

Software Test Plan

“PortAc”: Academic web portal

Supervised by:
Dr. S.C. Gupta
Course instructor CSL740

Submitted by:

ANUBAHV SINGH
RAHUL NAGAR
BHARAT RATAN
ABDUL HADI SHAKIR

INDEX

PURPOSE

AUDIENCE

1. INTRODUCTION
 - 1.1. Testing strategy
 - 1.2. Scope
2. RELATIONSHIP TO OTHER DOCUMENTS
3. FEATURES TO BE/NOT TO BE TESTED
 - 3.1. Features to be tested
 - 3.2. Features that will not be tested
4. PASS/FAIL CRITERIA
 - 4.1. Component Pass/Fail criteria
 - 4.2. Integration Pass/Fail criteria
 - 4.3. System Pass/Fail criteria
5. APPROACH
 - 5.1. General testing strategy
 - 5.2. Test Case alignment with test phase
6. SUSPENSION AND RESUMPTION
 - 6.1. Unit test
 - 6.2. System design Change
7. TESTING ENVIRONMENT (SOFTWARE AND HARDWARE REQUIREMENT)
 - 7.1. Hardware required
 - 7.2. Software required
8. TEST CASE AND TESTING STRATEGY
 - 8.1. Component Testing
 - 8.1.1.1. Login
 - 8.1.1.2. Signup
 - 8.1.1.3. Room booking
 - 8.1.1.4. Course registration
 - 8.1.1.5. Counselling system
 - 8.2. Integration testing
9. RISK AND CONTINGENCY FACTOR

Purpose

The STP we have developed is to make the software full proof prior to its use. The main aim of this STP is to find any bug or any feature that should be in the software but is not there or is not working as per requirement. We have thoroughly tested the software but if anything skipped our eyes the user is welcome to report it.

We have tested each component separately and then have tested it after integrating it.

Audience

The audience of this document is the project team and the instructor of the course. The document is intended to accomplish its purpose only for the intended audiences.

1. Introduction

“PortAc” is an academic web portal which offers a web based platform for an academic institute. It provides few of the basic features any institute must provide in order to run properly. The software is aimed to give the basic features like student/faculty LOGIN, Maintaining profiles, e-counselling for admission process (having the feature of choice filling, etc.), course registration for students, room booking system for faculty, college wiki, etc. Efficient use of database and algorithms is employed to give maximum efficiency. This Software Test Plan (STP) is designed to test the correctness and efficiency of the software before it is delivered to user. In this plan we have thoroughly tested each and every aspect of the software and have the software tested by different individuals.

1.1 Testing Strategy

The STP is developed in order to make the software full proof prior to its use. The main emphasis is given to the testing of handling of database as in this software handling the database is one of the main challenge. We have tried to test all the corner possibilities and have given this software to various individual and have taken their feedback. The external appearance and the employed algorithm is compared to many of the other colleges and even asked to others how will they rate it and if they can find any bug.

1.2 Scope

Special care has been taken to make sure that the software can be easily updated and can be used for any kind of environment. The database is tested thoroughly for all kind of entries and the software is tested for all kind of users.

2. Relationship to other documents

This section explains the relationship of the test plan to the other documents produced during the development effort such as the RAD, SDD, and ODD (Object Design Document). It explains how all the tests are related to the functional and nonfunctional requirements, as well as to the system design stated in the respective documents. If necessary, this section introduces a naming scheme to establish the correspondence between requirements and tests.

3. Features to be tested/not to be tested

3.1 Features to be tested

The features that are to be tested include those features whose working will affect the swift working of the software. Each and every module that are built are tested alongside and after integration the whole website was tested for proper coordination. Following features can be tested:

- Login
- Room booking
- E-counselling
- Course registration

And their associated login, allotment algorithm, error handling and interoperability.

3.2 Items that will not be tested

Off-The-Shelf components

Some of the components that are used directly from net will not be tested. For example college wiki is used from internet application.

