
Quality Assurance Plan

Actitime - Demo

4th November 2023

Heshan Weerasinghe

19020899

1 Introduction

1.1 PURPOSE

This Quality Assurance Plan describes the testing strategy and overall approach that will drive the validation of the HR Management application hosted at <https://demo.actitime.com/>.

The plan is designed to ensure that the application meets its functional requirements reliably and efficiently, providing HR personnel with the ability to manage employee profiles, review leave and attendance reports, and approve or reject timesheets.

1.2 PROJECT OVERVIEW

The project entails examining the HR Management app to ensure its functionalities operate smoothly, delivering an intuitive and streamlined experience for HR personnel. It must adeptly manage confidential employee data, enabling HR managers to carry out tasks accurately and effortlessly. Key functions encompass secure login, employee profile administration, monitoring leave and attendance, and endorsing timesheets.

2 Scope

2.1 IN-SCOPE

1. Ensuring the secure login's validity through credential verification.
2. Affirming access control measures: solely authorized HR personnel permitted to view and manage employee profiles.
3. Checking the employee profile section for both completeness and editing functionalities.
4. Examining leave and attendance reports, encompassing filtering, sorting, and export functions.
5. Reviewing the timesheet submission process, along with the subsequent approval/rejection functionalities.

2.2 OUT-OF-SCOPE

1. Penetration testing and in-depth security analysis are not part of this phase; they demand specialized security testing tools and expertise.
2. The demo application does not cover localization and internationalization testing, aligning with the current requirement for English language support exclusively.

3. Performance testing under high load is beyond the project's scope due to testing environment limitations, which fail to emulate the production scale.
4. These exclusions from testing stem from a current prioritization and resource allocation decision. Additionally, the demo application's scope may not comprehensively mirror all production features.

3 Testing Strategy

3.1 PRODUCT/APPLICATION/SOLUTION RISKS

Risks	Criticality	Mitigation Strategy
Inaccurate leave and attendance reports	High	Develop thorough test cases for all reporting features. Use automated testing to regularly check report accuracy.
System unavailability or downtime	High	Use redundant systems and perform regular backups. Implement a disaster recovery plan. Monitor system health proactively.
Performance issues under peak loads	Medium	Conduct stress testing to determine system's performance under peak loads. Optimize code and infrastructure based on the findings.
Regression bugs due to new updates	Medium	Implement a comprehensive suite of regression tests to be run before each release. Automate regression tests where possible.

3.2 LEVEL OF TESTING

Test Type	Description
Functional Testing	Verifies that each function of the software application operates in conformance with the requirement specification. This is a type of black-box testing.
Non-Functional Testing	Tests the non-functional aspects of the application, such as usability, reliability, and performance.
Compatibility Testing	Ensures the software is compatible with different computing environments, such as web browsers, databases, other software, hardware, and mobile devices.
Usability Testing	Assesses how easy the user interface is to use, including navigation, flow, clarity, and overall user experience.
Security Testing	Assesses the application's security features to ensure that data and resources are protected from unauthorized access and breaches.
Performance Testing	Evaluates how the system performs in terms of responsiveness and stability under a particular workload. It includes both load and stress testing.
Regression Testing	Checks for new bugs in existing areas of a system after changes (like enhancements or bug fixes) have been introduced. This helps ensure the new code does not disrupt existing functionality.
Acceptance Testing	Done to determine if the requirements of a specification or contract are met as intended. It usually involves running a suite of tests on the completed system.

3.2.1 FUNCTIONAL TESTING

Functional Testing: Ensures that the HR Management application performs according to its specifications. It involves validating functions such as secure login, employee profile management, and the review and approval/rejection of timesheets. Applied here, it verifies that each feature of the HR software works as intended, allowing HR staff seamless access to employee profiles and vital reports.

3.2.2 Non-Functional Testing

Focuses on aspects beyond individual functions, such as performance, scalability, and reliability. In the context of HR software, this type of testing assesses how well the application handles multiple users accessing employee data simultaneously, ensuring it remains responsive and reliable under various conditions.

3.3.3 Regression Testing

Ensures the HR Management application functions consistently across different devices, browsers, and operating systems. In this case, compatibility testing would guarantee that HR staff can access and utilize the software seamlessly, regardless of the device or browser they are using.

3.3.4 Usability Testing

Evaluates the application's user interface and overall user experience. In the HR context, usability testing ensures that HR personnel can easily navigate through employee profiles, leave, and attendance reports, promoting efficient and user-friendly interactions with the software.

3.5.5 Security Testing

Focuses on identifying vulnerabilities and ensuring the protection of sensitive data. For HR software, security testing would validate that the login credentials are securely handled, preventing unauthorized access and safeguarding employee information from potential breaches.

3.5.6 Performance Testing

Measures the application's responsiveness, speed, and overall performance. Applied to HR software, performance testing ensures that the system can efficiently handle tasks like reviewing attendance reports and processing timesheet approvals, even under high user loads.

3.5.7. Regression Testing

Verifies that new updates or changes do not negatively impact existing functionalities. In the HR Management application, regression testing ensures that modifications or additions to the software do not interfere with core features such as secure login and employee profile management.

