

**CSE300: Software Engineering**

**Monsoon Semester 2022**

**Group 7**

**Group Members**

| **Name** | **Enrolment Number** | **Email-id** |
| --- | --- | --- |
| Sakshi Shah | AU1940213 | sakshi.s5@ahduni.edu.in |
| Astha Patel | AU1940312 | astha.p@ahduni.edu.in |
| Kareena Matwani | AU1940314 | kareena.m@ahduni.edu.in |
| Vraj Parikh | AU1940185 | vraj.p1@ahduni.edu.in |
| Rushil Borad | AU1940179 | rushil.b@ahduni.edu.in |

**Project Topic: Workspace - Task Management System**

**Test Plan Document**

# 

[**1. References**](#_eql03u7y1nc1) **3**

[**2. Project Background**](#_r26w6roav9wn) **3**

[**3. Introduction**](#_feho5mokhln9) **3**

[**4. Purpose**](#_9jlilppzkftx) **4**

[**5. Scope**](#_bs8215ucyatx) **4**

[**6. Testing Approach**](#_7m4clh4tnf92) **7**

[**7. Entry and Exit Criteria**](#_bbtirl5837zd) **7**

[7.1 Entry Criteria](#_82s40sceel4k) 7

[7.2 Exit Criteria](#_75sbsecgpc6g) 8

[**8. Suspension Criteria & Resumption Requirements**](#_10w3espcolum) **8**

[8.1 Suspension Criteria](#_w8qt9rvxd7nf) 8

[8.2 Resumption Criteria](#_l83ucwbsxc80) 9

[**9. Test Deliverables**](#_nm970fd5kq79) **9**

[**10. Testing Strategy**](#_2g3b12xdqrzp) **9**

[10.1 Test Process](#_cyq8xstf5y9z) 9

[10.1.1 Understanding Requirements](#_pamctfmoyw0k) 9

[10.1.2 Develop Tests](#_mggjlse01k9c) 10

[10.1.3 Preparing Test Matrix](#_olrzqud0bh3k) 10

[10.1.4 Reviewing test cases and matrix](#_jh0tjvvi0hwz) 10

[10.1.5 Prepare to test](#_96a02g32hbjf) 11

[10.1.6 Run Tests](#_qew911ttuejl) 11

[10.1.7 Review Test Results](#_mbrsfmizcpuq) 11

[10.2 Testing Types](#_tgpf3vnlnz1h) 11

[10.2.1 Unit Testing](#_bvp4su77ki1u) 11

[10.2.2 Functional Testing](#_hcvitmf7gyc0) 12

[10.3 Resource and Environment Needs](#_t120kmtjv1x5) 12

[10.3.1 Testing Tool](#_uze5uitd6lnt) 12

[10.3.2 Test Environment](#_kxqzcdmys0s9) 13

[10.4 Testing Schedule](#_xpz52yvej8ik) 13

[**11. Risk, Assumptions and Dependencies**](#_narfh5rnmcjp) **15**

[11.1 Risks](#_9t0v4my05rfp) 15

[11.2 Assumptions](#_2b3bhqtisrs8) 17

[11.3 Dependencies](#_86zqnbab7boc) 18

[**12. Glossary**](#_foc726buv8eq) **19**

# **1. References**

**The list of documents for forming this test plan are:**

* **Software Requirements Specification(SRS)**
* **Design Document**

# **2. Project Background**

The **Task management system “WORKSPACE”** is a web application that helps students and employees to manage their daily work and teamwork in a more organized and simpler way. This application should be accessible to any browser on the desktop. This software includes important features like authentication to the portal, authorization to the data as per requirement in the teams or organization, note taking (in collaboration also), calendar updates after events/ tasks are managed, team rooms, chat rooms, and task list boards in each team room. The objective of the collaborative Task Management System is to manage tasks smoothly from one platform instead of using a collection of tools and services. It would be a one-stop platform for task and team organization with ease, that includes various management tools like event management, managing teams, team notes, task assignments, and more. This would ease the process of sharing and maintaining tasks efficiently. The system should be user appropriate, easy to use, provide easy recovery of errors, and have an overall end-use higher satisfaction.

# **3. Introduction**

The main purpose of the test plan document for the Task Management System (Task Workspace) project is to discuss the testing details of the use cases of the Task Management System. The Objective of Test plan is to define the various **Testing strategies and testing tools used for complete Testing life cycle** of this project. This Test Plan is designed to prescribe the scope, approach, resources, deliverables, environment and schedule of all testing activities of the Task Management System project. The plan identifies the items to be tested, the features to be tested, the types of testing to be performed, the personnel responsible for testing, the resources and schedule required to complete testing, and the risks associated with the plan. This document will address the different standards that will apply to the unit, functional and manual testing of the specified application. The design, development and testing of these reports will be based on the Task Management System project. Throughout the testing process we will be applying the standard test documentation specifications.

