

Assessment of the prototype demo

Team 5 - Just Right Showers

Rhiannon Tully-Barr

Kathleen Garland

Kushal Patel

Ushanth Loganathan

Assessment of the prototype demonstration by Team 14 - Puzzle

As detailed in the requirements specification, the demonstrated prototype provides a way to plan a degree program at Uvic on a per-term basis. The majority of the features from the requirements specification were present and shown in the demo.

The demo contained a list of software engineering courses. As mentioned in the requirements specification, this minimal system / proof of concept focuses on the software engineering program, to be expanded to the whole University curriculum in the future.

The courses can be dragged into the term lanes and re-arranged by dragging them.

If a course is placed in a term where it is not typically offered, the course becomes red in colour to indicate that there is a problem. An error message is shown by clicking on the course. This error detecting functionality is present for courses which are only offered e.g. in the spring and summer.

Part of this requirement is missing from the demo: the system does not detect errors due to prerequisites, for example if CSC 225 is moved to before the term with CSC 226, the system will not detect the problem.

The core functionality discussed in the requirements is present in the prototype. The prototype has a feature called "fit unscheduled courses" which takes all the courses in the sidebar (which haven't been placed in the schedule yet) and creates a valid course schedule. This schedule is subject not only to the requirements put forth by the university, but also the user's stated preferences.

When the user fits the unscheduled courses, a prompt appears in which the user can set two different parameters: Target graduation date and desired courses per term.

As a whole, the prototype does a great job of meeting the agreed-upon specifications. There are a handful of features that are incomplete, particularly error detection, but the functionality that is there suggests that these could be implemented in a timely manner. The minimal system is a successful proof of concept.

For future implementations, extending the system to handle all Uvic courses and programs, and potentially other universities, would be a plus. Integration with Uvic's systems and the ability to login with one's netlink ID would greatly improve the system, making it easier to pre-populate

course requirements. Web scraping is also an option for this feature, as mentioned in the demo. Providing more user-defined preferences for the automatic fit algorithm, and a way to prioritize them, would be an improvement as well.

Test cases:

1. Registration
This test case was not done in the demo, because an already-existing test user account was used. However, the functionality was present in the demo.
2. Program requirements
Scraping data directly from uvic.ca was not part of the minimum viable product, so automatically populating the user's list with courses was not possible. This was not tested in the demo.
3. Customizable schedule
This test was performed in the demo, and it succeeded admirably. The drag-and-drop interface is flexible, and meets the requirements well.
4. User-caused schedule conflict
This test was performed for the case where a course is placed in a term where it is not usually offered. The example data used was the course CSC 360, and it was placed in the fall although it was only offered in the spring and summer. The course became outlined in red to indicate the error, which was a great error-reporting technique that does not overly interrupt or restrict the user. The highlighting went away when the course was moved to the correct term.
This behaviour should extend to other conflicts such as prerequisite violations, but that functionality is not yet implemented in the prototype.
5. Schedule creation algorithm
The algorithm was tested during the demo with a predetermined list of courses. The algorithm successfully placed courses in the term lanes, producing a valid course schedule. The sample course list did contain courses that were only offered 1-2 terms per year, and courses with prerequisites.
6. Course overrides
This test case was not performed in the demo.
7. Saving user data
This test case was performed in the demo. The user logged out and then logged back in, and the data remained unchanged. In addition, the "undo" history was preserved even when the user logged out, which was a very nice touch.
8. User experimentation
User testing was not done during the demo, and we are not aware of whether the developer did any user testing prior to the presentation (there was a limited amount of time available for questions).