3.5.8 Acceptance Testing:

Validates whether the HR software meets specified requirements and is ready for deployment. In this scenario, acceptance testing ensures that HR personnel can successfully log in, access employee profiles, review leave and attendance reports, and approve/reject timesheets, meeting the criteria for system acceptance.

Test Approach

4.1 TEST DESIGN APPROACH

The chosen test design strategy for validating the HR Management application aligns with the Analytical and Model-Based approaches, emphasizing a systematic and standards-compliant methodology. This approach ensures a comprehensive understanding of the testing process at both the product and organizational levels.

Test Design Techniques:

1. Metrics-Based Technique:

- Burndown Chart: Effort estimation and reporting will be facilitated through a Burndown Chart, capturing and utilizing metrics to inform the team's velocity for subsequent iterations.
- Defect Removal Models: Historical defect volumes and time-to-resolution data will serve as metrics, providing a basis for estimating the test effort for future projects.

2. Expert-Based Technique:

- Planning Poker: Team members will leverage their experience to estimate the effort required for feature delivery through collaborative estimation sessions.

- Wideband Delphi: Groups of experts will provide estimates based on collective experience, ensuring a comprehensive and expert-driven evaluation of testing efforts.

Test Execution Schedule

The testing process is organized into four test suites—A, B, C, and D—each prioritized based on criticality and assigned specific due dates. The schedule prioritizes critical test suites, such as A and D, ensuring their completion before addressing medium and low priority suites, B and C.

Prioritization

Test suites are prioritized according to criticality, with due dates assigned.

Dependencies

Test execution order is influenced by the priority of each test suite, ensuring that critical and high-priority suites are addressed first, minimizing potential delays due to dependencies.

Confirmation Tests

Critical attention is given to confirmation tests for Test Suite A, validating that critical functionalities meet specified requirements and standards.

Regression Tests






Regression testing is embedded in the strategy, systematically addressing potential impacts of changes on previously validated functionalities.

This test design strategy combines analytical rigor, model-based planning, and a blend of metrics and expert-based techniques to ensure a thorough and efficient validation process for the HR Management application.

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>		The test environment closely mimics the productions setup for accurate results
<i>Test data is available</i>		Test data includes a variety of scenarios, including edge cases for comprehensive testing
<i>Code has been merged successfully</i>		Code integration has been completed with no merge conflicts.
<i>Development has completed unit testing</i>		Unit tests have been passed, ensuring that the basic building blocks of the application are stable.
<i>Test cases and scripts are completed, reviewed and approved by the Project Team</i>		All test cases and scripts have been created based on the test strategy and reviewed for quality and completeness.

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
<i>100% Test Scripts executed</i>		All test scripts have been run to ensure complete coverage of the application's features
<i>90% pass rate of Test Scripts</i>		This high pass rate indicates a stable application; issues found are documented for fixing.
<i>No open Critical and High severity defects</i>		Critical and high severity defects have been addressed to ensure they don't impact production.
<i>All remaining defects are either cancelled or documented as Change Requests for a future release</i>		Non-critical defects are tracked for future prioritization, ensuring the application's continuous improvement.

<i>All expected and actual results are captured and documented with the test script</i>		Ensures traceability of tests and supports quality control measures.
<i>All test metrics collected based on reports from daily and Weekly Status reports</i>		Facilitates ongoing monitoring and management of the testing processes and outcomes.
<i>All defects logged in -Defect Tracker/Spreadsheet</i>		Defect logging is crucial for traceability, accountability, and future reference.
<i>Test environment cleanup completed and a new back up of the environment</i>		Critical for maintaining a clean state for future testing and rollback scenarios.

3.3 DEFECT MANAGEMENT

Defect Life-Cycle

New

Upon the identification of an issue during the execution of HR software test scripts, testers log a new defect in the Defect Tracker or Spreadsheet.

Assigned

The newly identified defect is then assigned to the relevant developer or the team responsible for the specific HR application module where the flaw emerged.

Open

The developer acknowledges the defect and initiates the resolution process, actively working on providing a fix for the identified issue.

Fixed/Resolved

After resolving the defect, the developer marks it as 'Fixed/Resolved,' indicating their confidence in the applied solution.

Pending Retest

The defect enters a 'Pending Retest' status, awaiting retesting by the testing team to ensure the efficacy of the provided fix.

Retest

Testers conduct a thorough retest by re-executing related HR software test scripts to confirm the defect's resolution.

Verified

If the retest yields success, the defect is marked as 'Verified,' signifying its satisfactory resolution.

Closed

Verified defects are then officially closed, contributing to the ongoing stability and quality assurance of the HR software.

Special Cases within the Cycle:

Reopened

In the event that the issue persists post-retesting, the defect is reopened and sent back to the developer for further attention within the HR software context.

Duplicate

Identical defects are marked as duplicates, streamlining the defect management process for efficiency.

Rejected

If a defect is deemed invalid or irrelevant, it is promptly rejected, maintaining the focus on genuine software issues.

Deferred

Non-critical defects, considered non-urgent for the current software release, can be intentionally deferred to align with project priorities.