# **4. Purpose**

The main purpose of the test plan for the Task Management System Project is as follows:

* To identify the features of the system that will be tested.
* To identify and define all the activities necessary to prepare for and conduct the testing process on the Task Management System.
* To define the pass/fail criteria for each item that will be tested.
* To identify the deliverables of the testing phase.
* To discuss the testing techniques being used to test the System.

# **5. Scope**

Functions to be tested are as follows:

| Module | Applicable Roles | Description |
| --- | --- | --- |
| Login | Admin, User | User or admin should be able to login in the system |
| Dashboard | Admin, User | Admin and user must be redirected to the dashboard after login where all details regarding the project process should be visible. |
| Project List | Admin, User | An admin or user must be able to view the list of projects by clicking on the project → list |
| View/Delete/Update Project | Admin | An admin must be able to view / delete/ update project details from the action option |
| Add New Project | Admin | An admin must be able to add new project from the project → add new option |
| Task List | User | A user must be able to view the task assigned or their details by clicking on the Task→List option |
| Add Task | Admin | Admin must be able to add new task by clicking on the action → add new productivity button |
| Report | Admin, User | An admin or user must be able to view and print the project progress report by clicking on the report option from navbar |
| Calendar | Admin, User | Admin and user must be redirected to the calendar section. |
| Add, update and delete events. | Admin | Admin must be able to add, update and delete events from the calendar section. |
| User | Admin, User | Admin and user must be redirected to the user section by clicking on the user option from navbar. |
| User list | Admin | Admin must be able to view the list of users from the user→list option. |
| View, edit and delete user | Admin | Admin must be able to view , edit and delete user details from the action option. |
| Add new user | Admin | Admin must be able to add new users from the user → Add new option. |
| Update profile | User, Admin | A user or admin must be able to update their profile. |

# **6. Testing Approach**

The approach used, in accordance with requirements-based strategy, where an analysis of the requirements specification forms the basis for planning, estimating and designing tests. Test cases will be created during exploratory testing. All test types are determined in Test Strategy. The project is using an agile approach, with weekly iterations. This section of the test plan describes the overall approach for testing the Task Management System project. The approach followed for testing the Task Management System ensures that the major features of the project are adequately tested. The testing would be carried out on the Task Management System while logging into the system as a User or as an Admin.

# **7. Entry and Exit Criteria**

## **7.1 Entry Criteria**

Entry criteria for testing can be defined as specific conditions that must be met before a process can begin. The required Entry Criteria is specified by The Software Testing Life Cycle during each testing phase. The inputs must be met by the Development Phase and Test Phase.

The requirements needed to be fulfilled for the entry criteria from the testing phase include:

* Appropriately Defined and Approved Requirements
* Availability of complete or partially testable code
* Test Plan
* Access to sufficient and desired test data.
* The readiness of test cases
* Appropriate Test Environment with all the necessary resources like test tools and devices.

## **7.2 Exit Criteria**

Exit criteria in testing are often viewed as a single document commemorating the end of a life cycle phase. It can be defined as “The specific conditions or on-going activities that should be fulfilled before completing the software testing life cycle. STLC specifies which exit criteria is required at each testing phase”. The exit criteria can identify the intermediate deliverables and enable you to track them as independent events.

The following exit criteria should be considered for completion of a testing phase:

* Ensuring all critical Test Cases are passed
* Meets the desired and sufficient coverage requirements and functionalities.
* No critical bug has been left out during the testing process
* Identifying and fixing all the high-priority defects
* Re-testing and closing all the high-priority defects to execute corresponding Regression scenarios successfully
* Test Logs generated
* Test Summary report generated

# **8. Suspension Criteria & Resumption Requirements**

## **8.1 Suspension Criteria**

Suspension criteria is a set of conditions which when satisfied, the test team temporarily suspends the testing process. If any defects are found which seriously impact the test progress the test lead may choose to suspend testing.

The criteria which are considered for suspension or resumption are :

* Hardware / software not available at the time indicated in the project schedule
* The build contains many serious defects which seriously prevent or limit testing progress
* Assigned test resources are not available when needed by the test team
* If your team members report that there are 40% of test cases failed, you should suspend testing until the development team fixes all the failed cases.

## **8.2 Resumption Criteria**

If testing is suspended, resumption will only occur when the problem(s) that caused the suspension have been resolved. When a critical defect is the cause of the suspension, the “FIX” must be verified by the testing team before testing is resumed.