Infrastructure components

The actual database software utilized is assumed to work as designed and will not be directly tested for functionality. Performance tests will be done during system test with respect to GUI response time that will involve the database. However, no testing will be done directly against the database.

The internet/WIFI backbone will be utilized during testing, however, no tests will be written executed to directly test the communications backbone.

4. Pass/Fail criteria

This section specifies generic pass/fail criteria for the tests covered in this plan. They are supplemented by pass/fail criteria in the test design specification. Note that “fail” in the IEEE standard terminology means “successful test” in our terminology.

4.1 Component Pass/Fail criteria

Tests executed on components only pass when they satisfy the signatures, constraints, and interfaces dictated by the Design Specification for that component.

If a test exhibits a product failure to meet the objectives of the object design specification, It will be re-developed.

4.2 Integration Pass/Fail criteria

Tests executed on integrated components only pass when they satisfy the signatures, constraints, and interfaces dictated by both the object design specification and the system architecture specification. This includes boundary conditions, and tests that explicitly manipulate the interface environment (such as the physical connection to the database server).

4.3 System Pass/Fail criteria

Tests executed against the system use the functional requirements, non-functional requirements, and use cases as the oracle to determine pass or fail.

5. Approach

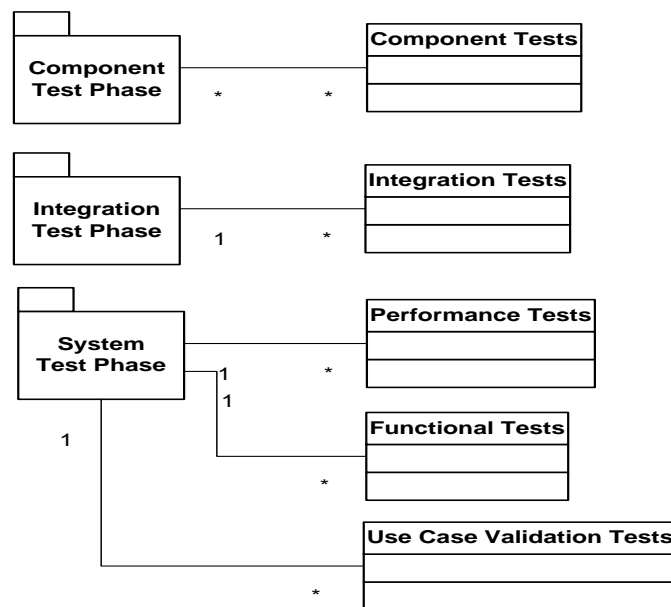
The test will be performed by exhaustively using all the possibilities and checking if the results match with what should be the result ideally. The test shall be conducted on a pre-maintained Database which should be according to the software requirement.

5.1 General Test Strategy

Unit testing and component testing will be performed on the components as they are developed. Test will be executed using test code and checking for all the test cases and boundary conditions.

As integration testing will include both front end and back end i.e. back end coding and algorithm as well as GUI therefore it will be manually done and will take time. manual end-to-end tests to validate use cases will be performed.

5.2 Test Case alignment with test phases



Here we have assumed that all the modules are complete in themselves (algorithms, database and interfaces) therefore it will be suffice to say that we will check only modules working and assume that the integrated system will work fine.

6. Suspension and resumption

This section specifies the criteria for suspending the testing on the test items associated with the plan. It also specifies the test activities that must be repeated when testing is resumed.

6.1 Unit Test

As components are being developed, unit tests will be developed to test the interfaces of the components and low-level unit tests will be developed to test the methods of the underlying classes in the components.

6.4 System Design Changes

If at any point in time issues are submitted that require a design change to the system, all testing will be suspended. After the changes to the requirements, system architecture, and object design are made, a review and updates will be performed of the test specifications to ensure they properly align with the revised system changes. After updates are made, testing will resume.

7. Testing materials (hardware/software requirements)

For testing the software following things are required:

7.1 Hardware required

A laptop or PC will be required for testing each of the components individually and then the integrated software.

