EVENT MANAGEMENT SYSTEM ‘EventLog’

Test Plan Document

[**1.Test Plan Identifier**](#_heading=h.aerurpli28t2) **3**

[**2. References**](#_heading=h.gi6ho1tviqm3) **3**

[**3. Project Background**](#_heading=h.os1xo66zj50z) **3**

[**4. Introduction**](#_heading=h.8ctas7wfczle) **4**

[**5. Purpose**](#_heading=h.3bnvb1onl68m) **4**

[**6. Scope**](#_heading=h.ohc2sah6lxd8) **5**

[6.1 Functions to be tested:](#_heading=h.r6004c3fixcl) 5

[6.2 Functions Not to be tested:](#_heading=h.yr24yop0h7x0) 7

[**7. Testing Approach**](#_heading=h.oqo6x4500n2p) **8**

[**8. Entry and Exit Criteria**](#_heading=h.szlgmbv85nm0) **10**

[8.1 Entry Criteria](#_heading=h.5okr8hp72r6t) 10

[8.2 Exit Criteria](#_heading=h.96lkoqnp62ks) 10

[**9. Suspension Criteria & Resumption Requirements**](#_heading=h.qftf21ule4gi) **11**

[9.1 Suspension Criteria](#_heading=h.z9r7u1b9jf62) 11

[9.2 Resumption Criteria](#_heading=h.s1xyhoj5qfpg) 11

[**10. Test Deliverables**](#_heading=h.pal61fgy2x7n) **12**

[**11. Testing Strategy**](#_heading=h.yhec799d8jla) **12**

[11.1 Test Process](#_heading=h.5m2qursfuzt) 12

[11.1.1 Understanding Requirements:](#_heading=h.mp0ygkrjbkl1) 12

[11.1.2 Develop Tests](#_heading=h.83e5zp46zsic) 12

[11.1.3 Preparing Test Matrix](#_heading=h.7khz4ir52i4) 13

[11.1.4 Reviewing test cases and matrix](#_heading=h.4e7i42l0nc8v) 13

[11.1.5 Prepare to Test](#_heading=h.sngdwgf9mezf) 13

[11.1.6 Run Tests](#_heading=h.3ujx4vl2pb6l) 13

[11.1.7 Review Test Results](#_heading=h.fmapqyxq0qnc) 14

[11.2 Testing Types](#_heading=h.vvm1edleae1b) 14

[11.3 Resource and Environment Needs](#_heading=h.e92gnh5kipwi) 17

[11.3.1 Testing Tools](#_heading=h.6uxc9m7153do) 17

[11.3.2 Test Environment](#_heading=h.pf1qmrc6mftv) 17

[11.4 Testing Schedule](#_heading=h.3ebh21weuo) 18

[**12. Risk, Assumptions and Dependencies**](#_heading=h.f00hfuna3l35) **19**

[12.1 Risks](#_heading=h.4v28fprdjjxs) 19

[12.2 Assumptions](#_heading=h.b2hnynw3nniy) 20

[12.3 Dependencies](#_heading=h.jpgkfmue02uo) 21

[**13. Glossary**](#_heading=h.9ea2nr6c1d6) **22**

# 

# 1.Test Plan Identifier

Unique Test plan Identifier is-

‘Master Test plan for Event Log TP\_1.0’

# 2. References

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

* Project Plan
* Software Requirements Specifications
* Design Document

# 3. Project Background

The application system, EventLog is a platform which would allow the event organizers from the university to make the students aware about their events. The events may include the club activities, talks or any other events around the university. Whenever any team of organizers have decided any event they can add every detail of the event to the platform. Whenever adding an event the organizer is told to fill out a form which would include every major details of the event, from name, venue, fees if any and so on.Every organizer can add the registration link to the form too. They are also given the authority to modify the event whenever there may be any change of any form. When first login the student is asked to list down their preferred club names. These clubs will be their preferred club list or the list of their officially registered clubs.From the student perspective they have every access to viewing every detail of any event which has been added by the organizer. For their ease there are various form of view which are visible to the students. The students can view all the events throughout the university, the list of events registered by them, events from their preferred clubs. Every view format has their own merits and makes it easier for the user to go through the events which is basically the whole idea of EventLog.

