Acceptance Test Plan

For

Dragon Course Scheduler

Prepared by

Nathan Gelfant,
Kevin Huang,
Stan Kolakowski,
Mark Scheid

At

Drexel University

Acceptance Test Plan Revisions History

Date	Description	Version	Editor(s)
2/9/13	Document created.	1.0	Mark Scheid, Nathan Gelfant, Kevin Huang, Stan Kolakowski
2/10/13	Grammar/ format.	1.1	Mark Scheid, Nathan Gelfant, Kevin Huang, Stan Kolakowski

Contents

1. Introduction

1.1 Background

This document provides the process to complete the acceptance test for the Dragon Course Scheduler. The Dragon Course Scheduler is a tool designed to help students plan their schedule with an eye to their graduation tracks, and multi-term planning. The process is more thoroughly described in the Software Requirements Specification (SRS) which details specific requirements and functionality of the software.

1.2 References

- Software Requirements Specification for Dragon Course Schedule (Current version 1.1)

1.3 Definitions

Browser A piece of client side software, such as Firefox or Internet Explorer, capable of displaying web pages written in HTML, CSS and Javascript code, as well as running java applets.

Class Selection Interface: A web page through which the user views the list of available classes for the selected term based upon their previously entered courses.

Concentration: Also referred to as "tracks", will be major specific information required or defined by Drexel.

Database: A database is a structured collection of data that is organized for later retrieval.

Dependant classes: Classes which require the course in question as a prerequisite.

Functional Requirements - Define the internal workings of the software.

Non-Functional Requirements - Specifies criteria that can be used to judge the operation of a system, rather than specific behaviors.

Prerequisites table: A table in the database that is populated based on Drexel Course Catalog, detailing what courses are necessary before one can register for a given class.

Program: The java backend that manages data flow between the interfaces and the database.

Selected Schedule Interface: A web page through which the user views their current weekly plan of classes for the selected term.

Server: A content hosting system that will deliver content to end users and host aggregated information.

Tester: A person who will execute the use case tests.

Test Team Leader - Person responsible for managing the testers and the test plan.

TMS Synchronizer: The server-side tool that automatically updates the database with available course information and major concentration definitions.

Unit Tests - A method by which individual units of source code are tested to determine if they are working properly.

User-Background Interface: A web page through which the user can input their major, concentrations, and the list of classes they have already taken.

User-History Object (UHO) The user history object contains a data structure capable of holding information containing all previous classes completed by the user, as well as the currently selected list of classes for the scheduled terms.

1.4 Abbreviations

DCS Dragon Course Scheduler
CRN Course Reference Number
JRE Java Runtime Environment
JVM Java Virtual Machine
SRS Software Requirements Specification
TBD To be dated/decided
TBA To be announced
TMS Term Master Schedule
UHO User-History Object

2. Test Approach

2.1 Overview

This section describes the objectives and structure while outlining constraints for the Dragon Course Schedule software.

2.2 Test Objectives

The main objective of the Acceptance Test Plan for Dragon Course Scheduler is to verify that the software and GUI meets requirements and functionality as specified in the SRS. The Acceptance Test Plan's main objective is to provide the criteria to determine if the project has been successfully brought up to the standards specified by the SRS.

2.3 Test Structure

Each test cases will be constructed in seven sections, which includes: ID number, name, SRS reference location, precondition(s), action descriptions, and postcondition(s). Upon a successful test, the tester will be able to verify expected behavior, given said conditions and actions.

3. Test Assumptions and Exclusions

3.1 Overview

This section goes more in depth about which functions and features of Dragon Course Scheduler will be covered by the Acceptance Test Plan process, and which functions and features will not be covered. It is presumed that these tests are to occur as a formal way to confirm the program has been completed to specification, having already been verified as completely as possible through Unit tests, Integration tests, and System tests of the program.

3.2 Assumptions

The Acceptance Test Plan will cover:

- External Interface Requirements of the software, as listed in the SRS
- Functional Requirements of the software, also listed in the SRS

3.3 Exclusions

The Acceptance Test Plan will not cover:

- Non-Functional Requirements of the software, outlined in the SRS
- Structural integrity of the source code.

