CourseForge

Chris Schlechty, Kenneth Kuan, Scott Clifford, Guanyu Chu,
Kansu Dincer, Sarah Tachibana, Andy Hou

Software Requirements Specification

Draft 1.0
4/17/2007

CSE 403 - CSRocks Inc.

Revisions

Version	Primary Author(s)	Description of Version	Date Completed
1.0	All	Final	4/17/07

Overall Description

Description

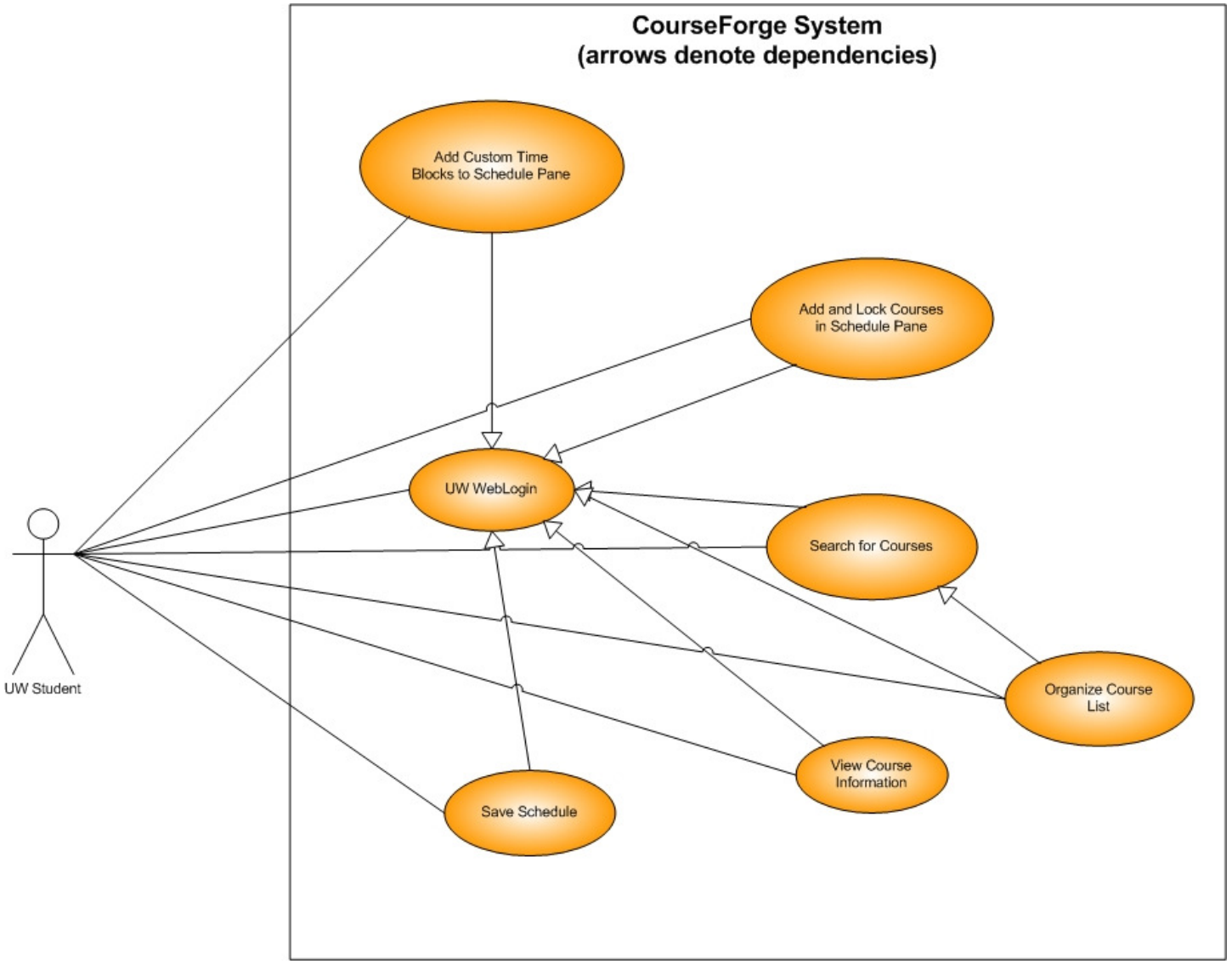
CourseForge is a web-based system for UW students which will greatly improve the UW course registration experience. At its core, CourseForge provides an easy visual way of planning schedules. It will provide numerous conveniences that the current system does not. At present, students are required to search through a maze of confusing and disjointed pages to gather all available registration information, and they can easily miss something which would have an impact on their decision to take/not take a course. Our system will combine the intuitiveness of a visual schedule with features that put schedule control at the student's fingertips, such as detailed course info and rankings, comprehensive course searching, comparing multiple schedules using tabs and "ghosting", and more into one easy, fluid, comprehensive system that makes registration research simple.

