

Executive Summary

These documents provide a review of the Technical Specification Document and the Conceptual Design Document provided by Puzzle™. The aforementioned documents constitute a complete description of the Course Scheduling Application that Puzzle™ will implement, and will serve as a contract for its completion.

The Conceptual Design Document has been included, and it contains our comments on the proposed system.

Also included is a test plan. This plan will be used to verify that the delivered system meets the requirements, as laid out in these documents.

Test Plan

Test # 1: Registration

This test involves going through the steps of creating an account. The account should be created and saved. The user should be able to log out and then log back in, and the account and all of the information should still exist and be unchanged.

This will be tested by registering for the service, logging out, and logging back in to verify that the account was created.

Test #2: Program requirements

When a program is chosen, the “required courses” field becomes populated with a list of courses. This list should accurately correspond to Uvic’s requirements for the program.

This will be tested by choosing a program during account registration, then validating it against a list of Uvic’s course requirements for that program.

(For the prototype, it is understood that complete course data will not be available. The “required courses” field will be validated against a limited list of courses).

Test #3: Customizable schedule

In the schedule view, a class can be manually dragged by the user into a different term. If the move is valid, the schedule should immediately be saved.

This will be tested by moving a class, then refreshing the webpage.

Test #4: User-caused schedule conflict

If a class is moved by the user into another term, and this violates one of the schedule’s rules, a visible warning message should pop up. The user should not be able to interact with the rest of the page until the dialog box has been dismissed.

This will be tested by creating a schedule conflict, observing the application’s response, and then trying to interact with the page before dismissing the dialog box.

The following types of scheduling conflicts will be tested:

- a. Prerequisite violation
Two courses will be added to the schedule, one being the prerequisite of the other. The higher-level course will be placed in an earlier term than its prerequisite.
- b. Term offering violation
A course that is known not to be offered next term will be added to the schedule next term.

- c. Enrolment limit violation
Seven full-credit courses will be added to a single term.

Test #5: Schedule creation algorithm

This test will verify the automatic schedule creation algorithm, a core aspect of the product. The generated schedule should be valid and conform to all of Uvic's course requirements, such as prerequisites and enrolment limits. The schedule should conform to the user's stated preferences as much as possible. It should prioritize the University's requirements above the user's, and prioritize the user's requirements according to a weighting given by the user. The system should inform the user when their requirements could not be met, and indicate which ones.

This will be tested by clicking the "Fit Courses" button while there are unscheduled courses in the list. Several tests of this feature will be conducted, with varying course lists. These will include:

- a. Courses with multiple prerequisites.
- b. Courses that are a prerequisite of multiple other courses.
- c. Courses that are only offered one or two terms per year.

The testing will involve varying the user-defined preferences about the schedule. At least one test input will be given that has no valid schedule that meets the user's requirements, and at least one test input will have no valid schedule that fits the University's program requirements.

Test #6 Course overrides

This test will verify that unique situations like course overrides can be handled by the system.

This will be tested by adding a course to the schedule without adding its prerequisite.

The expected behaviour of the system is to warn the user that the course has a prerequisite that hasn't been included in the schedule. The user should then have an option to proceed with the schedule as-is.

Test #7: Saving User Data

As Test #1, but this test will be done after a schedule has been created using the application. This will verify if the user's schedule is being saved.

This will be tested by creating a user account, creating a new schedule, then logging out and logging back in.