Could Not Reproduce

Instances where the reported issue cannot be replicated are appropriately marked to communicate the challenge of reproducing the defect.

Expectations

Testers are entrusted with executing all test scripts and overseeing the defect lifecycle from identification through to closure.

Retesting and subsequent defect closure are integral aspects of the tester's role, ensuring the sustained stability and quality of the HR software.

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 state</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	2
3	Senior QA Engineers	4
4	QA Engineers	10

5.2 ROLES AND RESPONSIBILITIES

QA Manager:

- Role: Oversees the entire testing process and ensures alignment with project goals.
- Responsibilities:
 - Develops test strategies and plans.
 - Manages resource allocation and team coordination.
 - Provides strategic guidance for quality assurance.
 - Liaises with project stakeholders to communicate testing progress.

QA Leads:

- Role: Supervises and guides QA Engineers in the testing process.
- Responsibilities:
 - Develops detailed test plans based on project requirements.
 - Assigns tasks to QA Engineers and monitors progress.
 - Collaborates with QA Manager to implement testing strategies.
 - Conducts regular team meetings for updates and issue resolution.

Senior QA Engineers:

- Role: Experienced members focusing on complex testing tasks.
- Responsibilities:
 - Designs and executes test cases for critical functionalities.

- Identifies potential risks and suggests mitigation strategies.
- Reviews test scripts and results provided by QA Engineers.
- Assists in mentoring and training junior team members.

QA Engineers:

- Role: Execute testing tasks as per the defined test plans.
- Responsibilities:
 - Develops and executes test cases based on project requirements.
 - Logs and tracks defects using the designated tools.
 - Collaborates with the team to ensure comprehensive test coverage.
 - Provides regular updates on testing progress and issues encountered.

6. Test Schedule

Test Suites

Test Suite - Authentication and User Access Control

Test Suite - Employee Profile Management

Test Suite - Leave and Attendance Tracking

Test Suite - Timesheet Approvals

Prioritization

Test Suite - Authentication and User Access Control: Critical

Test Suite - Employee Profile Management: Medium

Test Suite - Leave and Attendance Tracking: Low

Test Suite - Timesheet Approvals: High

Dependencies

The execution of Test Suite A (Authentication and User Access Control) is a prerequisite for Test Suite D (Timesheet Approvals).

No other inter-suite dependencies exist.

Confirmation Tests

Critical attention will be given to confirmation tests within Test Suite A (Authentication and User Access Control) to ensure secure login and user access control functionalities meet specified requirements.

Regression Tests

Regression testing will be applied to all test suites to systematically address potential impacts of changes on previously validated functionalities.

Test Execution Schedule

Test Suite - Authentication and User Access Control

Priority: Critical

Due Date: 19 Nov

Test Suite - Timesheet Approvals

Priority: High

Due Date: 20 Nov

Test Suite - Employee Profile Management

Priority: Medium

Due Date: 21 Nov

Test Suite - Leave and Attendance Tracking

Priority: Low

Due Date: 22 Nov

7. Test Reporting

7.1.TEST REPORTING APPROACH

#	Report Name	Owner	Audience	Frequency
1	TEST PROGRESS REPORT	Senior QA Engineer	QA Manager, Development Team	Weekly
2	TEST PROGRESS REPORT	QA Manager	Project Stakeholders	Monthly

7.2. QUALITY MATRICES

In assessing the quality of our HR software, key matrices will be employed to measure the effectiveness of our testing efforts. One crucial metric is the Defect Density Matrix, quantifying the number of defects per unit of code. This provides insights into the software's robustness.

Additionally, the Test Coverage Matrix will be pivotal, illustrating the percentage of the application covered by our test cases. This ensures a comprehensive evaluation of critical functionalities. Lastly, the Execution Efficiency Matrix will measure the efficiency of our test execution process, helping optimize resource utilization.

8. Test Environment Requirements

It's essential to verify the HR software's compatibility with diverse operating systems, databases, and browsers to cater to a wide user base. Additionally, assessing performance under different network conditions ensures the software's resilience in real-world scenarios, and scalability testing guarantees it can handle varying loads efficiently.

- Operating Systems Compatibility:
 - Ensure compatibility with Windows, Linux, and macOS to accommodate diverse user environments.
- Database Compatibility:
 - Verify seamless integration with Oracle, MySQL, and SQL Server databases for robust data management.
- Browser Compatibility:
 - Test across major browsers (Chrome, Firefox, Safari, Edge) to guarantee a consistent user experience.
- Network Configurations:
 - Assess performance under varying network conditions (low bandwidth, high latency) for real-world scenarios.
- Scalability Testing:
 - Evaluate system performance under varying loads to ensure scalability and resource optimization.

9. Dependencies and Assumptions

Test Item Availability

- Dependency on the timely provision of HR software builds for testing activities.

Testing Resource Availability

- Assumption that the required testing tools and environments will be accessible when needed.

Deadline Alignment

- Dependency on project timelines to synchronize testing phases with broader development milestones.

Assumptions

Build Consistency

Assumes a consistent release of stable software builds for effective testing.

Resource Availability

Assumes the availability of skilled testing resources in alignment with project needs.