

# Assignment 4 Proposal

Members: Justin Hu (j62hu) and Xin Lin (x58lin)

Widget id: esf14

Widget name: Exam Scheduler

## Overview

The widget help UW students by organizing every student's specific exam information (subject catalogue, location, time, and date) in a chronological order in one simple interface. This eliminates the need to filter through hundreds of courses to find the few courses that you are taken and the need to constantly remembering or checking the time and date of exams.

Moreover, this widget is also design to help UW students to organize their exam studying schedule. The widget has a built-in to-do list where the user can add items like "Complete chapter 9 by Dec 4". The widget make use of the count down and colour to inform the user about the upcoming exams and their exam to-dos.

## Web Services

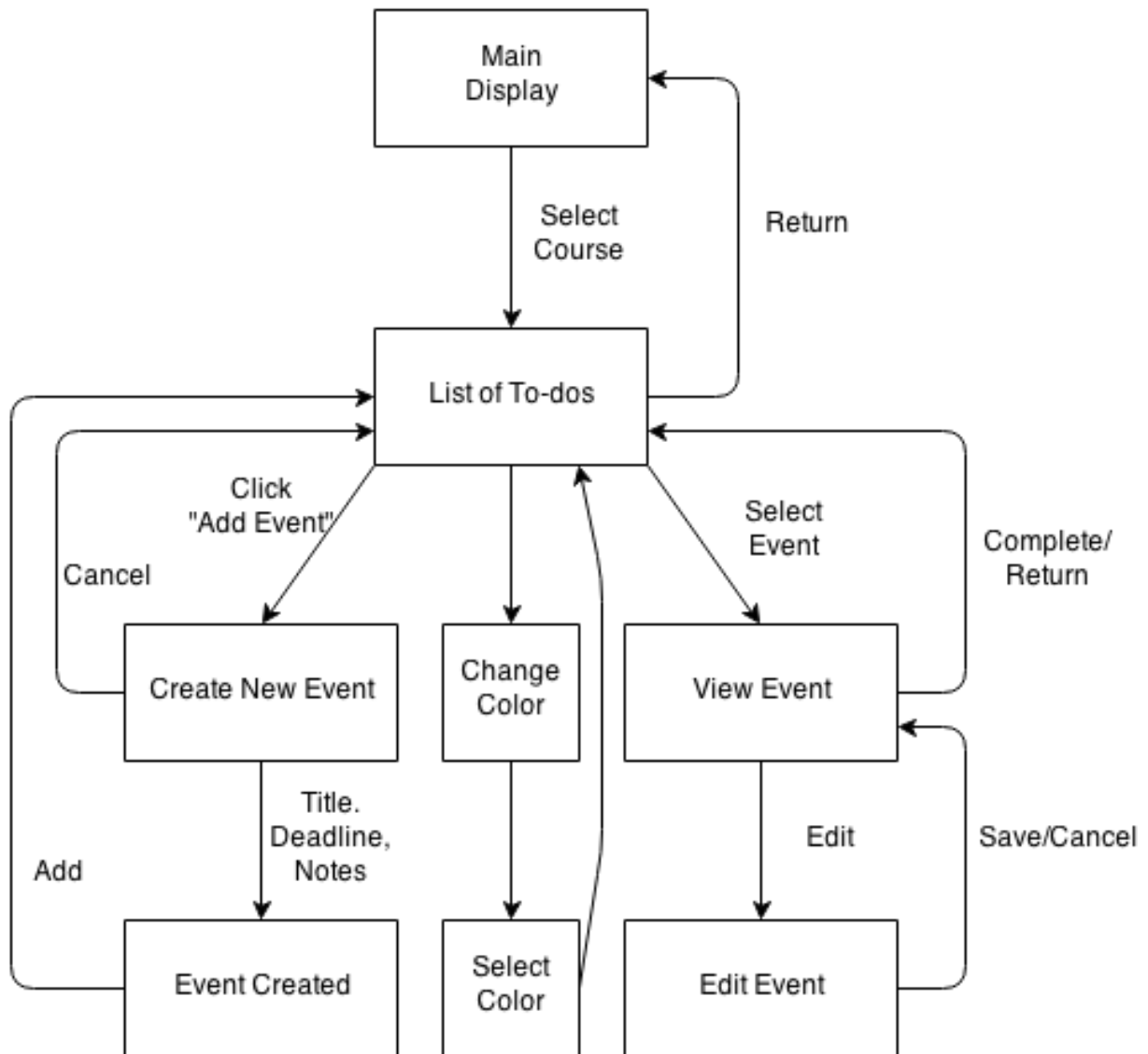
The widget proposed will make use of the following web services.