7.2 Software required

Software required in this system is minimal. The database server needs the appropriate database (MySQL) installed, setup, and configured properly.

The client machine needs Dreamweaver 3 installed and properly configured with Internet Explorer 6.0 or newer.

8. Test cases and testing strategy

The test will be done on each component individually and then each of the integrated component will be checked if they are interoperable. The test will be performed by exhaustibly using all the possibilities and checking if the results match with what should be the result ideally. The test shall

be conducted on a pre-maintained Database which should be according to the software.

8.1 COMPONENT TESTING

8.1.1 LOGIN

<i>Input</i>	<i>Expected Output</i>
Correct username and password of student	Redirect to Welcome page for existing student
Correct username and password of faculty	Redirect to Welcome page for faculty
Correct username and password of guest	Redirect to Welcome page for guest
Incorrect username or password:	
a) Username does not exist	Redirect to same page(login) with an error message
b) Password does not match with username	Redirect to same page(login) with an error message

8.1.2 SIGNUP

<i>INPUT</i>	<i>EXPECTED OUTPUT</i>
Valid: Name, User ID, Password, Rank, Date of birth	Update the database and redirect to the counseling home page.
Invalid: Name, User ID, Password, Rank, Date of birth	Show an error message and redirect to the counseling home page.

8.1.3 ROOM BOOKING

<i>Input</i>	<i>Expected Output</i>
If room is to be booked:	
a) Entries are filled correctly	User ID and password will be shown
b) Entries are not filled in the correct format	Reload the same page with corresponding error message
To cancel a request	
a) If request ID and Password is correct	Delete the request and Show the message
b) If request ID does not exist	Reload the same page with error message
c) If request ID and Password does not	Reload the same page with error message

match	
To check status	
a) User ID is correctly entered and room has been allotted	Page showing room number
b) User ID is correctly entered and room has not yet been allotted	Page showing request consideration
c) If request has expired and user is checking	Page showing error message

8.1.4 COURSE REGISTRATION

INPUT	EXPECTED OUTPUT
A course choice:	
1) From list of available options.	Update the DB and show the corresponding course in the cart.
2) not available in DB	Show an error message and redirect to course registration welcome page.
Delete a choice	Remove the course from DB and registration cart.
Registration status:	Show the user's allotted courses.
1) Course has been allotted	
2) Course has not been allotted	Show an error message and redirect to course registration home page.
Admin allots the course:	
1) Before the deadline date	Show an error message and redirect to admin's home page.
2) After the deadline date	Run course allotment algorithm and update the DB.

8.1.5 COUNSELLING SYSTEM

INPUT	EXPECTED OUTPUT
A program choice:	
1) from list of available options.	Update the DB and show the corresponding program in the registration cart.
2) not available in DB	Show an error message and redirect to counseling system home page.
See the registration cart	Show the choices corresponding to that user from the DB.
Registration status:	Show the user's allotted program.
3) program has been allotted	
4) program has not been allotted	Show an error message and redirect to counseling system home page.
user clicks on help	Show the list of available course programs, their codes and seats available in each program.
Admin allots the program:	
3) Before the deadline date	Show an error message and redirect to admin's home page.
4) After the deadline date	Run program allotment algorithm and update the

8.2 INTEGRATION TESTING

As all the components will be tested and every component is independent therefore integration testing will not be required. System testing will be done by giving all the permutation and combination of above test cases individually. Performance and other aspects will depend upon the server on which we are hosting the website therefore need not be tested.

9 Risks and contingencies matrix

The main risk associated is the late delivery of software. The other risk may be that the software is not up to the requirement and design documentation and may be some of the features have been omitted or missed. The integration step may also turn up in a risk as at that step may be the data types of different modules do not match.

Many of the risks can be overcome by clearly adhering to the requirement documentation and design documentation.

We also have dependency on the external server we are using to host our website. We have to trust the efficiency of the hosting site and that the data will not be erased.