Test Plan

KSU Student Portal

Version 2.0

Submitted in partial fulfillment of the requirements of the degree of MSE

Javier Ramos Rodríguez
CIS 895 – MSE Project
Kansas State University

Table of Contents

1. Test plan identifier

KSUPORTALMSE-V2.0

2. Introduction

The purpose of this document is to describe the plan for testing the main functionalities of KSU Student Portal application. The document will also describe the activities related to testing the software and the environment and tools that will be used to test the software.

3. Test items

The critical Use Cases of the application software will be tested. The test will be performed after the committee chair signs off the first version of the software. The critical use cases will be the ones chosen to be tested since they reflect the main functionalities of the system, it doesn't mean that the other use cases are less important but their functionality is very similar to the others and the developer will have to implement the same way to assure the desired quality.

4. Features to be tested

The next use cases will be tested:

- **4.1.** Register in the application
- **4.2.** Log in the Application
- **4.3.** Delete User
- **4.4.** Search Public Items
- **4.5.** Browse Events
- **4.6.** View Event
- **4.7.** Create Event
- **4.8.** Delete Event

4.9. Create Article

4.10. Edit Profile

4.11. Add Blog Entry

4.12. Change Password

4.13. Change Language

4.14. Check Visibility

Access control will be tested to check that an unknown user access to private information.

5. Approach

The tester will fill out the different forms in the application and analyze the system outputs. The application will check that all the fields are properly filled and will perform the desire operation.

Black Box testing will be the approach used by the tester. To help the tested the developer will write equivalence classes that will be use to denote what values are expected in the corresponding fields and what values should not be accepted. The equivalence classes will be use by the tested to perform the black box testing. So, the tester doesn't have to know how the functions are implemented, He/She only cares about the input and output of the different use cases.

For example in the Register use case the system should check that all the required fields are filled, that the login doesn't already exist and that the proper values are introduced in the corresponding fields (i.e. numbers are introduced in the telephone number field). If all the fields are correct a new user should be added to the application.

Unit test will be applied to the code to check the different methods and functions. Also, we will check the class interfaces to make sure that the different modules are integrated correctly.

6. Item pass/fail criteria

The software must be able to pass the tests for all the critical uses cases described in the Architecture Design document. Each feature will be considered to be passed if it satisfies the corresponding requirement in the document and failed if any or none of the behavioral expectations are met as described.

7. Suspension criteria and resumption requirements

7.1. Suspension criteria

If a test case fails, testing will be suspended for all dependent features. The failed test case will be logged into a test log along with a description of the failure.

7.2. Resumption requirements

Testing for the failed test case will resume after the bug has been identified and resolved. Independent test cases will continue to be executed in parallel to bug fixing.

8. Test deliverables

The following artifacts will be produced after the tests are conducted on the application:

- Test Plan
- Formal Inspection Checklist
- Assessment Evaluation

9. Environmental needs

This section will outline the necessary and desired properties needed for the test environment.

9.1. Physical Characteristics

The application should work fine in both Linux and Windows platforms and should be Database Independent.