# 4. Introduction

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

# 5. Purpose

The main purpose of the test plan for the Event Management 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 Event 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

# 6. Scope

## 6.1 Functions to be tested:

|  |  |  |
| --- | --- | --- |
| **Module** | **Applicable Roles** | **Description** |
| **Login** | User  Organizer | **User:** A user should be able to login in the system by doing google sign-in with his/her University Email Id.  **Organizer:** An organizer should be able to login by using registered university email id. |
| **Interested Club Details** | User | User**:** A user should be able to select at least 3 and at most 5 clubs which the user finds interesting on the first time of sign-in. |
| **View all Events** | User  Organizer | User and Organizer: should be able to select ‘all events’ lined up by the most recent events. |
| **View Club wise events** | User  Organizer | User and Organizer: should be able to select a club category from the bar menu and view the respective club events by double-clicking on the club name. |
| **View Registered Events** | User  Organizer | User and Organizer: should be able to view the events that the user has registered by clicking on the ‘Registered Events’ in the bar menu. |
| **View Suggested Events** | User  Organizer | User and Organizer: should be able to view the suggested events by clicking on the ‘Events for Me’ in the bar menu. It consists of all the events related to the clubs that the user has previously selected as ‘Interested Clubs’. |
| **View event details** | User  Organizer | User and Organizer: should be able to view the detailed description and details of the events by double clicking on the specific event part. It includes a detailed description, poster of the event, venue, date, time, fee, number of applicants registered and left seats, names of faculties or mentors if involved, and more. |
| **Register for events** | User | User: should be able to register for the events by clicking on register button |
| **View remaining seats** | User | User: should be able to view the number of seats left for the event that can be registered. |
| **Add Events** | Organizer | Organizer: should be able to add events by inputting required information such as Venue, Date, Time, Capacity of users, Faculty or mentors if involved, fees, poster, and so on. |
| **Modify Events** | Organizer | Organizer: should be able to modify events that he/she added |
| **Cancel Events** | Organizer | Organizer: should be able to cancel the events after stating valid reasons for the same. |
| **View Registered users** | Organizer | Organizer: should be able to view the registered users of the events that he/she added in the system. |

