

# **CSE300 Software Engineering**

# **Learning Management System**

**Group 8**

# **Test Plan Document**

# 

[**References**](#_9ms2ghl5mfok) **4**

[**Project Background**](#_wyfwwqq0l4r) **4**

[**Introduction**](#_59mqpdkoy4e6) **4**

[**Purpose**](#_2l00sdyvt1ne) **5**

[**Scope**](#_vqip34xsc2sx) **6**

[Functions to be tested](#_t3rw3vohbkeh) 6

[Functions Not to be tested](#_smh3xerh1350) 9

[**Testing Approach**](#_z2bd6hpjmhh1) **9**

[**Entry and Exit Criteria**](#_fxe75nzcnpnj) **9**

[Entry Criteria](#_pyt48qi6pggo) 9

[Exit Criteria](#_8x63odc656e4) 10

[**Suspension Criteria & ResumptionRequirements**](#_k895ajo8yabh) **11**

[Suspension Criteria](#_33oplw9639be) 11

[Resumption Criteria](#_4l84c85mcrd7) 11

[**Test Deliverables**](#_qeihiy3s6ltz) **11**

[**Testing Strategy**](#_ob0xonbm8jti) **12**

[Test Process](#_czv19ofg576c) 12

[Understanding Requirements](#_b8q2l89w6oas) 12

[Develop Tests](#_on20q2mcmsut) 12

[Preparing Test Matrix](#_6xvjqndhmgm9) 13

[Reviewing test cases and matrix](#_g5kzi6r3i0j9) 13

[Prepare to Test](#_dngp5kvaum4m) 14

[Test Run](#_x2zaowurx1ux) 14

[Testing Types](#_k39etqbhjyke) 14

[Manual Testing](#_ozf6yjcmh2gz) 14

[Unit Testing](#_76zcx2m3u4np) 15

[Resource and Environment Needs](#_utp5hbdgdwsc) 15

[Testing Tool](#_tnxf52pght7n) 15

[Test Environment](#_pa3cyhwj0k73) 16

[**Risk, Assumptions and Dependencies**](#_nurwqi2lkzvu) **16**

[Risks](#_wnzluzciobd) 16

[Assumptions](#_5v7awt5j7lpl) 17

[Dependencies](#_atrput2ub2bu) 18

[**Summary**](#_8oase8104qn7) **18**

# 

# **References**

The list of documents for forming this test plan are:

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

# **Project Background**

The Learning Management System, “LMS” is a Web

application which helps students and faculty belonging to a University to facilitate and manage the assignments and grades in a more organized way. This web application should be free and accessible from any browser on desktop. This software includes all posting assignments by the faculty for specific subjects, where students can also submit their assignments. Later these assignments can be graded by faculties and they can also add comments to it.

The objectives of the learning Management System is to simplify the management of all the data related to academics that take place in a University. It will be a one-stop solution for both students and faculty to check their work related status. Safety, ease of use and most importantly the efficiency of information display and retrieval are some benefits the development team are going to present with this system. This system should be user appropriate, easy to use, provide easy recovery of errors and have an overall end user high subjective satisfaction.

# **Introduction**

The main purpose of the test plan document for the Learning Management System project is to discuss the testing details of the use cases of the Learning 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 Learning 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 and manual testing of the specified application. The design, development and testing of these reports will be based on the Learning management system project. Throughout the testing process we will be applying the standard test documentation specifications.

# **Purpose**

The main purpose of the test plan for the Learning 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 Learning 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.

