
Quality Assurance Plan

<ActiTIME>

<3rd October 2023>

1. Introduction

1.1. PURPOSE

Comprehensively presenting the testing strategy and testing approach Human Resources tracking system; ActiTIME.

1.2. PROJECT OVERVIEW

The ActiTIME project

ActiTime application provides facilities to manage human resources in an organisation. The test plan for the following requirement is included in this document.

Requirement: The HR Management application allows HR to log in with a valid username and valid password. The HR once logged into the system successfully can be able to look into Employee profiles and review leaves and attendance reports and Approval/rejection of the timesheet

2. Scope

2.1. IN-SCOPE

- log in with a valid username and valid password.
- Viewing Employee profiles
- Reviewing leaves
- Reviewing attendance reports
- Approving/rejecting of the timesheet

2.2. OUT-OF-SCOPE

- Creating reports
- Creating charts and widgets
- Adding new task
- Adding new users
- Sending bulk invitations
- Managing departments

3. Testing Strategy

3.1. PRODUCT/APPLICATION/SOLUTION RISKS

Risks	Criticality	Mitigation Strategy
Unauthorized access to HR profiles	High	Enforce multi factor authentication.
Weak password risk for profiles	High	Enforcing strict password including length, complexity and regular password change.
Cookie based authentication vulnerability [when user keeps signed in]	Medium	Enable secure session management, use of tokens to authenticate
System failures when handling high volume of data	High	Conduct performance testing
System failures when handling high user load	High	Conduct performance testing
Data privacy risk	High	Enforce access control, Ensure compliance to data regulations
Lack of usability	Medium	Engaging users in usability testings
Difficulty to integrate with other HR tools	Medium	Ensure compatibility through integration testing

3.2. LEVEL OF TESTING

Test Type	Description
Functional Testing	To test the system functionalities, interactions between modules and to ensure the system meets the user requirements
Regression testing	To ensure the system meets user requirements after modifications
Non-functional Testing	To ensure the system meets the quality requested by users in terms of usability, security, reliability etc.

3.2.1 Functional Testing

- Unit testing
- Integration Testing
- User acceptance testing

3.2.2 Regression Testing

- Regression Testing

3.3.3 Non-Functional Testing

Performance Testing

- Load Testing
- Stress Testing
- Usability Testing
- Security Testing

4. Test Approach

4.1 TEST DESIGN APPROACH


Analytical Test strategy can be followed with risk-based testing. The following design techniques are used to design test cases.

Test Design Technique	Description
Decision Table	To build test cases for authentication during login
Equivalent Partitioning & Boundary Value analysis	To enforce password criteria such as length and complexity.
Error Guessing	To identify defects in usability and faults in UIs
State transition	To capture the navigation errors when performing functionalities such as viewing employee profile, viewing reports.

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.*

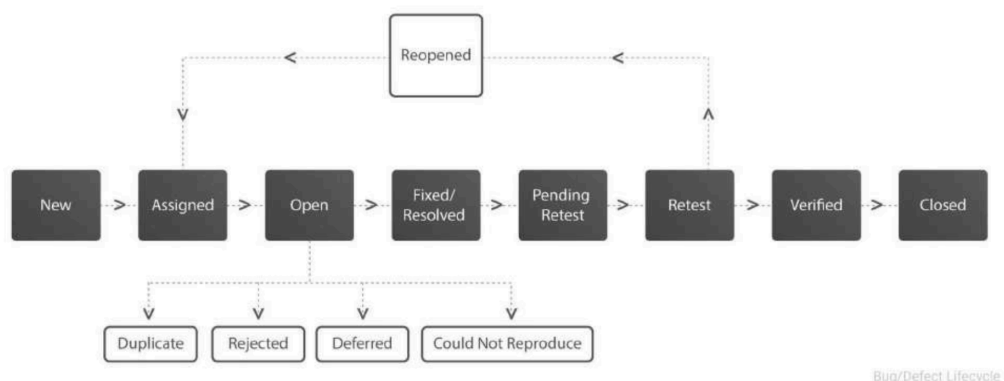
Entry Criteria	Conditions	Comments
<i>Test environment(s) is available</i>		
<i>Test requirements, user stories are defined</i>		
<i>Test data is available</i>		
<i>Code has been merged successfully</i>		
<i>Development has completed unit testing</i>		
<i>Test cases and scripts are completed, reviewed and approved by the Project Team</i>		