# **9. Test Deliverables**

The following test documentation will be produced:

* Test Plan - This document deals with what needs to be done in UAT(User Acceptance Testing).
* Test Cases - The values input and results expected from tests.
* Unit Test Cases
* Functional Test Cases
* Requirement Traceability Matrix

# **10. Testing Strategy**

## **10.1 Test Process**

The software tester checks whether the product behaves as expected during the testing process. They find the errors in the test stage and try to solve them. In this way, the final product becomes free of bugs or includes a minimal number of errors. We can say that the software test improves the quality of the software. Software testing is not just a stage at the end of the software life cycle. Software testing is an iterative process and continues after the software is completed.

### **10.1.1 Understanding Requirements**

* Requirement specifications taken from stakeholders.
* Understanding of requirements: Means understanding them and looking for what is missing and inconsistent from what is actually required.

### **10.1.2 Develop Tests**

**Derive Acceptance criteria:** Acceptance criteria are a formal list that fully enumerates user requirements and all the product scenarios put into the account. Acceptance criteria plainly describe conditions under which the user requirements are desired, thus getting rid of any uncertainty of the client’s expectations and misunderstandings. The set questions must be prepared to check if the requirements needed are prepared.

**Construct Test Cases:** Test Cases are the set of specific inputs and expected results which enable one or more Acceptance Criteria to be proved.

### **10.1.3 Preparing Test Matrix**

A matrix is a concise organizer of simple tests, especially useful for function tests and domain tests. It groups test cases that are essentially the same.

To create a test matrix, you will have to:

* Put the objects that you’re testing on the rows.
* Show the tests on the columns.
* Check off the tests that you actually completed in the cells.

### **10.1.4 Reviewing test cases and matrix**

Test case ensures that each and every functionality mentioned in Software Requirement Specification is covered. Test cases should be effective and also follow the standards to write test cases. To succeed and completeness of any test cases every test case should be reviewed. There are different types of test case review processes.

**Self-review:** It is done by the tester himself who has written the test cases. He can verify whether all the requirements are covered or not by looking into SRS/FRD.(Software requirement specification)

**Peer review:** It is done by another tester who hasn’t written those test cases but is familiar with the system under test. Also known as Maker and Checker review.

**Review by a supervisor:** It is done by a team lead or manager who is superior to the tester who has written the test cases and has great knowledge about the requirements and system under test.

### **10.1.5 Prepare to test**

* Preparing the environment to run the tests – Making sure that the people, processes, hardware, software etc. are all in place to enable the testing to take place.
* Preparing Test Data - Building the data files that are required to run the test cases.

### **10.1.6 Run Tests**

* **Running the tests** involves using the input and expected results from the Test Cases and applying the Test Scripts and other elements of the Test Procedure to run them.
* **Recording the results i**nvolves recording in the Test Log the activities that were done in what order, and the events that happened when the test was run. Any that have actual results that differ from the expected results have the information recorded in an Incident Report. The Incident Severity is also decided at this point.

### **10.1.7 Review Test Results**

When the tests have been completed then the acceptability of the system is assessed. A simple method is to check how many outstanding Incidents there are and their severity. However this is not sufficient as a simple count of Incidents does not give any idea about their impact on what the organization wants to achieve with the system. A flawed system which delivers capability to an organization is much better than a perfect system that does not. Therefore the test results need to be checked and traced to see what effect they have on:

* Scenarios, Requirements and their Business or System Impact

## **10.2 Testing Types**

### **10.2.1 Unit Testing**

| Test Objective | To test individual modules to determine if there are any issues. It is concerned with functional correctness of the standalone modules. |
| --- | --- |
| Technique | Manual |
| Completion Criteria | If a module passes all the Test Cases written for it, then it has Passed the test. |

### **10.2.2 Functional Testing**

| Test Objective | To test whether the function is working according to the requirement specification. |
| --- | --- |
| Technique | Manual |
| Completion Criteria | If a module is working according to the requirement specification. |

## **10.3 Resource and Environment Needs**

### **10.3.1 Testing Tool**

| **Process** | **Tool** |
| --- | --- |
| Test Case creation | Microsoft Excel |
| Test Case tracking | Microsoft Excel |
| Test Case execution | Manual |
| Test Case management | Microsoft Excel |

### **10.3.2 Test Environment**

* Windows 10: Chrome (latest), Firefox (latest), Edge (latest)
* MAC OS X: Chrome (latest), Safari (latest), Edge (latest)