Scope

CourseForge is specialized for creating and comparing course schedules. It is not meant to be used as a replacement for web-based calendar applications such as Google Calendar. CourseForge will save a user's schedules for use over multiple sessions, but it will not have certain calendaring features like viewing more than one week (since classes always repeat every week), or creating your own activities (other than marking blocks of time as busy). Additionally, CourseForge is not a long-term academic planner for mapping out a student's entire college career. CourseForge will only plan for the current academic quarter.

CourseForge will need to be able to read or scrape UW's existing course schedule database in order to get current course information. It will also depend on its own SQL database for storing course info, and information for each user like the schedules that they've saved. CourseForge will also depend on LAMP (Linux, Apache, MySQL, [Perl, PHP, or Python]) for getting user input in the web browser, communicating with the server, and displaying the results in the web browser.

Use Cases


CourseForge System
(arrows denote dependencies)
Add Custom Time Blocks to Schedule Pane
Add and Lock Courses in Schedule Pane
UW WebLogin
Search for Courses
Organize Course List
View Course Information
Save Schedule
UW Student

Goal	Once a client submits a search query to the interface, a course list should be populated and presented to them.
Level	Sub-function
Primary Actor	UW Student
Precondition	Client is already logged into the registration system.
Success end condition	A course list is populated for the client.
Failure end condition	A course list is not populated for the client.
Trigger	Registration time rolls around at UW.
Main success scenario	1. Client inputs information into a search field category(s). 2. Client submits their query by clicking a confirmation button. 3. System queries database using client's input. 4. System returns a list of courses to the interface.
Extensions	1a. No information is entered by client. 1a1. System displays an error dialogue. 2a. The client has no internet connectivity when they submit. 2a1. System does nothing and normal web browser behavior ensues. 3a. System cannot resolve the user's input after querying database. 3a1. System displays 'no courses found' in the list results. 3b. System finds user's input, but the class is not offered currently. 3b1. System displays 'course not offered during current quarter'.
Variations	1-2'. Client submits a DARS instead of search. 1-2''. System parses DARS and populates course list.

Goal	To add a course to the schedule pane and to lock it.
Level	Sub-function
Primary Actor	UW Student
Precondition	Client is logged in and the course list is populated.
Success end condition	A course is locked in the schedule pane.
Failure end condition	A course is not locked in the schedule pane.
Trigger	The course list pane is populated after a search query is performed.
Main success scenario	1. Client drags and drops a course from the course list pane to the schedule pane. 2. System generates “ghosted” boxes everywhere the course is scheduled to meet for class during the week. 3. Client left-clicks on the “ghosted” course in the schedule pane. 4. System locks the course the client clicked on.
Extensions	1a. Client drags and drops the same course again. 1a1. System does nothing. 3a. Client attempts to lock a ghosted course, but there is a conflict with another locked course. 3a1. System displays warning stating the course could not be locked because of a course conflict.
Variations	

Feature List

Feature	Target
Visual schedule pane with ghosting/locking	Alpha
Ability to add classes to schedule pane from list pane	Alpha
Search for courses by keyword	Alpha
Search for courses by specified time intervals	Alpha
Usable database with schema and data	Alpha
Course information pop-ups for class blocks	Beta
UW course directory scraper	Beta
UW professor info scraper	Beta
Authentication via login module	Beta
Search for courses by credit type	Beta
Search for courses by course level	Beta
Search for courses by number of credits	Beta
Build multiple schedules via tabbing	Final
Ability to save current schedule configuration on server	Final
List pane organizer	Final
Addable custom blocks to visual schedule	Final
Add code functionality	Final
Auto-register	Final
Populate course list via DARS/Transcript parsing	Stretch
View course routes via mapping	Stretch
Export schedule to local drive for external viewing	Stretch
Alternate schedule visualization	Stretch