4. Entry and Exit Criteria

4.1 Overview

This section lists the criteria which must be satisfied in order for the Acceptance Test Plan to begin, as well as the criteria which must be satisfied in order for the Acceptance Test Plan to be considered finished.

4.2 Entry Criteria

The Acceptance Test may commence after the following conditions have been met:

· All other unit tests, system tests, and integration tests are complete.
· A proper environment that meets Software Requirements outlined in the SRS is available.
· A copy of the latest version of the SRS has been received and reviewed.
· The latest version of Dragon Course Scheduler has been compiled and installed.
· Verification from each group member that the program is up to standards set in the SRS to their best knowledge.

4.3 Exit Criteria

The Acceptance Test Plan will be considered successful if after all Priority 1 requirements are tested and verified to be work as intended.
The Acceptance Test Plan will be considered a failure if any Priority 1 requirements are tested and verified to be not be working as specified.

5. Testing Participants

5.1 Overview

This section describes the roles and responsibilities of all the group members involved in the Acceptance Test Plan as well as the steps taken when a problem surfaces during testing.

5.2 Roles and Responsibilities

For the Acceptance Test Plan the roles and responsibilities are as follows:

Test Team Leader: Stan Kolakowski

Tester 1: Nathan Gelfant

Tester 2: Mark Scheid

Tester 3: Kevin Huang

5.3 Training Requirements

All parties involved with the Acceptance Test Plan of the Dragon Course Scheduler should be familiar with the DCS interfaces, system documentation, as well as the Software Requirements Specifications.

5.4 Problem Reporting

Any problem identified by the tester must be documented in the issue tracking system and reported to the Test Team Leader. The problem will be assigned to a tester for review before the next team meeting.

5.5 Progress Reporting

All progress should be made through the issue tracking system Jira. Meeting roadmap and milestones will be recorded in the group management gantt chart, and on the Wiki page of the Git repository.

6. Test Cases

6.1 Enter Major

ID	TC1
Name	Enter Major
Reference(s)	3.2.1
Description	The tester enters their major information
Precondition(s)	1. The tester opens the Dragon Course Scheduler website page 2. The tester entered major selection mode
Action(s)	1. The tester clicks the dropdown menu for major selection 2. The tester chooses a major
Post Condition(s)	Upon successful execution: 1. The Concentration list will reflect those of the selected major 2. The Courses taken input box is available to receive input Upon failed execution:

	1. The Concentration list may not highlight selected major 2. The course taken input box may not take any input

6.2 Enter Concentration Selection

ID	TC2
Name	Enter Concentration Selection
Reference(s)	3.2.1
Description	The tester enters concentration information
Precondition(s)	1. The tester has entered the major information
Action(s)	1. The tester selects concentration(s) in a list of concentrations available by clicking different concentration(s) 2. The tester selects a term 3. The tester advances to the Class Selection Interface
Post Condition(s)	Upon successful execution: 1. The Class Selection Interface displays the full list of courses required for the specified concentration in the “Concentration Requirements” part of the display. Upon successful execution: 1. The Class Selection Interface displays an incomplete or missing list of courses required concentration requirements in the “Concentration Requirements” part of the display.

6.3 Enter Completed Courses

ID	TC3
Name	Enter Completed Courses
Reference(s)	3.2.1
Description	The tester enters completed courses information
Precondition(s)	1. The tester has entered major information 2. The tester has completed some courses before

Action(s)	1. The tester enters a complete list of approved courses 2. The tester will select a term 3. The tester advances to the Class Selection Interface
Post Condition(s)	Upon successful execution: 1. The tester will be able to see a properly filtered list of classes based on this history Upon failed execution: 1. The tester will be able to see courses he has already input, or courses he lacks the prerequisites for.

6.4 Add Class

ID	TC 4
Name	Add Class
Reference(s)	3.2.2
Description	Add a class to the schedule.
Precondition(s)	1. The tester has filled out his major, concentrations, and course history. 2. The tester has selected more than 1 term. 3. The tester has advanced to the Class Selection Interface. 4. The tester has selected a class to add that has a dependant class offered in a later selected term.
Action(s)	1. The tester clicks the CRN of the class to be added. 2. The tester reviews the current term to evaluate Post Condition 1 3. The tester clicks the menu and changes to the subsequent term
Post Condition(s)	Upon successful execution: 1. The Class Selection Interface will be updated by removing other selections of the class and conflicting timeslots. 2. The Class Selection Interface will show the dependant courses as options for the tester. Upon failed execution: 1. The Class Selection Interface displays duplicate courses or conflicting classes 2. The dependant courses fail to display as expected.

