The Design of Everyday Life

By: Team Creative Name
Alexander Maxwell (100814643)
Matthew Penny (100937331)
Seena Rowhani (100945353)

COMP 3008

April 9, 2016

Section 1 – Requirement Gathering

1.1 – Overview

Planning ten months in advance is never an easy task. Yet every year come early summer students sit down to plan their lives for the upcoming eight months. Along with planning around the unpredictability of life, students need to consider degree requirement, course requirements and things alike. Taking all of this information into consideration can overwhelm students when trying to select courses for the upcoming year. Overloading the users with information not only causes them to make mistakes when selecting courses, but also leads to less desirable schedules. A student's timetable sets the pace of their semester. A poorly laid out out timetable can cause undue stress and can negatively impact performance in classes.

Our group has decided to design a University timetable planner, with the goal of aiding students in the course selection process. The application will address two key issues with timetable creation under two different categories.

The first category looks at degree requirements set out by the University. We will assist users in course selection by providing only valid course options for their degree to help prevent common mistakes. Along with automation we will look to provide users with only essential information when making course selection decisions.

The second issue we will address the creation of more efficient timetables. We will offer assistance by suggesting time slots or electives in order to free up time based on users' needs. Doing so will give students more time to complete work, partake in extracurricular activates or take much-need breaks.

By addressing these two key issues, our application will make student more productive during their semesters. We will alleviate the overload of information by providing the course information as needed, which will allow students to create timetables that suit their needs.

1.2 – User Interviews

The purpose of the interviews was to gather information from potential users of the course selection application. The goal was to gather information on how students currently create university timetables, and spots of bother with their current process. We sought out students who were presently in university as our interview subjects. We looked for diversity in our subjects, targeting students with different majors and lifestyles. Each of the interviewees was interviewed using the following questions.

Interview Questions

What platform are you most likely to use (e.g., desktop, mobile, web)?

How long do you spend creating your schedule?

How do you decide which courses to select?

What type of information do you look for when enrolling in a class?

What is your biggest problem with picking classes?

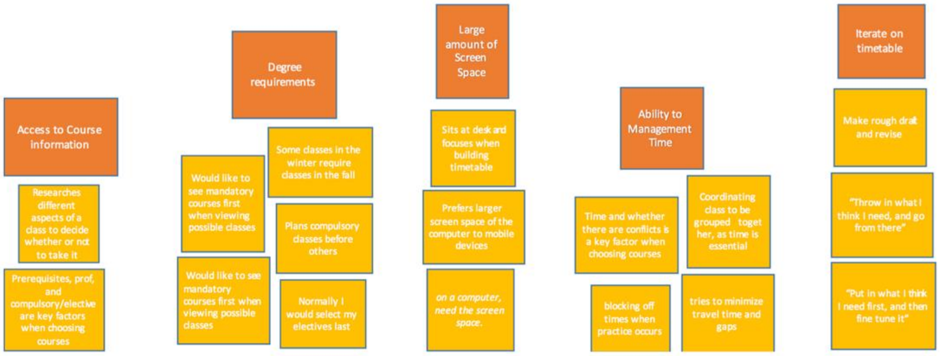
Would you like to see all classes, or just what's available for you?

Is there an order in which you select your classes?

How do you balance your timetable?

For interviewee descriptions and their responses, see appendix.

1.3 – Affinity Diagram and Requirements


Access to Course information
Researches different aspects of a class to decide whether or not to take it
Prerequisites, prof, and compulsory/elective are key factors when choosing courses
Degree requirements
Would like to see mandatory courses first when viewing possible classes
Would like to see mandatory courses first when viewing possible classes
Some classes in the winter require classes in the fall
Plans compulsory classes before others
Normally I would select my electives last
Large amount of Screen Space
Sits at desk and focuses when building timetable
Prefers larger screen space of the computer to mobile devices
on a computer, need the screen space.
Ability to Management Time
Time and whether there are conflicts is a key factor when choosing courses
Coordinating class to be grouped toget her, as time is essential
blocking off times when practice occurs
tries to minimize travel time and gaps
Iterate on timetable
Make rough draft and revise
"Throw in what I think I need, and go from there"
"Put in what I think I need first, and then fine tune it"

We used the insight gained from the interview process to set design requirements. The feedback from the interviewees was analyzed using grounded theory. The open coding process was done individually by each member of the team, where codes were assigned to the interview transcripts. The codes marked comments made by interviewees around the frustrations of their current course selection process and desired information to facilitate course selection. The codes were then grouped into similar issues as part of the axial coding process. Relationships were found between the groups and were categorized into feature requirements to address the course selection issues.

It became apparent that students' main goal when selecting courses is to create a balanced timetable (i.e., lack of large time gaps and manageable course difficulty). The application's requirements can be categorized under three categories. The first category is degree requirements which will contain items involving degree restrictions, mandatory courses, and course already taken. The second category is course selection which will classify items that involve viewing availability and course information. The third category is time management, where requirements address issues pertaining to building timetables where courses are conveniently placed in the user's week.

The primary objective for students in taking courses is getting one step closer to graduation. In order to graduate, students must complete the required courses and number of course credits to meet their degree requirements.

The first requirement of the application is the ability to determine the courses which need to be taken. Many students have an idea of what courses need to be taken, but would benefit from having the information easy available. This requirement should allow users to stay in the application to access this information. Doing so would improve the student's current selection process by alleviating the need to switch between applications to access the information.

The second requirement pertaining to degree requirements would be the ability to determine courses already taken. This requirement would prevent students from having mental slips and adding a course which has already been taken. This requirement should not prevent users from retaking a course completely, but ensure they are doing so knowingly.

Course selection requirements pertain to selecting from the acceptable courses. Degree requirements require students to enroll in mandatory, elective, and optional mandatory courses (i.e., any second year math course). While selecting mandatory courses does not require much thought, being aware of descriptions and prerequisites would be beneficial when deciding between optional mandatory classes and electives.

The first requirement relevant to course selection is the ability to navigate through all available courses and also filter them. The purpose of this requirement is to allow users to find electives and courses of interest they may not otherwise have been aware of.

The second requirement for course selection is to provide easy access to course information. The information serves to allow users to learn more about courses and easily confirm they are registering in the correct one. In addition to the title, there should be a course description which gives an overview of what the course covers.

Third, course selection should allow the ability to determine the prerequisites of a course. The ability to determine if a student has met the prerequisites would assist in verifying eligibility for the course as well as assist in selecting courses to meet future course prerequisites.

The final consideration students take into account when selecting courses is time management. Time management involves creating timetables that complement the student's lifestyle. We found that interviewees had differing opinions on what an ideal daily schedule was. Feedback from user interviews indicated students make iterations of schedules in order to achieve the timetable that best suit their needs. Interviewees also indicated they select some electives solely for the purpose of creating a convenient timetable.

The first requirement for creating a convenient schedule is the ability to easily determine openings in the student's current timetable. The ability to readily access a visual representation of free timeslots during the week is a top priority. The ability to easily add and remove courses to and from these timeslots (also through a visual representation of the schedule) is also of significant of importance. Real-time visual feedback allows changes to easily be conveyed to the user.

The second requirement is the ability to easily determine what time(s) courses are available at. Interviewees mentioned building time-efficient timetables was a key determinant when selecting courses. Information about the various times a course is available should be easy to access and act upon.

The third requirement is the ability to easy develop fall and winter schedules at the same time. Most interviewees stated they build their fall and winter timetables concurrently. The purpose of this requirement is to enable users to create time effective timetables by allowing the optimization of course selection over two semesters. There should be a clear indication of the semester the user is currently working on.

