data is available		
Code has been merged successfully		
Development has completed unit testing		
Test cases and scripts are completed, reviewed and approved by the Project Team		
Test environment is available		

3.2.2 Exit criteria

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 within the agreed limit		
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		
Evaluated level of performance, usability, and other attributes are sufficient		

3.3 Defect Management

Defects lifecycle



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	1
2	QA Leads	1
3	Senior QA Engineers	2
4	QA Engineers	4

5.2 Roles and Responsibilities

QA Manager

- Provide strategic direction for the testing process
- Responsible for quality of the system
- Ensure the testing process is carried out aligned with the plan

QA Leads

- Assign responsibilities to QA Engineers
- Work together with QA managers to plan the overall testing process

Senior QA Engineers

- Develop the skill of junior QA Engineers
- Discover test automation opportunities
- Test optimization and assist the test coverage analysis

QA Engineers

- Writing test cases
- Test environment setup
- Main the test documentation

6. Test Schedule

Testing Phase	Start Date	End Date
Test Planning	2023-11-15	2023-11-20
Test Environment Setup	2023-11-21	2023-11-25
Unit Testing	2023-11-26	2023-12-05
Integration Testing	2023-12-06	2023-12-21
System Testing	2023-12-22	2024-01-05
User Acceptance Testing (UAT)	2024-01-06	2024-01-20
Security Testing	2024-01-21	2024-01-25
Test Reporting	2024-01-26	2024-01-31
Test Closure	2024-02-01	2024-02-04

7. Test Reporting

7.1.Test Reporting Approach

#	Report Name	Owner	Audience	Frequency
1	TEST PROGRESS REPORT	Test Lead	Test team, Development team, Project Manager	Daily
2	TEST SUMMARY REPORT	Test Manager	Management team, Project Steering committee, Customer	End of each testing phase

7.2. Quality Matrices

- *Cost for testing should be within the budget*
- *Testing process should be completed within the defined schedule*
- *Test Coverage should achieve 95%*
- *All discovered defect information should be documented*

8. Test Environment Requirements

- *All the tests should be monitored and logged.*
- *Test environment should consist of necessary tools such as automation and management tools.*
- *Test environment should be able to manage several configurations*
- *Test environment should cover wide range of relevant data*

9. Dependencies and Assumptions

Assumptions:

- *Assume the HR members are properly familiarized in using digital devices*
- *Assume that the test environment will be stable for overall testing process*
- *Assume the test environment is properly configured*

Dependencies:

- *User availability*
- *System Documentation*
- *Hardware and software resources*