Integrated Campus

Test Plan V_2.0

Group 2

INTEGRATED CAMPUS | SEN-WINTER 2013

Revision History:

Version	Primary Author(s)	Description of Version	Reviewed By	Date Completed
V 2.0	Sushant Pritmani Nalin Patidar	Version for abstract view of our Testing phases	Jayesh Hatila	7-3-2013

Contents:

Introduction

Objective:

This document will describe the testing approaches that will be followed by the team to assure the quality and tracking of the project progress. In this document we will be considering abstract view of the testing phase. We will be mainly emphasizing on the following points:

- To detail the activities required to prepare for and support the test.
- To define the test tools and environment needed to conduct the test.
- To detail the dependencies and effort required to conduct the System Test.
- To describe the testing lifecycle and procedures to be used
- Test plan will intend to prove the functionality delivered by the development team, is as specified in the System Requirement Specification document.

Testing Strategy:

The Test Strategy describes the high-level strategy and methodology that will be used to plan, organize and manage testing of our project. The primary purpose of these tests is to uncover the systems limitations and measure its full capabilities.

As we are following evolutionary incremental model, we will be doing project testing in every increment (two increments will be there) and for every increment we will test the module from following eight levels:

- Component testing
- Integration testing
- Security testing
- Recovery testing
- Performance testing
- Acceptance testing
- System testing

We have explained each level in detail in section of testing strategies.

Scope:

Scope of our test plan is to test each and every feature of the project from component testing level up to system testing level and to provide a desired quality and best possible bug free software to client. To simulate the exact condition of the project as of real time deployed project we will test our features according to their dependencies and check the result of failures of one feature on the others.

Out of scope:

- Taking care of the issues after the system has been handed over to the client.
- Post deployment maintenance and improvements

Reference materials:

- System Requirements Specification.
- Test Plan Outline (IEEE 829 Format)

Test items

Program module:

We will test each feature after its implementation phase is done. As per now we have planned to make two increments:

In first increment we will be implementing-

1. Attendance Feature
2. Polling Feature
3. File Upload

In second increment we will be implementing -

1. Discussion Forum
2. Notice Board
3. Timetable Feature

So for testing purpose also we will work on each feature separately which is termed as component testing. Only after clearing the phase of component testing the feature will be forwarded for integration testing and further testing phase as listed in section of Testing Strategies.

Job control procedure:

For better readability of the code we will be following coding standards which will help us in easy integration and on long term basis this will help for further maintenance and expansion of the project if required. We will also consider separating of the modules on the low degree of their mutual coupling which will decrease their dependency and allows some degree of freedom to each sub part of the project. Every module implemented will have a comment section which would be like a short description of that particular module including the acronyms used, inputs to the module and expected output from it. This job control language procedure helps out the chief programmer to present the information in a way that is easy for the end-users and other programmers to read and includes other useful information. We will also update document versions depending on the updates of the features which will help us track the progress of the project

User procedure:

We will implement the best possible effort to ensure the recovery or avoidance of the errors and exceptions that can be generated by the unknown behavior of user inputs or unknown events that can lead the system to undesired state. The system will take care of and check if the user’s information is correct, complete and comprehensive. The System will check the completeness of information by checking the fields which are mandatory if they are filled or not**.**

Operator procedure:

Though we will be using those tools and techniques which can be used cross platform but we will assure that the system is deployable to its complete features on the environment specified in the System Requirement Specification (SRS) document. We will use/test our features on/from the below listed tools and techniques which are also described in SRS document

Features to be tested

This is a listing of what is to be tested from the **USERS** viewpoint of what the system does. This is not a technical description of the software, but a USERS view of the functions. As per the requirements captured by the SRS document the following modules would be tested in accordance with the test schedule.

1 Registration of users by admin
2 Login/Logout
3 Forget Password/Change Password
4 Creating Courses
5 TA Assignment to a course
6 Faculty Assignment to a course
7 Course Material Upload and Download
8 Adding/Updating of attendance by TA and/or Faculty
9 Viewing Attendance status by Faculty and Student
10 Creating Polls and Surveys by Faculty
11 Responding anonymously to Polls and Surveys by Students
12 Database Modification
13 Dynamic Query Generation
14 Noticeboard
15 Adding Topic by student
16 Answering Topic by Faculty/Student
17 Viewing Personalized Timetable by Student/Faculty
18 Checking the customized version of Timetable by Admin

19 Uploading Timetable by Admin

20 Viewing Topic by Student/Faculty

Features not to be tested

These are the features which are implemented using external APIs and hence are not entitled for testing, on the other hand these will be tested for proper integration with the complete software.

- SMS sending APIs for sending notifications to student and his/her parents regarding his/her performance.
- Mail sending APIs for sending reseted password in case when the user forgets his/her password

Approach

The purpose of the Test Plan is to achieve the following:

- Define testing strategies for each area and sub-area to include all the functional and quality (non-functional) requirements.
- Divide Design Spec into testable areas and subareas.
- Define bug-tracking procedures.
- Identify testing risks.
- Identify required resources and related information
- Provide testing Schedule.

In our current module, we'll start testing by using a Bottom Up testing approach. The testing will start by component testing in which every component control structure will be tests by the developer himself for any error and coding standards (this can be done concurrently). After checking, all the components will be integrated and each step of integration the modules will be tested under integration testing.