The fourth requirement is the ability to determine which courses are available at given times. The purpose of this requirement is to allow users to easily select from courses at times that suit their current timetable. Users should be able to find courses available at given times and days of the week. This can be done through some form of filtering to select the timeslot(s) of interest.

The fifth requirement is the ability to easily add hypothetical courses to the calendar. The purpose of this requirement is to remove the need for creating multiple iterations of the calendar. Being able to visually see a hypothetical course on the calendar will allow users to reason better about which courses work best. Hypothetical courses should be easily removed and set as selected courses. There should be a visual distinction between added and hypothetical courses.

The implementation of features that meet the requirements will provide users with the essential tools they need to create the most efficient timetable. Being able to create timetables to suit their needs will allow students to be more productive and attend lectures more regularly. Requirements should be met in the fewest number of features as possible. The minimalist approach to creating features should not impact the key functionality of the requirement.

Section 2 – Initial Design Prototypes

Prototype 1: Calendar-Focused Design

The first prototype proposed was the **Calendar-Focused Design**. The design employs a direct manipulation interface with the week view at the center of the design. Tools for adding and removing courses are found along the right side of the week view. A direct manipulation interface was chosen to provide users with instant visual feedback in order to quickly understand the effect of changes made the timetable. The primary focus of the design is the creation of time efficient timetables. The design facilitates the requirements in the following ways.

The first feature is the direct manipulation interface. The design encourages experimentation and rapid iterations of the timetable. Rather than creating many draft versions of timetables, the design caters to one timetable which is frequently improved upon. This is achieved by allowing users to browse courses with the weekly view being visible. Being able to reference the week view while searching allows users to easily identify openings in the schedule. When hovering over or selecting a timeslot for a course, a colored box is temporarily placed onto the timetable to indicate where the course will fit into the schedule. Multiple courses can be selected in this way and then added when the user is satisfied. This allows users to see exactly how a course will affect their timetable before adding it. Being able to temporarily add courses to the timetable satisfies the requirement of adding hypothetical courses. In addition, users can find courses to fill gaps in their timetable by selecting the timeslot on the week view. When a timeslot is selected, a dialog will present users with all courses offering during the selected time and day. This allows users to easy determine courses available at a particular time and optimize their day.

The design uses a cascaded style to organize courses in the course list. It is designed to only show users information which is contextually relevant to their place in the application. In addition to the course list, advanced search functionality is provided to allow the user to filter the course list. Filtering will allow users refine search results based on time of day and required courses for their degree program. The filtering functionality addresses many requirements including the ability to see courses available at given times, the ability to view courses required for the student's degree, and exclude courses the student has already completed. Filtering enables users to narrow down the course list to find mandatory courses and add them to the schedule first, before potentially finding that they conflict with electives.

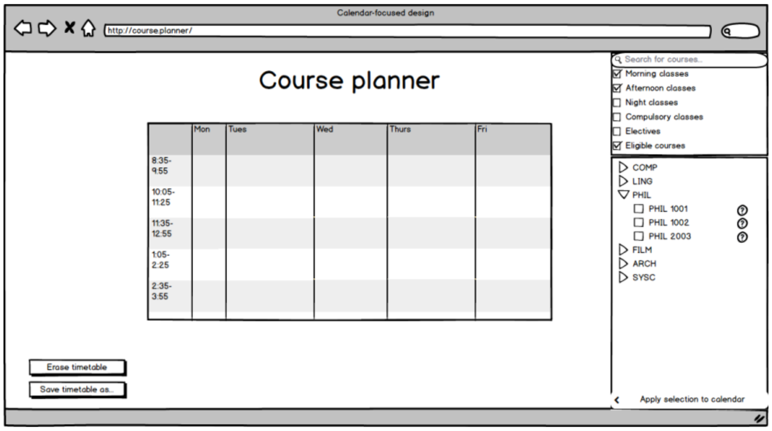
To address the requirement of easy access to additional course information, the course description and course name are provided to the user through a modal dialog box which is displayed when the course is clicked on. This allows users to quickly navigate through the minimalist course list but still access detailed information when it is desired.

Prototype 1 Storyboards

Adding a course

Suppose a user wants to add a course to their timetable. They open the course planner application to do so.

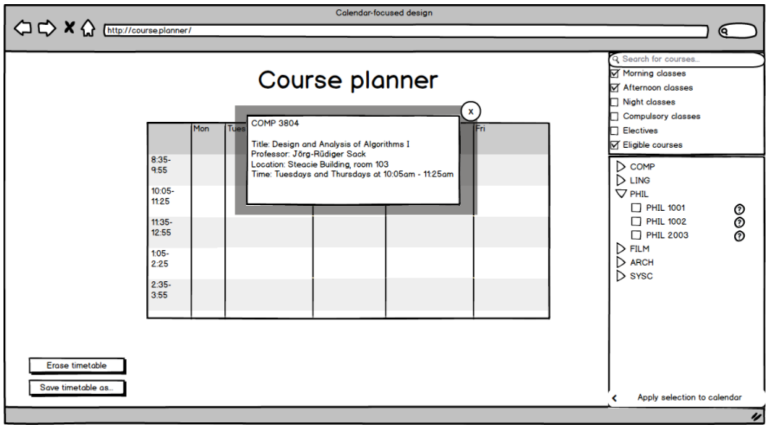
1) This is the main screen. The user searches for their desired course using the search field of the sidebar on the right. They can also use the filters below it to further refine their search.



☑ Morning classes
☑ Afternoon classes
☐ Night classes
☐ Compulsory classes
☐ Electives
☑ Eligible courses

☐ PHIL 1001
☐ PHIL 1002
☐ PHIL 2003

2) The user clicks the info button beside the course name to view more about it and verify that it is, in fact, a course they want to add to their calendar.



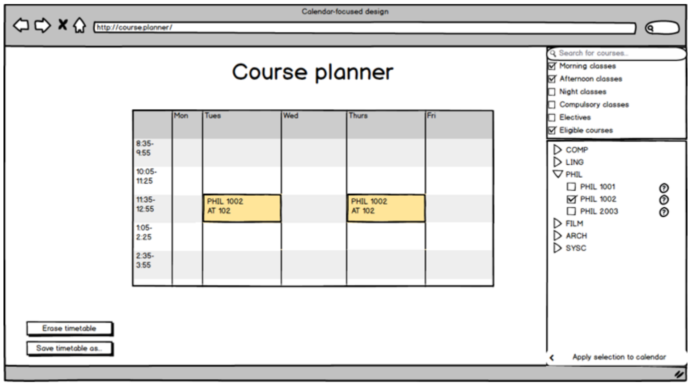
COMP 3804

Title: Design and Analysis of Algorithms I
Professor: Jörg-Rüdiger Sack
Location: Steacie Building, room 103
Time: Tuesdays and Thursdays at 10:05am - 11:25am

☑ Morning classes
☑ Afternoon classes
☐ Night classes
☐ Compulsory classes
☐ Electives
☑ Eligible courses

☐ PHIL 1001
☐ PHIL 1002
☐ PHIL 2003

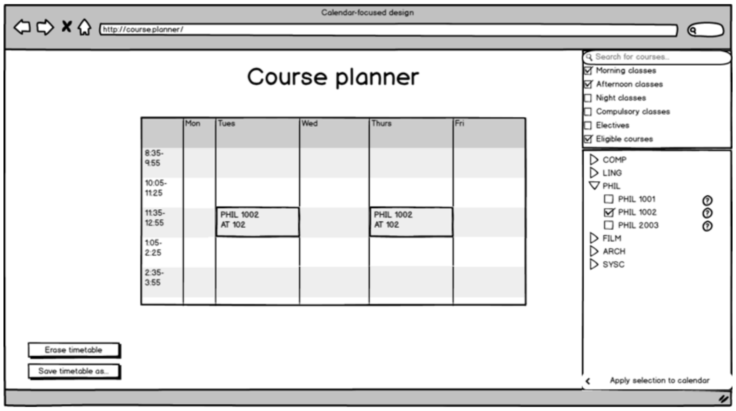
3) The user selects the checkbox for the course, which temporarily places (queues) it on the calendar, allowing them to see how it will fit in with the rest of their courses. The user can uncheck the checkbox to undo this action and remove the temporary addition. Multiple courses can be queued in this way at the same time to allow the user to see how they will affect the weekly schedule.



