

Visual Schedule Finder

Peter Beckfield

My Cam

Rylan Hawkins

Jordan Hoyt

Sandra Li

Devy Pranowo

Software Requirements Specification

Draft 1

April 15, 2008

CSE 403 - CSRocks Inc.

Revisions

Version	Primary Author(s)	Description of Version	Date Completed
1	Sandra Li, Jordan Hoyt, My Cam, Peter Beckfield, Devy Pranowo, Rylan Hawkins	Sandra and Jordan added content for Overall Description of project, including high-level description and scope. Also added features, as discussed with Jordan. My and Peter added their use cases. Devy added her UI mockups. Rylan integrated and did the final edits on the document.	04/15/08

Overall Description

Description

The software we will produce is a Web application called the *Visual Schedule Finder*, which is an easy to use online software tool whose primary purpose is to help college students at the University of Washington choose which courses to register for by providing them a clear representation of courses and time scheduling information. Our product will act as a preliminary tool for a student to assist them in the stage prior to the actual registration period for a particular quarter. Based on what classes a student have taken already, our system will generate a list of courses from which the student can choose. From there, the student can select desired courses, and then add them to a visual calendar.

The general idea behind the system is to generate a list of courses that are viable options for the next quarter. This list called the “My Course List” will be persistent across all sections of the application. The user will generate this list through several different methods. Primarily, the user will add courses from a list of possible classes for a student to choose from, including classes that not only help the student progress toward their major, but also those that will help the student fulfill general education requirements. The list will be generated by asking about what the student’s major is, or intended major, if undecided and having the user fill out their class history, including what classes the student has taken. This profile will also be saved as part of their account so it will not have to be re-specified upon each use of the tool. The user can then login to the system at a different time and retrieve course history information.

The “My Course List” contains not only the times for the lectures, but also the times for any associated lab sessions or quiz sections. These specific classes can then be selected for viewing in a visual calendar. The visual schedule is where any time conflicts appear as a result from the student’s course selections. The student can then finalize the process after ensuring all classes fit their time constraints, by retrieving a list of SLN numbers.

Furthermore, the system will support an admin interface for backend course details and degree requirements to be edited.

This web application will not actually complete the registration process. Also, our tool will not find out when a user’s specific registration date is, nor will it restrict the number of credits a student can register for, though it will keep a running list of credits, based on selected classes. It will not automatically determine a student’s course history, as he/she will need to do that when he/she creates a profile.

Our main objective for this Web application is to ease the registration process for students by helping them figure out their schedules prior to the registration date. Our goal is to make the registration process faster and smoother for our target audience of University of Washington students.

Some benefits of this application include convenience to the student, as he/she can find out what classes he/she needs to take, as well as what classes will fit the time constraints by having the ability to see a visual schedule as he/she adds and/or removes classes. It also removes confusion about what classes a student can take, since our system will figure all that out, based on what classes the student has already taken, so the student just needs to choose from a list of classes presented to him/her. The system should serve as a full vertical solution for students to enter with almost no precondition and leave all but registered.

Scope

As aforementioned, our system is a preliminary tool that facilitates the registration process for UW students. That said, it will not actually complete the registration process for the student, nor will it alert the student to when his/her actual registration date is. This tool is also not responsible for limiting the number of credits a student chooses, as it will rely on the student's own judgment. It will not give real-time information about the enrollment statistics of a class, either, as it will only be refreshed at certain periods of time. The data will only be as current as a snapshot of the University of Washington database for course information.

The system will support the following:

Operating Systems (OS)