UI Prototype

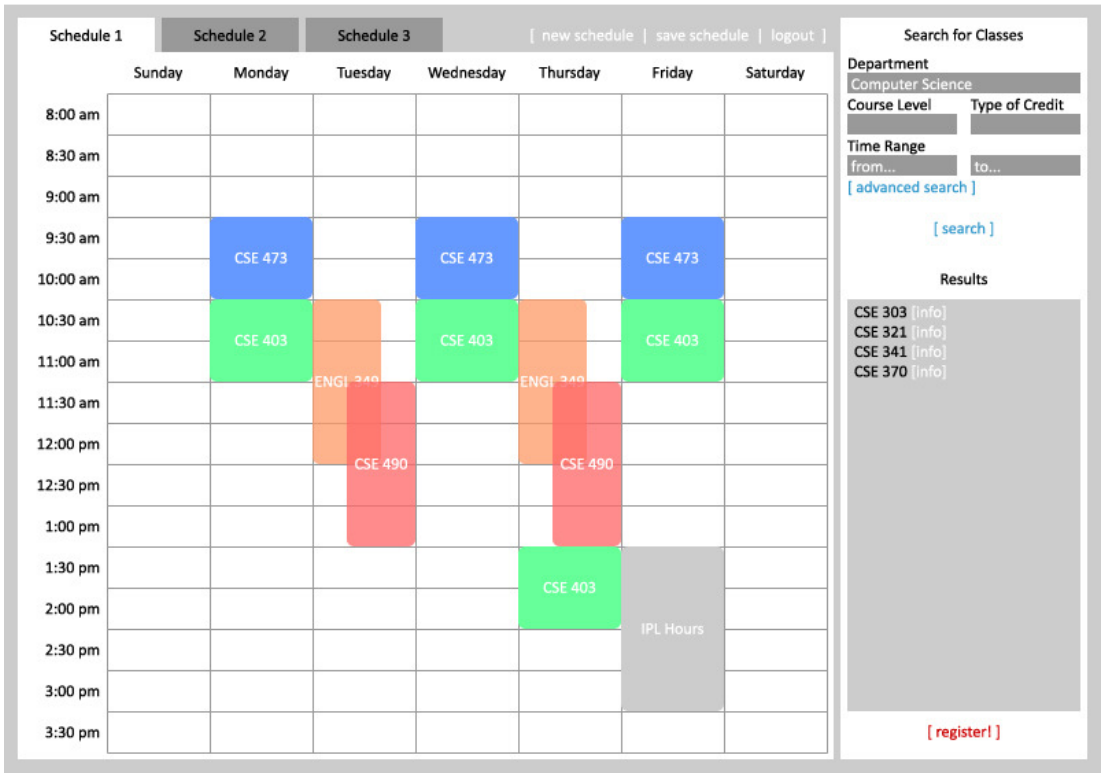


Fig. 1: Main page. The visual schedule pane on the left is already populated by a potential schedule with “ghosted” classes (red and orange), locked classes (blue, green), and a custom commitment block (gray). “Ghosted” classes can overlap. Tabs at the top of the schedule pane indicate alternate schedule options.

The list pane on the right contains search fields, search button, and course list. The register button below the list pane is available when the student is ready to register for their schedule.

UI Prototype

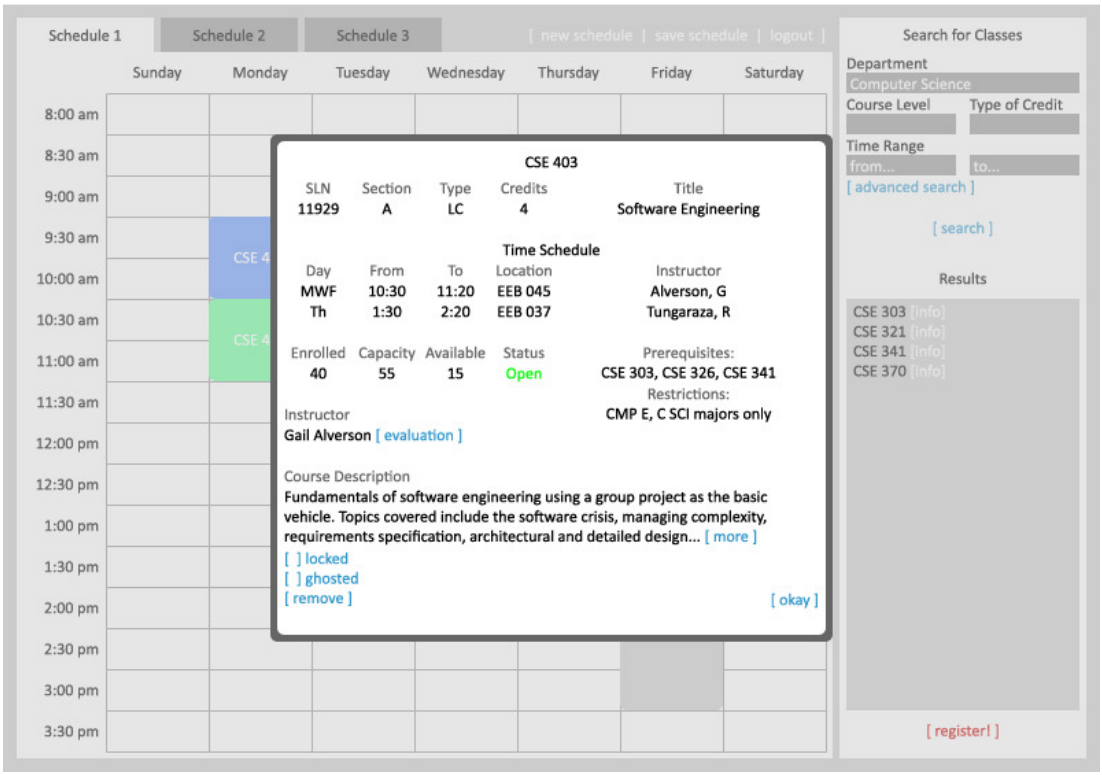


Fig. 2: Info popup. This displays all relevant info about a class when the class is double-clicked on.