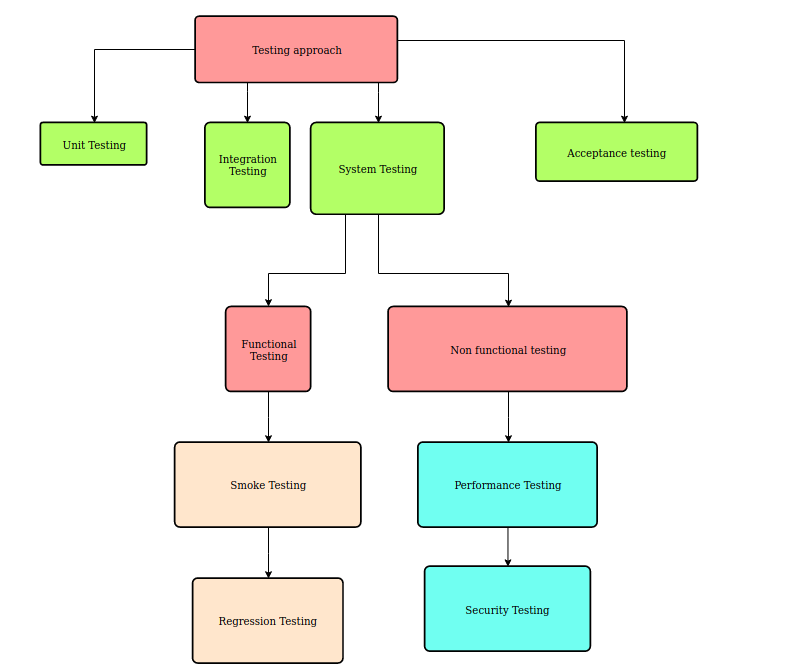
## 

## 6.2 Functions Not to be tested:

* Google OAuth Login will not be tested as the functionality is validated and already tested by google services and is not provided by our system.
* Material-UI components work in isolation. They are self-supporting, and will only inject the styles they need to display. They don't rely on any global style-sheets such as [normalize.css](https://github.com/necolas/normalize.css/).
* React Router is the standard routing library for React. React Router keeps your UI in sync with the URL. It has a simple API with powerful features like lazy code loading, dynamic route matching, and location transition handling built right in. Make the URL your first thought, not an after-thought.
* react-router-dom: It contains the [DOM](https://www.w3.org/TR/WD-DOM/introduction.html) bindings for React Router. In other words, the router components for websites react-router-dom: It contains the DOM bindings for React Router. In other words, the router components for websites.
* So these four modules are used directly from the standard libraries or authorized sources which are already tested and verified. That’s why we don’t have to test them again.

# 7. 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. At the end of each week the requirements identified for that iteration will be reviewed and will be tested. This section of the test plan describes the overall approach for testing the Event Management System project. The approach followed for testing the Event Management System ensures that the major features of the project are adequately tested. The testing would be carried out on the Event Management System while logging into the system as a User or an organizer of the system.



# 8. Entry and Exit Criteria

## 8.1 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 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
* Test Cases and Test Data
* Test Tools
* Appropriate Test Environment with all the necessary resources like tools and devices
* Executing the primary functional flows successfully by leveraging various test inputs

## 8.2 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
* Achieving complete Functional Coverage
* Identifying and fixing all the high-priority defects
* Fixing all the ‘Showstopper defects’ or ‘Blockers’ and ensuring that none of the identified Critical/Severity 1 defects are in Open Status
* Re-testing and closing all the high-priority defects to execute corresponding Regression scenarios successfully
* Test Logs generated
* Test Summary report generated

# 9. Suspension Criteria & Resumption Requirements

## 9.1 Suspension Criteria

Suspension Criteria in the context of software testing means suspending the complete or part of the testing activities.

* The build contains many serious defects which seriously or limit testing progress.
* More than 40% of the test cases fails
* Significant change in requirements suggested by the client
* Change of business requirements.
* Software/Hardware problems
* Assigned resources are not available when needed by the test team.

## 9.2 Resumption Criteria

Resumption criteria imply resuming the previously suspended activities. The component testing will resume when All issues in suspension criteria have been resolved or mitigated.

# 10. Test Deliverables

The following test documentation will be produced:

* **Test Plan** - This document deals with what needs to be done in UAT.
* **Test Cases** - The values input and results expected from tests.
* **Unit Test Cases**
* **IntegrationTest Cases**
* **Requirement Traceability Matrix**
* **Test Summary Report** - Summary of testing.

# 11. Testing Strategy

## 11.1 Test Process

The testing process outlines the testing process to be applied and can be considered to have four steps: Develop Tests, Prepare to Test, Run Tests and Review Test Results. These four steps are controlled by Plan Testing and Change Management.

### 11.1.1 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.

### 11.1.2 Develop Tests

* **Derive Acceptance Criteria** - Once the previous two activities are underway or completed then the set of questions to be asked about the system to see if it matches the capability 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.

### 11.1.3 Preparing Test Matrix

* Preparing a test matrix which maps test cases to respective requirements.
* This will ensure the coverage for requirements.

### 11.1.4 Reviewing test cases and matrix

This is a key quality process of checking all documentation produced during the development of the system.

* Peer review will be conducted for test cases and test matrix by QA Lead
* Any comments or suggestions on test cases and test coverage will be provided by reviewer respective Author of Test Case and Test Matrix
* Suggestions or improvements will be re-worked by author and will be sent for approval
* Re-worked improvements will be reviewed and approved by reviewer

### 11.1.5 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.

### 11.1.6 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.

### 11.1.7 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 organisation wants to achieve with the system. A flawed system which delivers capability to an organisation 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

## 

## 11.2 Testing Types

**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. |

**Integration testing**

|  |  |
| --- | --- |
| Test Objective | When putting several units together that interact we conductIntegration testing to make sure that integrating these units together has not introduced any errors. |
| Technique | Manual |
| Completion Criteria | After integrating the modules, if they work without introducing any errors it has passed the test. |