☑ Morning classes
☑ Afternoon classes
☐ Night classes
☐ Compulsory classes
☐ Electives
☑ Eligible courses

☐ PHIL 1001
☑ PHIL 1002
☐ PHIL 2003

4) The user clicks the “apply selection to calendar” button to apply their queued changes. Once added, the course changes color to match the background of the rest of the calendar (indicating that it has been added and that its presence is no longer temporary).



☑ Morning classes
☑ Afternoon classes
☐ Night classes
☐ Compulsory classes
☐ Electives
☑ Eligible courses

☐ PHIL 1001
☑ PHIL 1002
☐ PHIL 2003

Prototype 2: Multiview Material Design

The second prototype proposed was the **Multiview Material Design.** This design is instructions design which involves stepping users through pages to interact with the application. Navigating through the pages is done using the navigation bar to the left of the screen. Separating the application into pages allows the users to sequentially step through the searching, adding, and removing processes. Feedback for actions is provided by the view changing from one page to another. The design fulfills the requirements in the following ways.

The prototype excels in providing users with the ability to browse through the course list. The courses are displayed in as small cards with course title, number and a small description of the course. The image featured on the card would be a photograph of the professor teaching the course, making the experience more personable. Providing easy access to course information allows users to quickly exclude courses they are not interested in. A search bar allows for advanced searching of the course list. By specifying search criteria such as course title or number, the user can narrow down the course list. Search bar results can be further refined using the filter button, which provides the abilities to view courses available at a certain time of day, show only required courses, and exclude courses already taken. This advanced filtering functionality satisfies the requirement of viewing required degree courses while excluding courses already taken.

When a course is selected from the course list, a dialog appears providing the user with additional course information including its description, prerequisites, and available timeslots. The time and day of the course is provided through a visual representation of where the course will appear on the week view. Different timeslots of the course have their own individual course cards. Switching between different course timeslots must be done through the course list.

To add courses the user must navigate to the information dialog box and click the plus button in the bottom right corner. Once selected, the information dialog will disappear and a check mark will appear on the course on the course list view. To confirm the adding of a course, the user must click the large plus button at the bottom of the course list. If the user wants to see how the course will appear on their current timetable without adding it, they must select the calendar button next to the plus button from the course information card. This will open the week view with the course added to allow the users to see if the hypothetical course fits well in the timetable. Removal of selected courses is done by pressing the minus button on the course title from the week view.

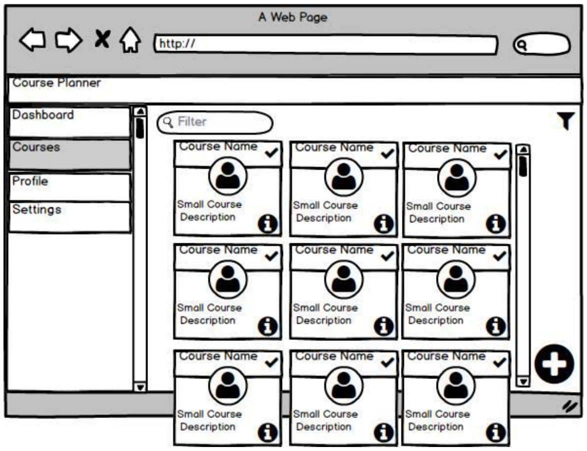
Stepping the user through the adding process enables learnability by solidifying the process in their mind and prevents mistakes from being made in the future. Users can add the most important courses first using the mandatory course filtering options then change the filter to time of day when selecting electives. Being able to see if a course will fit into the schedule before adding it eliminates the need to add, and then remove it. This results in a quicker timetable creation process.

Prototype 2 Storyboards

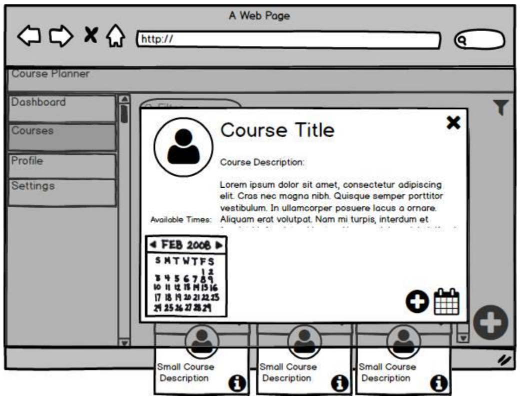
Adding a course

Suppose a user wants to add a course to their timetable. They open the course planner application to do so.

1) Start on the courses page. Available courses are listed, which the user can search through using the search box. When the filter icon is clicked, the user is presented with advanced options for searching. Basic, high-level course information is displayed for each result

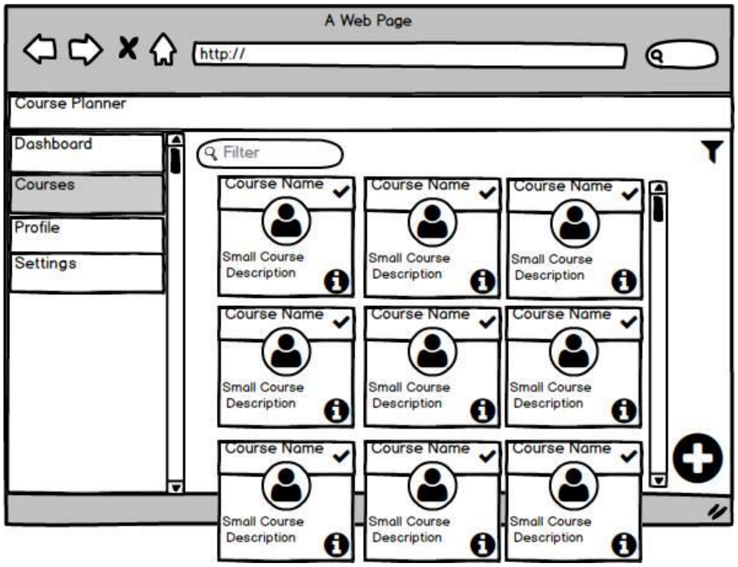


2) The user clicks on the info button for a course they are interested in. This displays a detail view for the selected course. The timeslot in which the course will go on the schedule is displayed in the bottom left.

