**Project Test Plan**

Charlie Farris

University of Arizona Global Campus

CST 499: Capstone for Computer Software Technology

Dr. Charmelia Butler

November 30, 2022

**Test plan for**

**Student Portal**

*ChangeLog*

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Change Date** | **By** | **Description** |
| version number | Date of Change | Name of person who made changes | Description of the changes made |
| 1.0 | 11/30/2022 | Charles Farris | Initial document |
|  |  |  |  |

1 Introduction 2

1.1 Scope 2

1.1.1 In Scope 2

1.1.2 Out of Scope 2

1.2 Quality Objective 2

1.3 Roles and Responsibilities 2

2 Test Methodology 3

2.1 Overview 3

2.2 Test Levels 3

2.3 Bug Triage 3

2.4 Suspension Criteria and Resumption Requirements 3

2.5 Test Completeness 4

3 Test Deliverables 4

4 Resource & Environment Needs 4

4.1 Testing Tools 4

4.2 Test Environment 4

5 Terms/Acronyms 5

# Introduction

The purpose of the Student Portal Website is to provide a means for each student to register, login, maintain their information, and register for courses. In the future, additional functionality may be added to the Portal to provide additional information or access to other applications like course Transcript information, course materials, and grades.

## Scope

### In Scope

This project's scope is to create a system to allow students to register and manage their personal information and course enrollment. This system is intended to be used by learning institutions. The system will need to enable new students (users) to register for an account, manage their information, enroll in courses, and drop courses. As part of registering for a new account, the system must assign a unique identifier at the time of registration. Upon completion of registration, the user must be able to log into the system and have access to the pages and data commensurate with their role within the system (i.e., student or admin). Users must be able to enroll in courses in any of the available terms (Fall, Winter, or Spring) for any open year. Available courses for the selected term and year must be presented to the user. Each course must have a minimum and maximum capacity assigned to facilitate waitlist functionality.

### Out of Scope

The functionalities that are considered out of scope for this project are an admin page to add new courses to the system, and a user support system, including the ability to contact support and for admins to answer those support inquiries. Additional waitlist functionality for the system to check the waitlist for a given course and automatically enroll the first student on the waitlist should an enrolled student drop the course. This would include system notification to the newly enrolled student.

## Quality Objective

Overall objectives planned to achieve without testing.

* Ensure the Application Under Test conforms to functional and non-functional requirements
* Ensure the AUT meets the quality specifications defined by the client
* Bugs/issues are identified and fixed before going live

## Roles and Responsibilities

The test manager will be responsible for facilitating the testing project, coordinating the availability and schedule of testers, and training them as needed. Each tester must understand the expectations regarding the completion date and level of quality for each test. The Test Manager will also communicate any risks to the team.

## Staffing and Training Needs

Testing will be conducted by three to five testers. All testers will conduct testing on each system but not on their own code. The testers assigned should have at least a basic knowledge of the Student Portal platform.

# Test Methodology

## Overview

The Agile methodology has been chosen for this project due to its iterative nature, allowing testing teams to test while defects are reported back to development for resolution. This process will allow for the best possible quality product while confirming that all requirements are met.

## Test Levels

2.2.1 Component Testing

Component testing verifies whether an individual software component meets the specifications outlined in the scope of work.

It also verifies each component’s specifications within the developed software program. Component testing is essential in determining whether the tasks, behaviors, structure, interfaces, or any subsystem within the component are operating according to the design specifications (Pressman, 1992).

2.2.2 Integration Testing

Integration testing checks whether a collection of components interact with the system per their configuration within the overall technical system design. It is crucial to validate the comprehensive system's technical components having been split out from the scope of work (Pressman, 1992).

2.2.3 System Testing

System testing checks whether or not the required system specifications within the system design have been met. It also verifies the design process of the functional system. System testing is vital for validating the technical components outlined in the scope of work to ensure the fulfillment of the functional system design requirements and the overall system (Pressman, 1992).

2.2.4 Acceptance Testing

Acceptance testing verifies whether the developed system meets the requirements and expectations outlined in the scope of development. They also confirm whether a system will meet the requirements and expectations of end users (Pressman, 1992).

## Incident Reporting

The goal of incident reporting is to

* To define the type of resolution for each incident.
* To define the severity and priority for each incident.
* To prioritize incidents and defects and determine a schedule for all “To Be Fixed Defects’.

## Suspension Criteria and Resumption Requirements

2.4.1 Suspension Criteria

Incidents requiring suspension of all testing are fatal errors that halt all progress of a component, or the system, preventing further testing, or that fail to meet any of the pre-defined functional or non-functional requirements.

2.4.2 Resumption Criteria

System testing can resume once all identified system issues preventing further testing have been found, an incident has been generated and documented, and all defects have been verified as corrected by the original reporting user.

## Test Completeness

Testing will be considered completed on this project when all the following exit criteria have been met. These exit criteria are a set of conditions or activities that need to be satisfied to complete our testing on the ecommerce solution.

* Test coverage reaches 98%
* All manual and automated test cases have been executed
* All open defects have been fixed and implemented or will be fixed in the next release
* All high-priority defects have been fixed and implemented
* All critical test cases have been executed and passed

# Test Deliverables

The following deliverables will be delivered throughout the duration of the project.

* Test Plan
* Test Cases
* Requirement Traceability Matrix
* Bug Reports
* Test Strategy
* Test Metrics
* Customer Sign Off

# Resource & Environment Needs

## Testing Tools

GitHub will be used to manage the different iterations of the software and will handle the versioning.

Jira will be used to log and track all defects.

## Risks

Test-driven development will mitigate project risk, and all tests will be designed to quickly identify and resolve defects. This process will allow the verification of requirements to ensure they are being met. It will also aid in ensuring that the testing and overall project remains on time and within budget. Early testing will also help provide a higher-quality end product.

Potential risks that can be encountered:

* Testing doesn't begin early enough in the project causing additional costs
* Testing budget exceeded due to potentially preventable defects not getting caught
* Unable to staff qualified testers adequately to meet testing requirements
* Testing outside the defined scope and requirements of the project or over-testing.

# Terms/Acronyms

| TERM/ACRONYM | DEFINITION |
| --- | --- |
| API | Application Program Interface |
| AUT | Application Under Test |
| SDLC | Software Development Life Cycle |

**References**

*Free estimate templates | smartsheet*. (n.d.). Retrieved from

<https://www.smartsheet.com/job-work-estimate-templates>.

Homès, B. (2012). *Fundamentals of Software Testing* (1st ed.) [E-book]. Wiley-ISTE.

Kannan, M., & Lokeshwari, K. (2017). Comparison of software testing tools with respect to tools

and technical related parameters. *International Journal of Advanced Research in Computer Science, 8(9)*, 70–75. <https://doi.org/10.26483/ijarcs.v8i9.4908>.

Pressman, R. S. (1992). Software engineering: A practitioner's approach (8th ed.). Nueva York:

McGraw-Hill.

Spillner, A., & Linz, T. (2014). *Software Testing Foundations: A Study Guide for*

*the Certified Tester Exam* (4th ed.). Rocky Nook.

Tsui, F., Karam, O., & Bernal, B. (2018). *Essentials of software engineering* (4th ed.). Jones &

Bartlett Learning. Retrieved from <https://platform.virdocs.com/r/s/0/doc/581911/sp/175232049/mi/563314392>.