6.5.1 Remove Class

ID	TC 5.1
Name	Remove Class A
Reference(s)	3.2.3
Description	Remove a class from the schedule.
Precondition(s)	1. The tester is viewing the Class Selection Interface
Action(s)	1. The tester clicks the course id on his schedule.
Post Condition(s)	Upon successful execution: 1. The program will warn the tester of potential changes of the targeted class were to be removed. Upon verifying the removal, the program will update the schedule. Upon failed execution: 1. Does not remove class from the current term schedule and/or the pop for confirmation pop up will not show to ask for deletion confirmation.

6.5.2 Remove Class

ID	TC 5.1
Name	Remove Class B
Reference(s)	3.2.3
Description	Remove a class from the schedule.
Precondition(s)	1. The tester is viewing the Selected Schedule Interface
Action(s)	1. The tester clicks the course id on his schedule.
Post Condition(s)	Upon successful execution: 1. The program will warn the tester of potential changes of the targeted class were to be removed. Upon verifying the removal, the program will update the schedule. Upon failed execution: 1. Does not remove class from the current term schedule and/or the pop for confirmation pop up will not show to ask for deletion confirmation.

6.6 Change Selected Term

ID	TC 6
Name	Change Selected Term
Reference(s)	3.2.4
Description	Change the term that is being viewed in the Selected Term Schedule or Class Selection Interface.
Precondition(s)	1. The tester is viewing the Class Selection Interface or the Selected Schedule Interface
Action(s)	1. The tester selects “Change Term” from the drop down menu in the top right corner of either interface and selects desired term.
Post Condition(s)	Upon successful execution: 1. The tester now sees the Selected Schedule View of the selected term Upon failed execution: 1. The tester does not see the Selected Schedule Interface of selected term

6.7 View Schedule

ID	TC7
Name	View Schedule
Reference(s)	3.2.5
Description	Switch from Class Selection Interface to Selected Schedule Interface upon tester request
Precondition(s)	1. The tester is viewing the Class Selection Interface
Action(s)	1. The tester clicks the menu button and selects View Schedule from the drop down list
Post Condition(s)	Upon successful execution: 1. The tester now sees the Selected Schedule Interface Upon failed execution: 1. The tester does not see the Selected Schedule Interface

6.8 Select Classes

ID	TC8

Name	Select Classes
Reference(s)	3.2.6
Description	Switch from Selected Schedule Interface to Class Selection Interface upon tester request
Precondition(s)	1. The tester is viewing the Selected Schedule Interface
Action(s)	1. The tester clicks the menu button and selects Select Classes from the drop down list
Post Condition(s)	Upon successful execution: 1. The tester now sees the Class Selection Interface Upon failed execution: 1. The tester does not see the Class Selection Interface

6.9 Print Schedule

ID	TC9
Name	Print Schedule
Reference(s)	3.2.7.1
Description	Send the tester schedule to the printer
Precondition(s)	1. The tester is viewing the Selected Schedule Interface
Action(s)	1. The tester clicks the menu button and selects Print schedule from the drop down list
Post Condition(s)	Upon successful execution: 1. The tester's browser prompts him to select a printer, then generate an accurate representation of the course info list and schedule Upon failed execution: 1. The tester is unable to print an accurate representation

6.10 Export Schedule to CSV

ID	TC10
Name	Export Schedule to CSV

Reference(s)	3.2.7.2
Description	Generate a CSV file for the tester
Precondition(s)	1. The tester is viewing the Selected Schedule Interface
Action(s)	1. The tester clicks the menu button and selects Export to CSV from the drop down list
Post Condition(s)	Upon successful execution: 1. The tester's browser prompts him to save a file, which contains an accurate list of courses in CSV format Upon failed execution: 1. The tester is unable to save an accurate CSV file