3.2.2 Exit criteria

- *The exit criteria are the desirable conditions that need to be met in order 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



Defects Cycle

- It is expected that the testers execute all the scripts in each of the cycles described above.
- The defects will be tracked through 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)	<ul style="list-style-type: none"> ■ <i>Functionality is blocked and no testing can proceed</i> ■ <i>Application/program/feature is unusable in the current</i>
2 (High)	<ul style="list-style-type: none"> ■ <i>Functionality is not usable and there is no workaround but testing can proceed</i>
3 (Medium)	<ul style="list-style-type: none"> ■ <i>Functionality issues but there is a workaround for achieving the desired functionality</i>
4 (Low)	<ul style="list-style-type: none"> ■ <i>Unclear error message or cosmetic error which has minimum impact on product use.</i>

5. Test Team Structure

5.1 TEAM STRUCTURE

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

5.2 ROLES AND RESPONSIBILITIES

QA Manager

The person responsible for project management of testing activities, resources, and evaluation of a test object.

Responsibilities

- High-level documents maintenance
- Estimate, Plan, and Adapt planning
- Introduce metrics
- Make decisions
- Support testing process
- Coordinate Monitoring and test control
- Reporting to management and clients

QA Leads :

Reporting to the test manager and is responsible for project management of a particular test level or a particular set of testing activities.

Responsibilities

- Define test strategies
- Test planing
- Overseeing and leading testing activities of the assigned testing level
- Reviewing test documentation
- Tracking metrics
- Test execution
- Setting and monitoring test closure procedure

Senior QA Engineers /QA Engineers

- Reviewing and contributing to test plans from the tester perspective
- Analyzing, reviewing and assessing the test basis for testability and to detect defects early;
- Identifying and documenting test conditions and test cases
- Designing, setting up and verifying test environments
- Designing and implementing test cases and test procedures, including automated tests where appropriate
- Acquiring and preparing test data to be used in the tests;
- Creating a detailed test execution schedule (for manual tests);
- Executing the tests, evaluating the results and documenting deviations from expected results as defect reports;

6. Test Schedule

Test suite A [unit testing] , Priority: Critical , Due date: November 15 2023

Test suite B [Integrated testing] , Priority: Critical , Due date: November 30 2023

Test suite C [System testing] , Priority: High , Due date: December 05 2023

Test suite D[Performance testing] , Priority: Medium , Due date: December 15 2023

Test suite E [Regression testing] , Priority: low , Due date: December 20 2023

7. Test Reporting

7.1.TEST REPORTING APPROACH

<Add test reporting details here >

#	Report Name	Owner	Audience	Frequency
1	TEST PROGRESS REPORT	QA Manager	Management, Client	Biweekly
2	TEST PROGRESS REPORT	QA Lead	QA Manager	Weekly

7.2. QUALITY MATRICES

- Percentage of planned work done in test case preparation .
- Total number of test cases run by the QA team.
- The total number of test cases passed/blocked/failed in each phase.
- Total number of bugs detected/accepted/rejected in each phase.
- Test effort.
- Test effectiveness.
- Test coverage of requirements, user stories, acceptance criteria, risks, or code .
- Cost of testing.

8. Test Environment Requirements

Hardware requirements

- Devices and servers with sufficient capacity to run tests.
- Number of devices that can meet the need of QA team
- Ex: computers with required RAM capacity, Web servers etc.

Software requirements

- Operating system, applications, libraries, and drivers required for test execution
- Ex: Android or iOS simulators for mobile testing, IDEs, specific web browsers and drivers.

Network requirements

- Network with sufficient bandwidth and enforced security

Data requirements,

- Defining conditions for data quality.
- Identifying data volume
- Define test data for tests

Tools requirements

- Test management tools such as Jira, Git
- Test automation tools such as Cypress, Selenium

9. Dependencies and Assumptions

- Scope is determined by considering only the given requirement.
- Assuming the human resource availability is a team consist of 10 members including 1 QA manager for the project and, 1 QA lead and 2 QA engineers per each test level.
- Assuming the deadline to deliver the project is December 30 , 2023, the test schedule is planned to complete on December 20 , 2023.