- Mac OS X 10.5 and later
- Microsoft Windows XP and later

Browsers

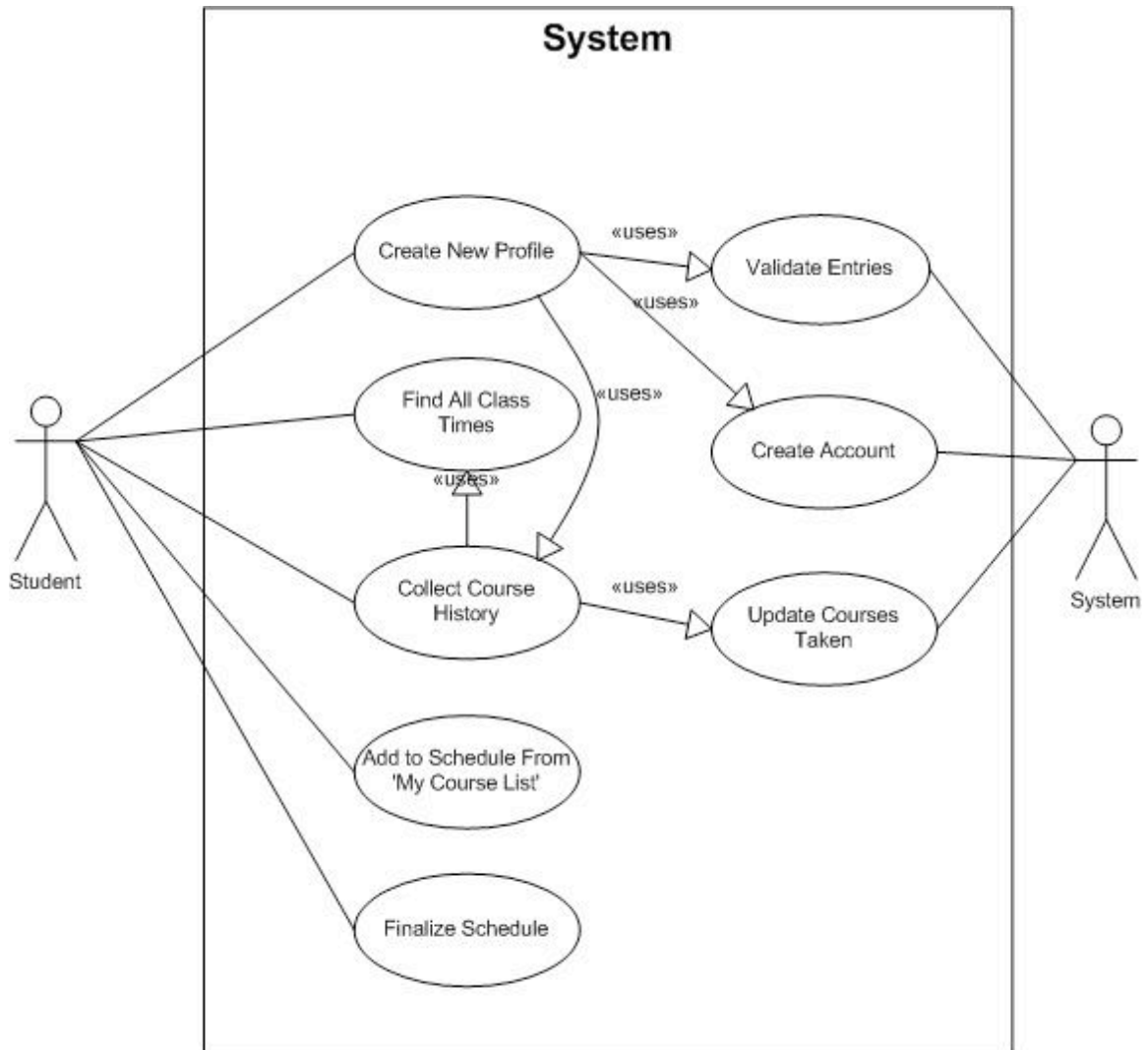
- Firefox 2
- Internet Explorer v7
- Safari 3

Other Requirements

- JavaScript must be installed and running
- Best viewed with 1024x768 or better resolution

Use Cases

Use Case Summary Diagram



Use Case (Find All Class Times)

Goal	Student wants to find all available time for a specific class
Primary actor	Student
Scope	Vis Schedule Finder (VSF)
Level	User
Precondition	Student has logged in.
Trigger	Student logs into system
Success end condition	Class times are found
Failure end condition	Class times are not found
Main Success Scenario	<ol style="list-style-type: none">1) Student enters desired course by course number (i.e. CSE 403)2) Student Clicks "Search"3) System searches record (database) for inquired course4) System list found result5) Student selects a class time from list to add to <i>My Course List</i>6) System adds selected class to <i>My Course List</i>
Extensions	<ol style="list-style-type: none">3a) System cannot find inquired course<ol style="list-style-type: none">3a.1) System alerts student and asks student to enter another course
Variations	<ol style="list-style-type: none">1a) Student searches by course name/keyword