## **10.4 Testing Schedule**

| **Task Name** | **Date** | **Comments** |
| --- | --- | --- |
| Review Requirements  documents | 10/11/2022 | The requirements gathered from  various stakeholders were reviewed. |
| Test Planning | 11/11/2022 | Decided regarding the approach we  are going to follow during the testing  phase |
| Develop test cases | 12/11/2022 | Creating the test cases manually |
| Create Test Case Matrix | 12/11/2022 | Formulated the test cases in matrix form for easy and better understanding. |
| Review Test cases and  matrix | 16/11/2022 | Reviewed whether the test cases cover all the requirements. |
| Unit Testing | 17/11/2022 | To test the functioning of individual modules |
| Review Testing Result | 21/11/2022 | Check the results (pass/fail) of the test cases and change code accordingly |
| Functional testing | 21/11/2022 | To test rest of the modules  functionality using appropriate test  cases. |
| Review Testing Result | 22/11/2022 | Check the results (pass/fail) of the test cases and change code accordingly |
| Resolution of final errors | 26/11/2022 | To check the status of all the previously addressed errors after improvement. |

# **11. Risk, Assumptions and Dependencies**

## **11.1 Risks**

| **Number** | **Description** | **Status** | **Impact (I)** | **Probability (P)** | **Severity (I\*P)** | **Mitigation Type** | **Detail of mitigating action to be taken** | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | The classes in the database might not be properly linked | Closed | 4 | 1 | Medium | Avoidance | Testing of the relations in the database | | |
| 2 | Continuously changing requirements | Work in Progress | 3 | 5 | High | Acceptance | Continuously adapting to the changing  requirements | | |
| 3 | Evolving technology | Open | 3 | 1 | Low | Acceptance | Make the software as compatible as possible | | |
| 4 | Developing the wrong features | Closed | 5 | 3 | High | Reduction | Reducing and avoiding the wrong features by feedbacks | | |
| 5 | Prerequisite entry criteria is not met for testing | Closed | 2 | 3 | Medium | Avoidance | Tester will define the prerequisites that must be met before Testing can start. | | |
| 6 | Test data proves to be inadequate | Closed | 2 | 2 | Medium | Reduction | Tester will indicate what is required and will verify suitability of test data. | | |
| 7 | Lack of personal resources when testing is to begin | Closed | 2 | 2 | Medium | Avoidance | Test leader ensures the constant supply of resources for testing | | |
| 8 | Delays in training the application | Closed | 3 | 4 | High | Reduction | Developers in team communicate and find ways to  modularize work for faster delivery | | |
| 9 | Lack of knowledge of tasks in team members | Closed | 3 | 2 | Medium | Avoidance | Make sure the team members have appropriate knowledge of their domain | | |

## **11.2 Assumptions**

| **Number** | **Description** | **Status** | **Reason for Assumption** | | **Action to Validate** | **Impact if Assumption is Incorrect** | |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | User has active Internet connection | Closed | The server can only be accessed with internet connection | | The system is hosted on a server | The user cannot access the system | |
| 2 | User can navigate to the software | Closed | The user will be able to see their projects and tasks only when they navigate to the website | | The user is able to navigate | The user will not be able to view the projects and tasks assigned to them. | |
| 3 | Sufficient memory is available for the system to function | Closed | The system will not function if sufficient memory is not available | | The user has appropriate storage space | The system will not work properly | |
| 4 | The addition and deletion of the tasks and project need to be done offline | Closed | No add and delete functionality available in the system | | The user communicate with the authority offline after viewing the allotted tasks and projects | The tasks and projects can be missed by the user if wrongly allotted. | |
| 5 | The addition of events/event registration process is assume to be done offline | Closed | No event / calendar functionality available in the system | | The user communicate with the organizer offline after viewing the respective event | The events will stall and will not function properly | |

## **11.3 Dependencies**

| **Number** | **Description** | **Status** | **Priority** |
| --- | --- | --- | --- |
| 1 | The database is dependent on SQL servers | Closed | Medium |
| 2 | The system is dependent on internet based servers for its web based features to be displayed over the website | Closed | Medium |

# **12. Glossary**

This document is a detailed report that describes the test strategy, objectives, schedule, estimation and deliverables and resources required for testing. Test Plan helps us determine the effort needed to validate the quality of the application under test. The test plan serves as a blueprint to conduct software testing activities as a defined process which is minutely monitored and controlled by the test manager. It helps people outside the test team such as developers, business managers, customers understand the details of testing. It also guides our thinking. Important aspects like test estimation, test scope, Test Strategy are documented in Test Plan, so it can be reviewed by Management Team and re-used for other projects. It lists all the prerequisites, entry and exit level criterias, tools and resources required, process to set up the test environment, testing schedule and test deliverables.