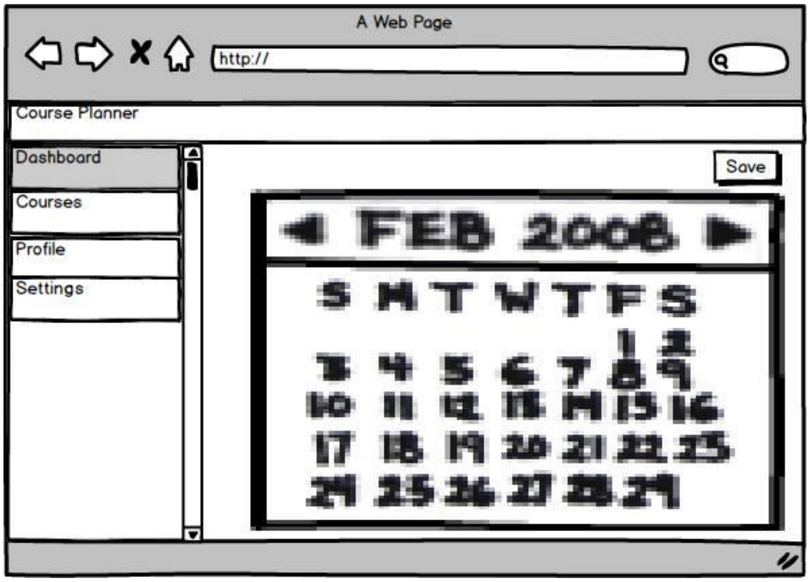


Lorem ipsum dolor sit amet, consectetur adipiscing elit. Cras nec magna nibh. Quisque semper porttitor vestibulum. In ullamcorper posuere lacus a ornare. Aliquam erat volutpat. Nam mi turpis, interdum et

3) From the course info view, the user clicks the plus button to indicate they wish to add the course to their timetable. Upon doing this, a checkmark will appear beside the course in the course list. The user then confirms any changes using the large plus (confirm) button at the bottom right of the page. Multiple courses can be added at once in this way.



4) Viewing the dashboard page displays all courses that have been added to the user’s timetable.



Section 3 – Prototype Evaluation and Iteration

3.1 – Cognitive Walkthroughs

We conducted cognitive walkthroughs of the prototypes to determine the usability and learnability of essential features to complete tasks. Evaluators stepped through each task to determine if it was evident what to do next, how to do it, and whether there was sufficient feedback for actions. Each prototype was stepped through the following six tasks. The results are listed below.

1. Search for course
2. Filter out courses
3. Get course information
4. Adding a course
5. Remove a course
6. Saving timetable

Prototype One (Calendar-Focused Design)

1. Searching for course
    a. Will the correct action be sufficiently evident to the user?
        Yes, courses are visible and it makes sense to be able to search them.
    b. Will users be able to notice that the correct action is available?
        Yes, the search bar is labelled.
    c. Will the user interpret the resulting behavior/feedback correctly?
        Yes, user is presented with courses based on their search.

2. Filtering Courses
    a. Will the correct action be sufficiently evident to the user?
        Yes, it is already evident that courses are searchable.
    b. Will users be able to notice that the correct action is available?
        Yes, the filter section is labelled.
    c. Will the user interpret the resulting behavior/feedback correctly?
        Yes, the results will be filtered based on their actions.

3. Getting course information
    a. Will the correct action be sufficiently evident to the user?
        Yes, it seems evident to the user that this action is possible.
    b. Will users be able to notice that the correct action is available?
        Yes, user has two options that seem quite apparent, clicking info on the tree view / clicking on the actual course itself.
    c. Will the user interpret the resulting behavior/feedback correctly?
        Yes, course information will be displayed immediately.
4. Adding Courses
    a. Will the correct action be sufficiently evident to the user?
        Yes, the next logical step after finding a course would be to add it.
    b. Will users be able to notice that the correct action is available?

Not necessarily, apply could be more descriptive. As well, the course button could be more visible.

   c. Will the user interpret the resulting behavior/feedback correctly?

      Not necessarily, feedback could be clearer.

5. Removing Courses
   a. Will the correct action be sufficiently evident to the user?

      Yes, it seems understandable that a course should be removable.
   b. Will users be able to notice that the correct action is available?

      Yes, the subtract button on the course seems to meet cultural standards for removal. As well there is the ability to unselect a course form the list. It follows the same behavior as the add process, where the apply button must be selected again.
   c. Will the user interpret the resulting behavior/feedback correctly?

      No, the change in state may be confusing to the user. If a button is unselected accidentally and the apply button is clicked, a course may appear to have been removed without reason. Perhaps a dialog to confirm the removal behavior should be added.

6. Saving timetable
   a. Will the correct action be sufficiently evident to the user?

      Yes, there should be a capability to save a timetable.
   b. Will users be able to notice that the correct action is available?

      Yes, the save button is clear enough, but not in an obvious location. It should be moved to the top of the screen to conform to norms.
   c. Will the user interpret the resulting behavior/feedback correctly?

      Yes, the save button only being selectable when a change has been made means there would be a visible change in the button when the timetable has been saved.

Prototype Two (Multipage Material design)

1. Searching for course
   a. Will the correct action be sufficiently evident to the user?

      Yes, it is expected that there should be a way to look through the courses.
   b. Will users be able to notice that the correct action is available?

      Yes, selecting the course heading is intuitive.
   c. Will the user interpret the resulting behavior/feedback correctly?

      Yes, user is immediately presented with the course page.

2. Filtering Courses
   a. Will the correct action be sufficiently evident to the user?

      Yes, it seems possible there would be a functionality to narrow down the search results.
   b. Will users be able to notice that the correct action is available?

      No, the filter button appears separate from the search field and is unlabeled.
   c. Will the user interpret the resulting behavior/feedback correctly?

Yes, the number of results will change. This is signified by the scrollbar changing in size. Although, if none of the visible results have been filtered out, it may be confusing.

3. Getting course information
    a. Will the correct action be sufficiently evident to the user?
        Yes, it seems possible there would be a course description available.
    b. Will users be able to notice that the correct action is available?
        Yes, the “i” button is standard.
    c. Will the user interpret the resulting behavior/feedback correctly?
        Yes, the course list will be replaced with the course information view.

4. Adding Courses
    a. Will the correct action be sufficiently evident to the user?
        Yes, the next logical step after finding a course of interest would be to add it.
    b. Will users be able to notice that the correct action is available?
        No, having to go to another view to add the course is not necessarily apparent.
    c. Will the user interpret the resulting behavior/feedback correctly?
        Yes, a checkmark will appear on the course in the course list and the course will appear in the week view.

5. Removing Courses
    a. Will the correct action be sufficiently evident to the user?
        Yes, it seems understandable that a course should be removable.
    b. Will users be able to notice that the correct action is available?
        Yes, selecting the remove button is very large.
    c. Will the user interpret the resulting behavior/feedback correctly?
        Yes, the checkmark will disappear from the course list and the course will be removed. Perhaps a dialog to confirm the removal behavior should be added.

6. Saving timetable
    a. Will the correct action be sufficiently evident to the user?
        Yes, there should be a capability to save a timetable.
    b. Will users be able to notice that the correct action is available?
        Yes, the save button is clear enough, but not in an obvious location. Perhaps moving it to the outside of the calendar would make its presence more obvious.
    c. Will the user interpret the resulting behavior/feedback correctly?
        Yes, the save button only being selectable when a change has been made means there would be a visible change in the button when the timetable has been saved.

3.2 – Prototype Selection

The team decided to proceed with Calendar-Focused Design prototype. Results from the cognitive walkthrough and how well the design met the application requirements were considered when making the decision. The results from the cognitive walkthrough were similar for both designs.

The designs shared a common workflow for completing tasks (i.e., first queuing and then applying changes).