Use Case (Create New Profile)

Goal	Student creates his/her own VSF account
Primary actor	Student
Scope	Vis Schedule Finder (VSF)
Level	User
Precondition	Student is at the log in screen
Trigger	Student is at the log in screen
Success end condition	Student profile created
Failure end condition	Student profile not created
Main Success Scenario	<ol style="list-style-type: none"> 1. Student clicks "Create New Account" 2. System displays page on screen requesting the following information <ol style="list-style-type: none"> a. First & last name, middle initial b. Birthday c. Year in college d. Major or intended major 3. Student enters required information 4. Student clicks "Submit" 5. System validates student entries (use case "Validate Entries") 6. System creates student record in database (use case "Create Account") 7. System displays Degree Requirement page (use case "Collect Course History")
Extensions	<p>5a) Some requested information omitted</p> <ol style="list-style-type: none"> 5a.1) System asks user to enter omitted information 5a.2) System goes back to step 2 <p>5b) Some requested information in invalid form</p> <ol style="list-style-type: none"> 5b.1) System asks user to enter information in valid form 5b.2) System goes back to step 2 <p>6a) System cannot create account, displays error message on screen, notifies user to submit again</p>
Variations	None

Use Case (Collect Course History)

Goal	Student stores his/her degree course history in profile
Primary actor	Student
Scope	Vis Schedule Finder (VSF)
Level	Summary
Precondition	Student has an account with VSF
Trigger	Student logs into system
Success end condition	Student course history is captured in database
Failure end condition	Student cannot enter information
Main Success Scenario	<ol style="list-style-type: none"> 1. System displays Degree Requirement page (list of classes), according to student's major, on screen 2. Student indicates courses they have completed thus far 3. System saves information in database (use case "Update Courses Taken") 4. System calculates a list of courses that student has fulfilled prerequisites for 5. System displays the list 6. From this list student choose classes to take and add to My Course List (use case "find all class times")
Extensions	3a) System cannot update profile information, return error
Variations	6a) Student choose to add classes not on list

Use Case (Add to Schedule from 'My Course List')

Goal	Student fills visual schedule with classes
Primary actor	Student
Scope	Vis Schedule Finder (VSF)
Level	User
Preconditions	Student is logged into VSF Student has at least one class added to their 'My Course List'
Trigger	Student clicks on course to add to schedule
Success end condition	System updates visual schedule based on new class added
Failure end condition	System is unable to update visual schedule
Main Success Scenario	<ol style="list-style-type: none"> 1. Student designates a class to be taken in their 'My Course List' by selecting a specific section 2. System adds the selected course to the visual schedule
Extensions	<p>2a) A different section of the same class already exists in schedule</p> <p style="padding-left: 20px;">2a.1) System removes old section and replaces with new</p> <p>2b) There is a time conflict with the new class and an existing one</p> <p style="padding-left: 20px;">2b.1) System highlights the conflict visually</p> <p>2c) There are too many time conflicts in a given slot, and the visual schedule is unable to display all of them usefully</p> <p style="padding-left: 20px;">2c.1) System does not add course to visual schedule</p>
Variations	

Use Case (Finalize Schedule)