# **Scope**

## Functions to be tested

|  |  |  |
| --- | --- | --- |
| Module | Applicable Roles | Description |
| Login | Faculty, student | Faculty or student should be able to login in the system |
| Dashboard | Faculty, student | Faculty, student must be redirected to dashboard after login where all its courses are visible. |
| Course Page | Faculty, student | Faculty, student must be redirected to specific course page after clicking one of the courses from dashboard |
| Edit Profile | Faculty, student | Faculty, student should be able to change password |
| Assignment List on faculty side | Faculty | Faculty should be able to view all its added assignments and perform various activities like update/add and delete. |
| Faculty Edit/Upload Assignment | Faculty | Faculty should be able to add deadlines, weightage and assignment that they want to provide. |
| Faculty Gardes | Faculty | Faculty should be able to view marks of students that he added till now. |
| Edit/Upload Grades | Faculty | Faculty should be able to add marks of assignments uploaded by student |
| Faculty Resource | Faculty | Faculty should be able to view all its resources of a particular course that he has added till now. |
| Faculty resource Upload | Faculty | Faculty should be able to add new resources to a particular course |
| Enrolled Student | Faculty | Faculty should be able to see list of all students enrolled in a particular course |
| Student assignment | Student | Student should be able to see a list of all the assignments added by faculty with deadline. |
| Student Submit Assignment | Student | Student should be able to add their assignment solution |
| Students Grades | Student | Student should be able to see grades and comments uploaded by faculty |
| Student resources | Student | Student should be able to see and download resources uploaded by faculty. |

## **Functions Not to be tested**

All the functionalities from Admin side are not to be tested as this project uses Django authentication system which provides inbuilt testing.

# **Testing Approach**

The approach, that used, in accordance to 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 Learning Management System project. The approach followed for testing the Learning Management System ensures that the major features of the project are adequately tested. The testing would be carried out on the Learning Management System while logging into the system as a Student or as a Faculty.

# **Entry and Exit Criteria**

## 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 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

## 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

# **Suspension Criteria & ResumptionRequirements**

## Suspension Criteria

Suspension criteria is a set of conditions which is 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.

## 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.

# **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
  + Manual Test Cases
* Requirement Traceability Matrix

# **Testing Strategy**

## 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.

### 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.

### 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.

### 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.

### Reviewing test cases and matrix

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

**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.

### 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.

### Test Run

The Test Case gives you context of the test while the Test Run includes information related to running the Test Case. The Test Case will include the test steps, conditions and expected result. A Test Run will keep track of the test result, when the test was run, Test Run status, Not Run, In Progress, Passed or Failed, Assign and the Test Cycle the test run lives in.

## Testing Types

### Manual Testing

|  |  |
| --- | --- |
| Test Objective | Test cases are executed manually without using any automated tool. All test cases executed by the tester manually according to the end user's perspective. It ensures whether the application is working, as mentioned in the requirement document or not. |
| Technique | Manual |
| Completion Criteria | If application is working as per the requirement document |

### 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 | Django tests |
| Completion Criteria | If a module passes all the Test Cases written for it, then it has Passed the test. |

## Resource and Environment Needs

### **Testing Tool**

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

### Test Environment

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

# **Risk, Assumptions and Dependencies**

## Risks

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Number** | **Description** | **Status** | **Severity** | **Mitigation Type** | **Detail of mitigating action to be taken** | | |
| 1 | Database might  not be properly linked | Closed | Medium | Avoidance | Testing the relations in the database | | |
| 2 | Continuously changing requirements | Work in Progress | High | Adapting | Continuously adapting to it. | | |
| 3 | Test data proves to be inadequate | Closed | Medium | Reduction | Testing will indicate what is required and will verify  suitability of data. | | |
| 4 | Lack of personal resources when test-  -ing is to begin | Closed | Medium | Avoidance | Ensuring the constant supply of  resources | | |
| 5 | Lack of knowledge of tasks in team  members | Closed | Medium | Avoidance | Make sure the team members have appropriate  knowledge of their domain | | |

## 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 website | Closed | The user will be able to see their courses only when they navigate to the website | | The user is able to  navigate | The user will not be able to view and submit assignments. | |
| 3 | All the courses are assigned to students and faculties by authority | Closed | No head authorization functionality in the system | | The organizer will need to first validate the courses details before assigning the courses to the users. | The system then can have multiple invalid, overlapped and unauthorized courses assigned to wrong users. | |
| 4 | The add and drop course need to be done offline | Closed | No add and drop course functionality  available in the system | | The user communicate  with the authority offline after viewing the allotted courses | The courses and assignments can be missed by the student or faculty if wrongly allotted. | |
| 5 | All the request for changes in grade and discussion on resources are done offline | Closed | No exchanging comments functionality  in the system, one way commenting is added. | | The faculty should discuss with the students individually. | The system will have no capability between the users sharing comments. | |

## 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 |

# **Summary**

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.

# 