**Regression testing**

|  |  |
| --- | --- |
| Test Objective | The re-execution of some subset of tests that have already been conducted to ensure that changes have not propagated unintended side effects |
| Technique | Manual |
| Completion Criteria | When all the different modules are integrated, the changes that are incorporated with it are taken care of in depth(exhaustively) so as to test that there are no additional errors introduced.  Eg - When we integrate a login module with another module, we perform the same test cases again. |

**Smoke testing**

|  |  |
| --- | --- |
| Test Objective | It comprises a set of tests that aim at ensuring that the most important functions work. The result of this testing is used to decide if a build is stable enough to proceed with further testing. It covers most of the major functions of the software but none of them in depth. |
| Technique | Manual |
| Completion Criteria | When two modules are integrated, the changes that are incorporated with it are taken care of non-exhaustively so as to test that there are no additional errors  introduced i.e. to ensure that most important functions work.  E.g. - Payment details invoice must be generated correctly when the user submits the application and the admin approves it. |

**Security testing**

|  |  |
| --- | --- |
| Test Objective | A software testing that intends to uncover vulnerabilities of the system and determine that its data and resources are protected from possible intruders |
| Technique | Manual |
| Completion  Criteria | Captcha functionality is kept to block spammers from intruding into the software |

## 11.3 Resource and Environment Needs

### 11.3.1 Testing Tools

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

### 11.3.2 Test Environment

* Windows 10: Chrome (latest), Firefox (latest)
* Mac OS X: Chrome (latest), Firefox (latest), Safari (latest)
* Linux Ubuntu: Mozilla (latest)

## 11.4 Testing Schedule

|  |  |  |
| --- | --- | --- |
| **Task Name** | **Date** | **Comments** |
| Review Requirements  documents | 04/05/2020 | The requirements gathered from  various stakeholders were reviewed. |
| Test Planning | 05/05/2020 | Decided regarding the approach we  are going to follow during the testing  phase |
| Develop test cases | 06/05/2020 | Creating the test cases manually |
| Create Test Case Matrix | 06/05/2020 | Formulated the test cases in matrix form for easy and better understanding. |
| Review Test cases and  matrix | 07/05/2020 | Reviewed whether the test cases cover all the requirements. |
| Prepare to Test and Run Tests | 07/05/2020 | Preparing the environment and the test data to run the tests |
| Review Testing Result | 08/05/2020 | Check the results (pass/fail) of the test cases and change code accordingly |
| Functional testing | 08/05/2020 | To test rest of the modules  functionality using appropriate test  cases. |
| Review Testing Result | 09/05/2020 | Check the results (pass/fail) of the test cases and change code accordingly |
| Regression testing | 09/05/2020 | To ensure that previously developed and tested software still performs  after a change. |
| System testing | 10/05/2020 | To test a complete and integrated software. The purpose of this test is to evaluate the system's compliance with the specified requirements. |
| Resolution of final errors | 11/05/2020 | To check the status of all the previously addressed errors after improvement. |

# 12. Risk, Assumptions and Dependencies

## 12.1 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 test-  -ing 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 | | |

## 12.2 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 website | Closed | The user will be able to see and register for events only when they navigate to the website | | The user is able to  navigate | The user will not be able to view and register for events | |
| 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 approp-  -riate storage space | The system will not work properly | |
| 4 | All events and talks permission and the  location have already been taken by the  clubs from the authorities before adding into the system. | Closed | No head authorization functionality in the system | | The organizer will need to first validate the event details before adding the event to the site | The system then can have multiple invalid, overlapped and unauthorized events | |
| 5 | The paid events registration process is  assume to be done offline | Closed | No payment functionality  available in the system | | The user communicate  with the organizer offline after viewing the respective event | The paid events will stall and will not function properly | |
| 6 | All clubs members/organizers  registration is done offline. | Closed | No head authorization functionality  in the system | | The head should add the members or incharge users  to be organizers | The system will have no distinguish capability between organizer and  normal user | |

## 

## 12.3 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 |
| 3 | The system is dependent is Google  OAuth for login functionality | Closed | High |

# 13. 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.