Goal	Student receives a useful representation of their chosen schedule
Primary actor	Student
Scope	Vis Schedule Finder (VSF)
Level	User
Preconditions	Student has an account with VSF Student has at least one class added to their visual schedule, and has no overlapping classes
Trigger	Student asks for output
Success end condition	Student gets SLNs for their schedule
Failure end condition	Student does not receive SLNs for chosen schedule
Main Success Scenario	<ol style="list-style-type: none"> 1. Student asks System to output their schedule 2. System checks to make sure the schedule is valid 3. System creates a document containing pertinent schedule information 4. System asks Student to download the document
Extensions	<p>2a) Schedule is invalid</p> <p style="padding-left: 40px;">2a.1) System informs user that schedule is invalid, asks user to fix schedule</p> <p>5a) List is too long.</p> <p style="padding-left: 40px;">5a.1) System display a threshold # of classes on first view</p> <p style="padding-left: 40px;">5a.2) Upon clicking "next" Student can view the next set of classes (not exceeding threshold #)</p> <p style="padding-left: 40px;">5a.3) repeat 5a.2 until no more classes</p>
Variations	<p>2a) System informs user that schedule is invalid, but allows user to receive output</p> <p>3a) System creates a web page with pertinent information</p> <p>4a) System directs user to the web page created in variation 3a</p>

Feature List

Persistent student accounts	Beta
Complete list of UW courses	Beta
Filter list of courses	Beta
View associated sections/labs for lectures	Beta
View course descriptions	Beta
Keep a "My Course List" possible courses considered	Beta
Add/delete from "My Course List"	Beta
Display class times on visual calendar	Beta
View overlapping times on calendar	Beta
Supports super administrator for modifying time schedules and/or course listings	Final
Supports administrator role for department head to alter degree requirements	Final
Search through classes by keyword/name/course number	Final
Disable ability to add classes that student cannot take	Final
Generate final list of SLNs for selected classes	Final
View map to find location of classes	Final
View class enrollment & class capacity statistics	Stretch
Specify time preferences / outside obligations on visual calendar	Stretch
Generate calendar where classes respect additional time constraints	Stretch
Create a demo/tour of our Web application	Stretch

Visual schedule finder

My Course List

- CHEM 142 : GENERAL CHEMISTRY
 - A MWF 9:30-10:20
 - AA T 9:30-10:20
 - AB Th 9:30-10:20
 - AC Th 10:30-11:20
 - B
- TC 333 : TECHNICAL WRITING
 - B T Th 1:30-2:30

Find Courses

Filter Search

College of ...

Department...

Search result

TC 333 : Technical Writing

Add to My Course List

More...

TC 341 : ...

Degree Requirement

College of Art and Science

Computer Science



pdf version

Senior Electives (20 credits)

Required (35 credits)

- CSE 142
- CSE 143
- CSE 303
- CSE 322 Add
- CSE 326
- CSE 341 Add
- CSE 370
- CSE 378 Add

- CSE 401
- CSE 403 Add
- CSE 421
- CSE 421 Add
- CSE 444
- CSE 451
- CSE 455
- CSE 457
-

Next

Chem 142 : General Chemistry
 description
 location — link to campus map
 Instructor
 Capacity : 95/100
 VLP/ALS / ...

Language Skill (5-20 Credits)

- TC 333
-

Reasoning (15 credits)

- MATH 124
- MATH 125
-

Areas of Knowledge

- VLP/A
 -
- I/S
 -
- Natural Science
 - CHEM 142

Visual schedule finder

My Course List

- CHEM 142 : GENERAL CHEM
- A MWF 9:30-10:20
- AA T 9:30-10:20
- AB Th 9:30-10:20
- AC Th 10:30-11:20
- B ⊖ remove
-
- TC 333 : TECHNICAL WRIT
- B T Th 1:30-2:30 ⊖ remove

Find Courses

Filter Search

College of ... ▾

Department... ▾

Search Result

TC 333 : Technical Writing

➕ Add to My Course List

More...

TC 341 : ...

Degree Requirement

Visual Schedule

Monday Tuesday Wednesday Thursday Friday

Chem 142 : General Chemistry
 description — link to campus map
 location —
 instructor —
 capacity : 95/100
 VLPA/IS / ...

MUSIC 120
Gug 211

MUSIC 120
Gug 211

MUSIC 120
Gug 211

MUSIC 120
Gug 211

MUSIC 120
Gug 211

11 a.m
12 p.m
1 p.m
2 p.m
3 p.m
:
:
:

TC 333
Loew 121

TC 333
Loew 121

ENGL 131
MGH 279

ENGL 131
MGH 279

VIEW SLNS