However, considering in detail how users would perform tasks revealed design flaws in the Multipage Material Design. In some instances, there were issues with the association of common elements on the page. An example of this is was the distance between the search field and the advanced filtering button. Horizontal distance between the two elements violated the principle of proximity, resulting in elements being perceived as unrelated. This could lead to confusion regarding the functionality of the filter button, resulting in the perceived absence of functionality. Another instance of poor design was found in the process of adding a course to the timetable. From the course selection screen it was not clear how a course should be added. In order to add a course, users need to open the course information window. This becomes an issue for the use-case of adding mandatory courses where users may not be interested in the course description and is looking to add the course from the course list view.

The Calendar-Focused Design experienced issues when considering how the interface provided feedback for actions. Problems occurred during the addition and removal of courses. After clicking the apply button to add courses, the only confirmation of a course being added is its appearance on the timetable. Confirming the course was added would involve the user looking for the course on the week view, which would be tedious if the number of courses on the timetable was large. Another problem occurred when looking at the course removal process. To remove a course, the user would unselect it from the course list and press the apply button. This becomes problematic if a course is accidentally unselected and the user adds another arbitrary course. Since adding and removing is done through the same "Apply" button, the user would be unknowingly removing the course. The user should be presented with some form of additional confirmation action when removing a course.

We also looked at how well the designs met the identified design requirements. Although all requirements were considered, the deciding requirement between the two designs was the ability to easily determine openings in the user's schedule. The Calendar-Focused Design meets this requirement by having the week view constantly visible, especially while searching through the course, unlike the Multipage Material Design. Not being able to see the calendar requires the user to remember a potentially large number of time openings. The cognitive strain of remembering all possible openings could cause users to making mistakes as a result of forgetting the available times.

We believe the Calendar Focused Design is better suited for a wide range of users with vastly differing levels of experience. Additionally, having the week view visible at all times better aligns with the needs articulated by users during the interviews. Based on the cognitive walkthroughs and how well the prototype meets the design requirements, we believe the Calendar Focused Design is a better starting point for iterations.

3.3 – Iterations

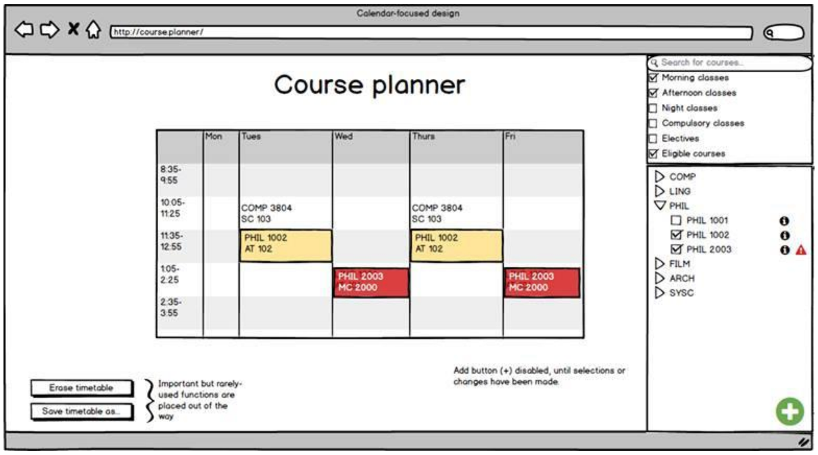
The base prototype of our iterations was the Calendar-Focused Design. We gathered insight from participants using the Wizard of Oz technique. Before each iteration, we simulated interaction with the interface with two individuals. We gathered the participants' initial impressions of the interface then stepped them through a sequence of tasks, collecting feedback and observing their behavior along the way. Feedback from participants was consolidated and analyzed to determine the root cause of any issues and possible improvements to be made to the prototype.

Iteration one

To start the testing, participants were presented with the prototype setup with a few courses already added. The initial reaction of the first user was to interact with the items on the calendar. They contemplated the meaning of the colors of the tiles. The participant's impression was correct in interpreting red courses as having errors, but incorrect in interpreting white courses as mandatory yellow courses as problematic in some way. White and yellow did not effectively symbolize the appropriate state of the course and caused confusion. The second participant's initial reaction was to look to the calendar view, but took the time to familiarize themselves with the different components of the application. Both participants perceived the calendar to be the main focal point of the application and correctly assumed the week view was the current state of the timetable being created.

For the first task, users were instructed to view the course information for COMP 3804 which was located on the week view. Both participants reacted by clicking on the course tile on the week view. After selecting the course tile, an information dialog with the course information was displayed. Participant one believed the information dialog should include functionality to change the course timeslot to a different time or remove the course. Participant two was content with the information provided from the course information dialog, but didn't believe the course on the week view looked clickable.

After returning to the week view, the participants were instructed to find information about a specific course and add the course to the timetable. Both participants focused on the course list to the right of the calendar. Participant one's attention was drawn specifically to the search bar and expressed interest in entering a topic into the search field. After explaining the behavior, the participant clicked on the question mark button next to the course to view information. At this point the participant wanted to add the course from the information dialog (which was not possible). Going back to the week view, the participant selected the checkbox next to the timeslot in the course list, but took some time to find the apply button to confirm their changes. Participant one's incorrectly perceived the course information dialog as the main interaction point for adding, removing and picking a timeslot of the course. The participant believed the course list was cluttered with filtering options being displayed at the top of the course list. Participant two used the check box to select the course but did not comprehend the dotted course outline on the calendar. After explaining the dotted line was where the course will go, the participant was confused. Participant two's thought that the course would be automatically added to the calendar when the course timeslot was checked. For this reason, participant two didn't try to find the apply button.

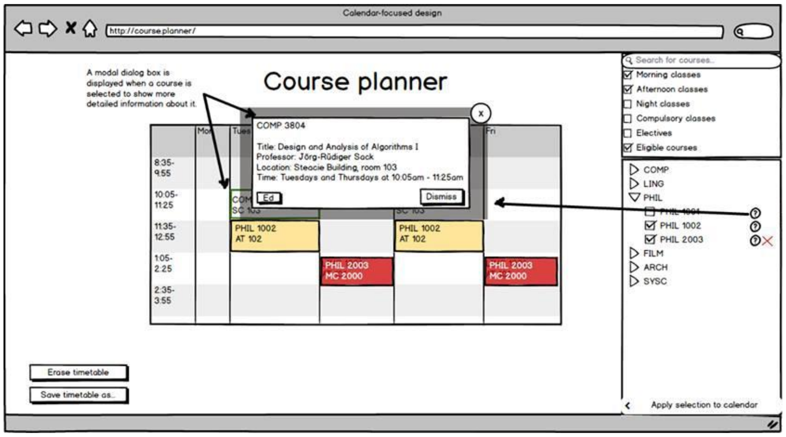


	Mon	Tues	Wed	Thurs	Fri
8:35- 9:55					
10:05- 11:25		COMP 3804 SC 103		COMP 3804 SC 103	
11:35- 12:55		PHIL 1002 AT 102		PHIL 1002 AT 102	
1:05- 2:25			PHIL 2003 MC 2000		PHIL 2003 MC 2000
2:35- 3:55					

Important but rarely-used functions are placed out of the way

Add button (+) disabled, until selections or changes have been made

After considering the results from the Wizard of Oz simulations, we made a number of revisions to the application. The first change addressed the confusion around the process of adding courses. Participant one had problems locating the apply button and Participant two did not believe an apply button was part of the adding process. We believe participant two was unable to realize the apply button was part of the process because they were not aware of it on the interface. This is understandable as the apply button was out of the way, located in the bottom right corner. To address these issues, we made the apply button larger and added color to draw attention to it. These visual changes will increase the button's presence, helping users easily locate it. Being aware of the apply button will have the added effect of clarifying the meaning of dotted course outline on the calendar.