Then the integrated system will be tested under validation testing which will make sure that all the requirements are addressed by the system and finally the whole system will be tested under System Testing which will see how the system works together(this may include running on DA-IICT server for testing).

Test Strategy

The test strategy consists of a series of different tests that will fully exercise the Integrated Campus. The primary purpose of these tests is to uncover the system's limitations and measure its full capabilities. A list of the various planned tests and their brief explanation are as follows.

Component testing:

As evolutionary incremental model is being followed for the project, each increment's goal will be divided in certain modules in order to make the task easier. Component testing will verify important control paths to uncover errors within boundary of the module. The relative complexity of tests and uncovered errors is limited by constrained scope established for unit testing. In order to test the individual units and debug the errors if any we have taken up the combined approach of brute force, back tracking and cause elimination. The brute force method involves incorporating "print statements" to print intermediate values with the hope that this will help in manual debugging. The values have been checked if they are arriving right by this manner.

The backtracking method helps us in fixing the bugs and checking the values one by one received from other dependent modules. Beginning from the location where the symptom/value is observed/ received, source code will be tracked backwards till the error/result is discovered.

In order to fix few bugs we also intend to use the cause elimination approach where a list of causes which could affect the outcome is made and tests will be conducted to eliminate these causes. We will be testing all the classes or cluster of classes in the core module comprising of all the base classes.

Component Testing will ensure that each module so developed is working as desired and have all the desired functionalities before passing it on for integration with the other modules.

Integration testing:

Component testing will be followed by Integration Testing. Modules will be integrated in a planned manner and after each integration step, will be subject to testing to make sure that integration has not affected the individual functionality of the module and the integrated module thus obtained has desired functionalities. We will use a bottom up method to integrate our application. We would use integration test cases to test if the modules work fine once they are integrated.

Security testing:

Security Testing is intended to cater following six basic issues: confidentiality, integrity, authentication, availability, authorization and non-repudiation. For example, security tests will verify that unauthorized user access is prevented, certain editing/deleting rights functions as desired etc. System will also ensure the protection of information & data and maintains functionality as intended

Recovery testing:

It is vitally important that all the data is recovered in case of a system crash or failure and no corruption of the data has occurred. Recovery Testing will be done to get a hold on to system's ability to recover from sudden crashes or hardware failure.

Performance testing:

As the system is built for the academic purpose it is bound to handle
Many user at a time which make the response time and load handling vital issue to be tested. To check the responsiveness and stability of system while handling the workload, performance testing will be done**.** Basic issues like numbers of user accessing the system at the same time and how the response time of system acts accordingly will be addressed while doing performance testing.

Acceptance testing:

Once the system or the increment is ready for implementation, user acceptance testing will be done through traceability matrix which clearly state the requirement of the user. The purpose of these tests is to confirm that the system so developed is in accordance with the specified user requirements and is ready for operational use.

System testing:

After being done with all the increments, the whole integrated system will be tested in totality to ensure that system meets all the specified requirements. System Testing would be carried out to fully exercise the program as a whole to ensure that all elements of the integrated system function properly.

Pass/Fail criteria

Each module has its own functionality which will state the pass/fail criteria for that module while it's undergoing the testing phase. After going through all the test cases at all the levels, if modules functions as desired then it will be said to have satisfied the passing criteria.

Suspension criteria:

The following specify the conditions when all or a portion of testing activity can be suspended:

1. Hardware/software is not available at the times indicated in the project schedule.
2. Source code contains one or more critical defects, which seriously prevents or limits testing progress.
3. Assigned test resources are not available when needed by the test team.

Resumption criteria:

Resumption of the testing will begin only after the error is fixed or any of the condition that resulted in the suspension of the testing process in addressed successfully. The whole module or unit will be tested again in such cases.

Approval criteria:

After going through all kinds of testing required, to check the proper functionality of the module, the module will be approved if it's giving the expected output. If in case any error occurs it will go down in resumption mode for error fixing.

Testing process

Test deliverable:

Testing will provide specific deliverables. They are System Test Plan, Test Cases, test logs and test report.

Testing task:

Testing tasks will include writing a test plan, building the test cases, conducting test and evaluating their results and finally documenting the result.

Responsibility:

The responsibility for managing, designing, preparing, executing, witnessing, checking, and resolving test activities will be vested to 3-4 team members.

Environmental Requirement

Software Requirements:

XAMPP
Development Tools: Dreamweaver, Sublime Text Editor
Database Server MySQL

Operating System
Web Browser

Hardware Requirements:

PC, Laptop, Tablet or Mobile Phones (assumed that all support web browsing through wired or wireless connection.)

Risk and contingencies

Requirements:

As we are following the evolutionary incremental model so depending on the feedback from client we can update features but for now we will be considering the features listed in current SRS document.

Schedule:

Considering the dependencies of features we will try to strictly follow the project timeline which will satisfy the precondition for its dependent features.

Technical:

As for database we will be using MySQL database which has inbuilt functionality of rollback so we can rest assure of recovery in case of system crash or some other crash related technical fault.

Management:

At the time of missing the deadline or lagging on the time scale of the project management support will help us retaining the time span allocated to test plan which we have decided while making the timeline of phases. Management can reduce the risk of delays by supporting the test team throughout the testing phase.