- Academic data for the user currently logged into the portal.  
The widget needs to obtain the list of courses details of the user logged into the portal. The widget will make use of <https://cs349.student.cs.uwaterloo.ca:9410/api/v1/student/stdCourseDetails/UUUU> where UUUU is the userid of the user logged in.
- Current term  
This widget needs to know the current term in order to filter out the courses taken by the user. The data will be provided by UW Open Data API V2. More specifically, <https://api.uwaterloo.ca/v2/terms/list.json>.
- Current term exam schedule of the courses taken by the user.  
To do this, the widget will make use of the data obtained from UW Open Data API V2. More specifically, [https://api.uwaterloo.ca/v2/courses/{subject}/{catalog\\_number}/examschedule.json](https://api.uwaterloo.ca/v2/courses/{subject}/{catalog_number}/examschedule.json).

## External Libraries

With almost zero experience in JavaScript, we have not realized the usefulness of any other external libraries. If we ever come across the necessity of using external libraries in the future, we will update our proposal to reflect it.

## Interaction Sequence Diagram



## Interface Schematics

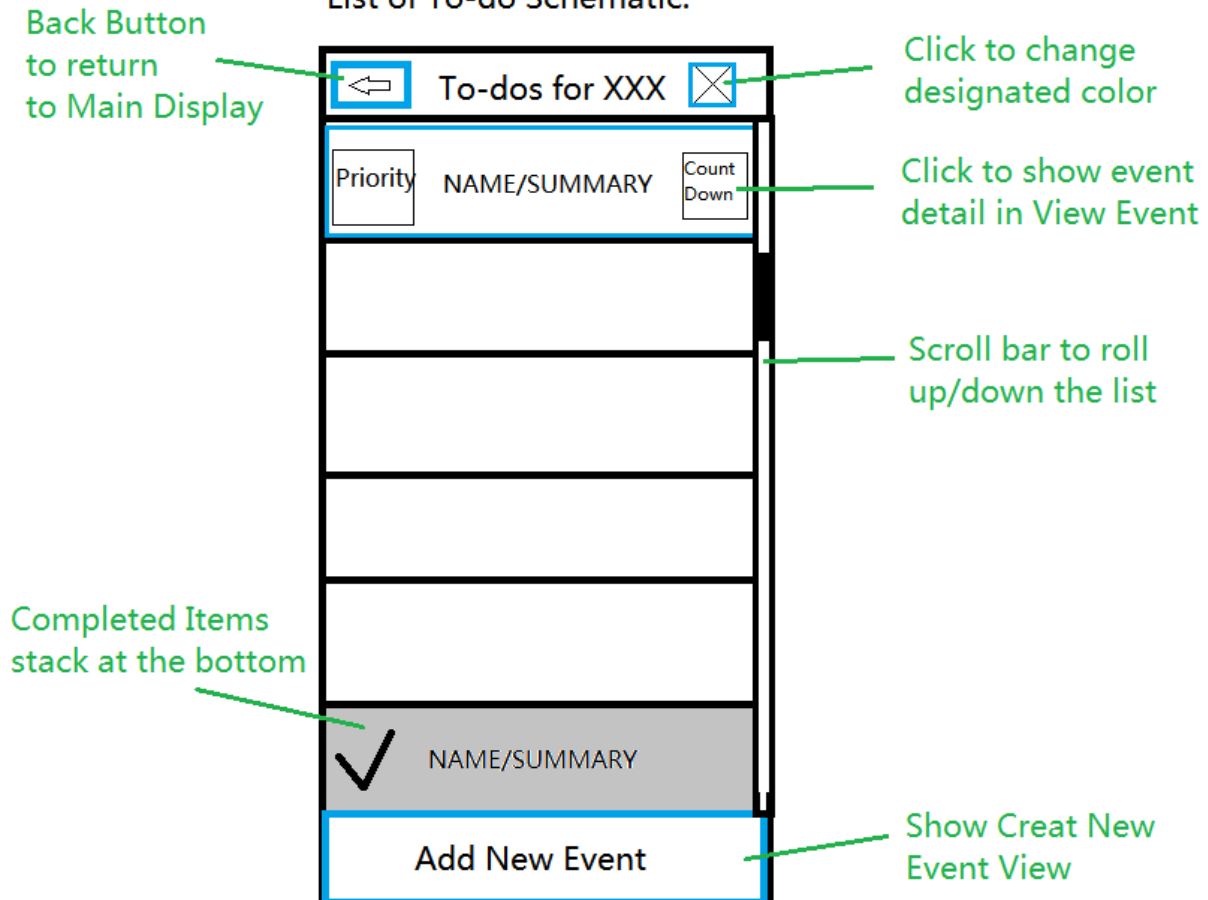
Main Display:

Main Display Schematic:

Exam Schedule			
Subject Catalog		Count	show list of to-dos for this course
Time	Location	Down	
Subject Catalog		Count	
Time	Location	Down	
Subject Catalog		Count	
Time	Location	Down	

## List of To-dos:

### List of To-do Schematic:



## Create New Event:

### Create Event Schematic:

The schematic shows a mobile app interface for creating a new event. It features a title bar with a back arrow and the text 'New Event'. Below the title bar are four input fields: 'Name' (text input), 'Deadline' (two date pickers), 'Priority' (a dropdown menu), and 'Notes' (a text area). At the bottom are two buttons: 'Add' and 'Cancel'. Green lines with labels point to specific elements: 'Back to List of Todos' points to the back arrow; 'Dropdown List for Dates' points to the month and day pickers; 'Dropdown List for Priorities: High/Medium/Low' points to the priority dropdown; and another line points from the 'Back to List of Todos' label to the 'Add' button.

**New Event**

Name: Chapter 5 pg 100 to 120

Deadline: December 03

Priority: High

Notes: This is AWSOME!

Back to List of Todos

Add Cancel


Dropdown List for Dates

Dropdown List for Priorities: High/Medium/Low




























Change Color:

Change Color Schematic:

Back to  
List of Todos



Pick A Color

Set

Cancel

Color Pick that  
user can pick a  
color

View Event:

## View Event

Schematic:

The schematic shows a mobile app screen titled "View Event". At the top left is a back arrow icon. Below the title are three input fields: "Name:" with the text "Chapter 5 pg 100 to 120", "Deadline:" with "December" and "03", and "Priority:" with "High". Below these is a "Notes:" section with a text area containing "This is AWSOME!". At the bottom are two buttons: "Edit" and "Complete".

Annotations with green lines:

- "Back to View Event" points to the back arrow icon.
- "Show Edit Event View" points to the "Edit" button.

View Event	
Name:	Chapter 5 pg 100 to 120
Deadline:	December 03
Priority:	High
Notes:	This is AWSOME!
Edit	Complete

## Edit Event:

Edit Event  
Schematic:

The schematic shows a mobile app interface for editing an event. At the top is a title bar with a back arrow icon and the text 'Edit Event'. Below this is a form with the following fields: 'Name' with a text input containing 'Chapter 5 pg 100 to 120'; 'Deadline' with two dropdown menus, the first showing 'December' and the second showing '03'; 'Priority' with a dropdown menu showing 'High'; and 'Notes' with a text area containing 'This is AWSOME!'. At the bottom are two buttons: 'Save' and 'Cancel'. Green arrows point from text labels to specific UI elements: 'Back to View Event' points to the back arrow icon; 'Dropdown List for Dates' points to the 'December' dropdown; 'Dropdown List for Priorities: High/Medium/Low' points to the 'High' dropdown.

Back to View Event

Dropdown List for Dates

Dropdown List for Priorities: High/Medium/Low

Save Cancel

## Challenges

Both of us have very limited experience using HTML, CSS and JavaScript. Thus, we will face more technical challenges than some of the other students. One of the challenges that we may face would be to interpreting and integrating data from the web services described above. Another technical challenge is to create HTML and CSS that exactly present what we want our interface to look like. However, these challenge can be overcome with getting more familiar with the language by reading sample code, tutorial, and trial and error and with help of Google. Another possible challenge is to have both members committed and complete their tasks on time. At last, there is a challenge of using subversion. We may have been modifying the same file which would result in a conflict. To help avoid potential merging issue of subversion, each member should have their own local copy just in case.