A modal dialog box is displayed when a course is selected to show more detailed information about it.

COMP 3804

Title: Design and Analysis of Algorithms I
Professor: Jörg-Rüdiger Sack
Location: Steacie Building, room 103
Time: Tuesdays and Thursdays at 10:05am - 11:25am

Following this, we added the ability to manipulate the course from the course information screen. This was done to address participant one's feedback of wanting the ability to add and remove courses from their information dialogs. We also addressed the user's desire of removing and editing the course from the week view. The ability to remove a course is now available in top right corner of course tiles. We decided not to add the functionality of adding a course from the information dialog, as we believe it would add confusion around when clicking the apply button was required to add the course.

To address concerns regarding clutter we added more space between search results to help distinguish between them. We also changed icon on the course information from a help icon to an information icon, to better align with interface standards.

To help users distinguish boundaries between components, we added borders around courses on the week view. Borders on courses in the week view will be a signifier that the course tile can be clicked and will assist in determining what time the course begins and ends.

These changes will aim to provide a better user experience by providing more interaction points for editing or removing courses. As well, making the apply button stand out clarifies the process for adding courses. This will allow users to take advantage of the hypothetical course feature by making dotted course outline on the calendar clear.

Iteration two

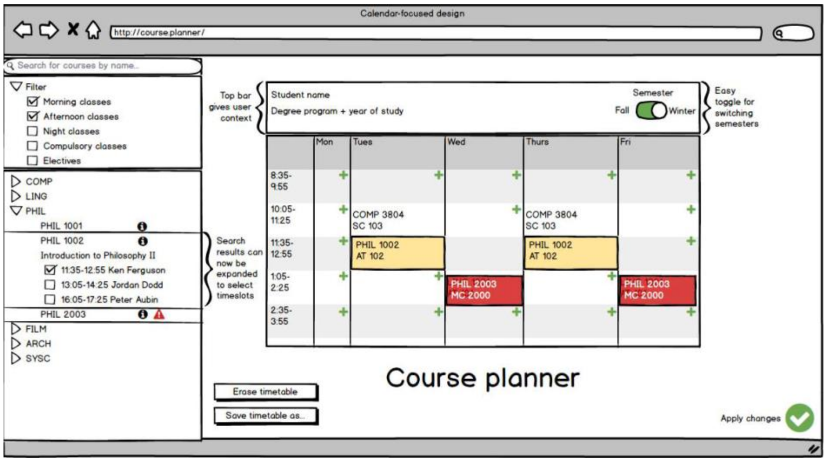
To start, the users were presented with a week view with a few courses already added to the timetable. The attention of both participants started on the week view and moved from left to right across the screen. The initial impression was all interactions were to occur through the week view. Participant two believed the application was missing context of who the calendar was for, and was concerned with how to change between semesters.

The participants were then asked to search for a course. Both participants indicated they would use the search bar to narrow down the results. Participant one was unsure of where the results would appear as the course list was already filled with courses. The problem seemed to be the distinction between filtering out courses that didn’t match the search, versus finding courses that did. Participant two understood where the results would appear, but was unsure of how to format their search query.

After searching, we asked the participants to find information about a course. Participant one’s reaction was to click on the course tiles on the week view to access course information. The participant took longer to discover the information button on the courses in the course list. Participant two found the information icon on the course list items, but expressed a dislike for the application behavior. The participant was not satisfied with the information icon as it seemed to be a secondary action. The participant also did not care for the full screen information dialog, as they felt it interrupted the searching process.

Next, the participants were tasked with adding a course to the timetable. Both participants followed the same process of adding a course using the checkboxes and pressing apply. Participant one immediately understood the dotted line course tile on the week view while Participant two needed to experiment with the checkboxes to understand the behavior. Through experimentation, Participant two discovered the ability to add multiple courses to the calendar by selecting numerous checkboxes. Participant one stated that the hypothetical courses should use color instead of dotted outlines to indicate more actions are required.

The final task was to remove a course from the timetable. Participant one removed the courses by selecting the courses to be removed from the week view. Based on their earlier discovery, participant two used the course list to remove multiple courses. Participant two indicated the ability to substitute courses in the same apply would be convenient.



Top bar gives user context

Easy toggle for switching semesters

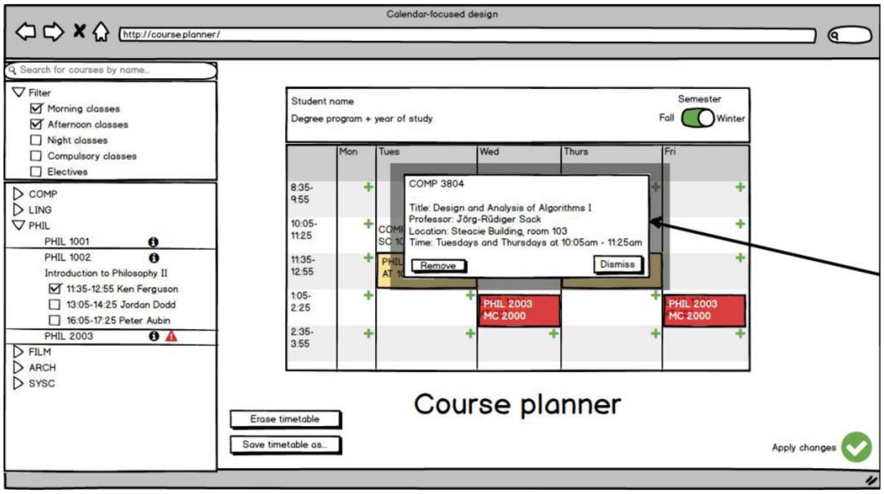
Search results can now be expanded to select timeslots

	Mon	Tues	Wed	Thurs	Fri
8:35-9:55					
10:05-11:25		COMP 3804 SC 103		COMP 3804 SC 103	
11:35-12:55		PHIL 1002 AT 102		PHIL 1002 AT 102	
1:05-2:25			PHIL 2003 MC 2000		PHIL 2003 MC 2000
2:35-3:55					

After considering the results from the Wizard of Oz simulations, we made the following changes. The first change addresses the confusion regarding the workflow of the application. The fact that the participants' attention was drawn to the calendar was concerning because most of the interaction is done through the course list. With the simulated state, the attention on the week view was not an issue due to courses already being added to the calendar. The problem occurs when there are no existing courses on the timetable and users incorrectly assume that interactions start with the calendar. To change the users' workflow, we moved the course list to the left side of the screen. We believe this change will increase the usability of the application as the desired workflow of the user will align to natural the workflow of users from cultures who read from left-to-right.

To help guide users with the searching functionality, we added sample queries in the search field (as placeholder text) when it is empty. The samples will assist users in formatting queries. The ability to properly format queries will allow users to get the desired results out of their search. To remove the perceived disconnect between the advanced filtering option and the search bar, we made the advance filtering options available through a drop down menu below the search field. The small distance between the two screen elements uses the proximity principle to show a relation between the two functions.

Another change we made was updating the search result list in real time. This was to address participant one's confusion around how filtered search results are shown. As a query is entered or filter checkbox is checked, the course list will automatically start filtering to match. This provides immediate feedback which helps users them learn the results of their actions and how the application works.



COMP 3804

Title: Design and Analysis of Algorithms I
Professor: Jörg-Rüdiger Sack
Location: Steacie Building, room 103
Time: Tuesdays and Thursdays at 10:05am - 11:25am

To simplify the course list, the different timeslots for a selected course now organized in a cascading style. The change was made to address the interruption the course information dialog caused when trying to quickly add courses to the timetable. The timeslots remain hidden until needed and are presented to the user when they click on a given course.

Finally, we changed the process of adding and removing courses to a queuing process. Changes to the timetable are now made by first adding them to the queue. They are then applied using the apply button. Doing so addresses the potential issue of courses being accidentally removed by clicking the minus button on a course in the week view. The addition and removal of courses becomes a two-step process which will likely reduce the number of mistakes. We also believe the change will help users discover the ability to add and remove multiple courses with one apply operation.

These changes aim to provide a better user experience by improving the workflow of tasks and better aligning the application with users' mental models.

Section 4 – High Fidelity Prototype

4.1 – Design Document: The Course Planner

The Course Planner's vision is to provide students with the ability to create course schedules that best suit their preferences and lifestyle. The application is designed around the user's needs, providing them the tools they need to plan their academic semesters. This is facilitated through tools that assist users in finding courses that they want to take, at times that fit their schedule.

The Course Planner is a web-based application best suited for devices with large displays. The web was chosen because it is platform independent. Web-based applications do not have to be redesigned to align with each operating system's design standards and only need to be implemented once. The use of a large interface allows users to see the course list and week view at the same, which suits the direct manipulation design approach.

The application layout is broken into two main components. The first component is the course list, which provides users with access to all the available courses. The second component is the week view. Both components are used together to find the courses and course times that meet the user's needs. The application can be set for the fall or winter semester using the toggle in the top right corner. This allows users to quickly and conveniently build the fall and winter timetables at the same time and balance course difficulty across them. The header changes color with the semester providing an easy way to determine which semester is being edited.

Using the course list, the user can browse all available courses during the semester and use multiple filtering options to aid in the search. The two main filtering features provided to the user are the search bar and the advanced filtering menu. The results of the filters are applied to the list in real time, keeping the course list is sync with the current filtering options. Additional information about a course is available via the course information button (which is visible when a course is expanded). Once expanded, course in the course list also provide the user with all available timeslots for that particular course. Users can select a timeslot's corresponding checkbox to add a course to the apply queue and immediately see a visual representation of were the course fits onto the timetable. Additionally, the course list can detect course conflicts (indicated visually) and prevent the user from adding overlapping courses to their schedule.

The week view is a visual representation of the current state of the timetable. It provides constant feedback on the state of the timetable, displaying all courses added as well as courses queued to be added or removed. With the constant visual representation, users can see where conflicts exist and how their courses are distributed throughout the week. To help fill gaps in the timetable, the user can click an open timeslot to see courses offered at that time.

All the features come together to create an intuitive application with a full suite of course selection features. The large week view is the center of it all, allowing users to easily interpret the current state of the timetable and see the results of their interactions. The application meets the needs of the user by giving them a means to create a timetable that aligns with their study preferences and desired lifestyle.

4.2 – Source Code Documentation and Running the Application

Running the application's code is very straightforward, as the application uses static web pages. Navigate to the enclosed "High fidelity prototype" folder and open index.html in your web browser. The code is structured into four separate categories: fonts, CSS, JavaScript, and HTML markup.

We made use of a web framework called Materialize which provides common and consistent design patterns throughout the application. Each JavaScript object in our code is created within a closure to ensure that its properties are not globally accessible.

The application consists of a sidebar, a top bar, and a calendar (table). Each view is marked up in HTML, styled with CSS, and controlled with its own controller.

Controllers manage the model (object state) and the rendered view. This application's controllers consist of cal.js, and sidebar.js. Cal.js manages the calendar, which consists of handling DOM manipulations for the timetable, and reacting to certain events triggered by user interactions (such as clicking on a timeslot). Sidebar.js handles search queries, filtering, and queuing courses for addition to the timetable by interacting with cal.js.

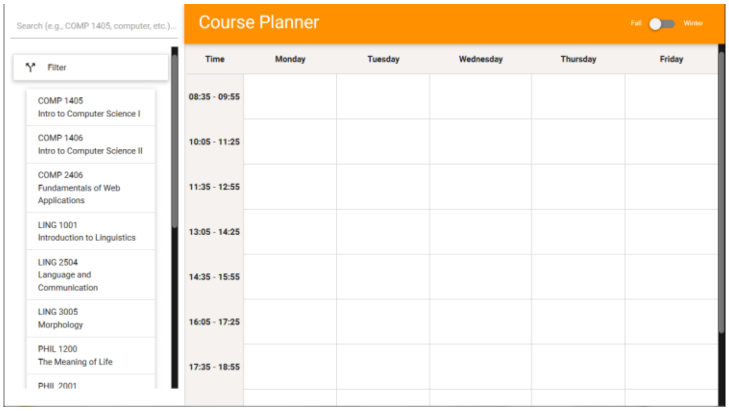
Models maintain the state of the application and are interpreted by the controller. Courses.js and profile.js are objects that represent the underlying structure of the application. All information about the user or the courses is rendered based on information managed by these files. These files are in the bottom of the DOM body, to prevent blocking when loaded.

To reiterate, running the application only requires opening index.html file in a web browser. For ease of access, the private GitHub repository used during the development of the application also serves as a demo. The application can be accessed by browsing to http://srowhani.github.io/3008/.

4.3 – Help Documentation

Main Screen

Upon opening the application, you will be presented with the main calendar view. This view displays your weekly timetable, in addition to courses queued for addition or removal.



COMP 1405
Intro to Computer Science I

COMP 1406
Intro to Computer Science II

COMP 2406
Fundamentals of Web Applications

LING 1001
Introduction to Linguistics

LING 2504
Language and Communication

LING 3005
Morphology

PHIL 1200
The Meaning of Life

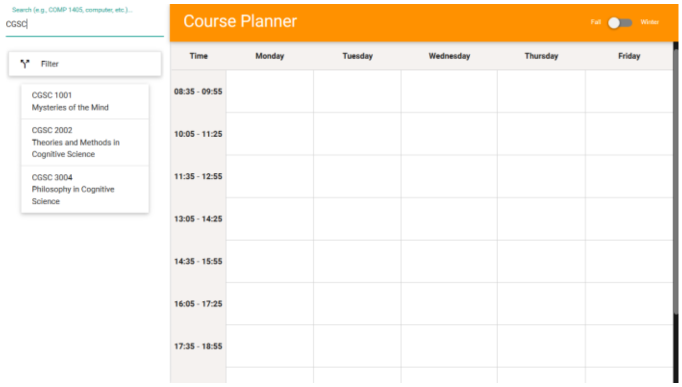
PHIL 2001

Time	Monday	Tuesday	Wednesday	Thursday	Friday
08:35 - 09:55					
10:05 - 11:25					
11:35 - 12:55					
13:05 - 14:25					
14:35 - 15:55					
16:05 - 17:25					
17:35 - 18:55					

The main page of the application.

Searching for courses

Searching is performed using the left sidebar. Use the name or code of a course in the search field to refine the resulting course list (for example, you can search for introductory courses using the search query “introduction” or “1001”).



CGSC

CGSC 1001
Mysteries of the Mind

CGSC 2002
Theories and Methods in Cognitive Science

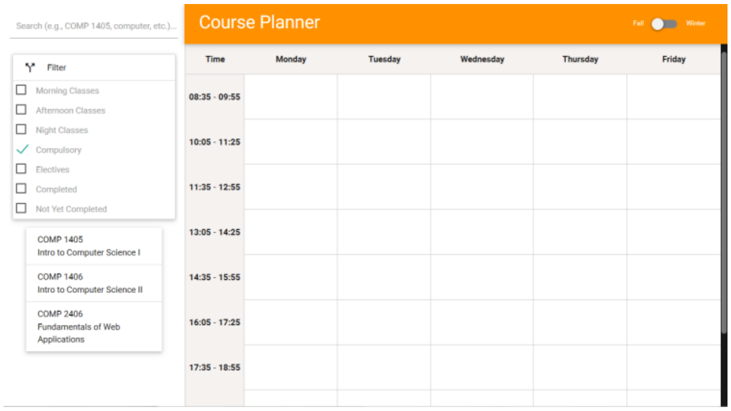
CGSC 3004
Philosophy in Cognitive Science

Time	Monday	Tuesday	Wednesday	Thursday	Friday
08:35 - 09:55					
10:05 - 11:25					
11:35 - 12:55					
13:05 - 14:25					
14:35 - 15:55					
16:05 - 17:25					
17:35 - 18:55					

Note the sidebar. The search functionality has been used to find cognitive science courses using the search query “CGSC”.

Advanced searching (filtering)

Click the filter dropdown below the search field to expand it and view the advanced search options. Selecting one or more of the checkboxes will filter the search results even further based on the selected criteria. This is useful for finding courses which must be taken or those occurring at particular times of day.



□ Morning Classes
□ Afternoon Classes
□ Night Classes
✓ Compulsory
□ Electives
□ Completed
□ Not Yet Completed

COMP 1405
Intro to Computer Science I

COMP 1406
Intro to Computer Science II

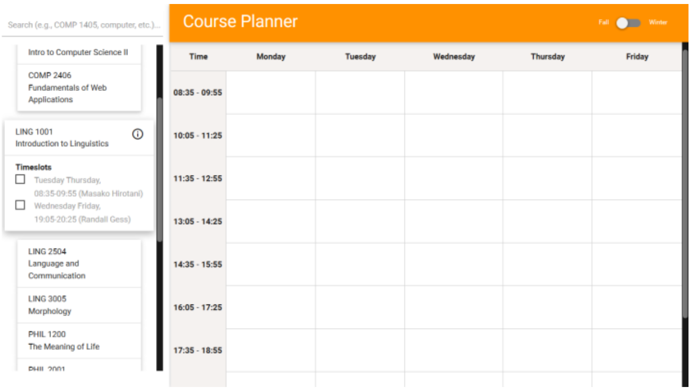
COMP 2406
Fundamentals of Web Applications

Time	Monday	Tuesday	Wednesday	Thursday	Friday
08:35 - 09:55					
10:05 - 11:25					
11:35 - 12:55					
13:05 - 14:25					
14:35 - 15:55					
16:05 - 17:25					
17:35 - 18:55					

The filtering functionality has been used to only display courses which are compulsory.

Viewing information about a search result

To view more information about a course in the search results list, click on it. It will expand and show the available timeslots and an information button will appear. This button can be clicked to view even more information about the course.



Intro to Computer Science II

COMP 2406
Fundamentals of Web Applications

LING 1001
Introduction to Linguistics

Timeslots
□ Tuesday Thursday, 08:35-09:55 (Masako Hirotani)
□ Wednesday Friday, 19:05-20:25 (Randall Gess)

LING 2504
Language and Communication

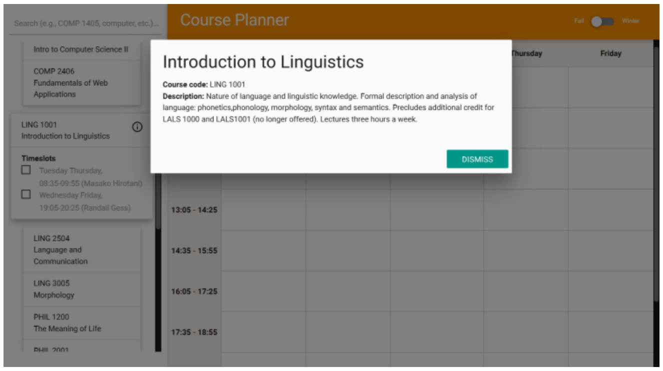
LING 3005
Morphology

PHIL 1200
The Meaning of Life

PHIL 2001

Time	Monday	Tuesday	Wednesday	Thursday	Friday
08:35 - 09:55					
10:05 - 11:25					
11:35 - 12:55					
13:05 - 14:25					
14:35 - 15:55					
16:05 - 17:25					
17:35 - 18:55					

After expanding a search result, its available timeslots become visible.



Introduction to Linguistics

Course code: LING 1001
Description: Nature of language and linguistic knowledge. Formal description and analysis of language: phonetics,phonology, morphology, syntax and semantics. Precludes additional credit for LALS 1000 and LALS1001 (no longer offered). Lectures three hours a week.

Intro to Computer Science II

COMP 2406
Fundamentals of Web Applications

LING 1001
Introduction to Linguistics

Timeslots
□ Tuesday Thursday, 08:35-09:55 (Masako Hirotani)
□ Wednesday Friday, 19:05-20:25 (Randall Gess)

LING 2504
Language and Communication

LING 3005
Morphology

PHIL 1200
The Meaning of Life

PHIL 2001

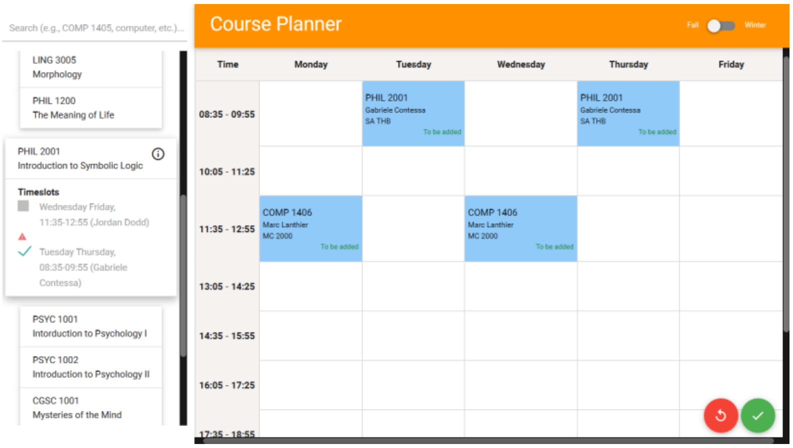
After clicking the info button, a dialog will be opened to display additional course information.

Adding courses to the timetable

Courses are added using a queueing process. Changes must first be queued, and then applied. This allows you to see the outcome of your actions before they are carried out. There are multiple ways to queue courses for addition to the timetable:

Queueing course additions via search results

After clicking a search result to expand it, a timeslot's checkbox can be checked to queue the course's addition to the timetable. This will cause the course to temporarily appear in blue on the timetable. If there is a course conflict (i.e., a course has already been added in the same timeslot), the checkbox will be disabled and hovering over it will highlight where the conflict lies.



LING 3005
Morphology

PHIL 1200
The Meaning of Life

PHIL 2001
Introduction to Symbolic Logic

Timeslots

▨ Wednesday Friday, 11:35-12:55 (Jordan Dodd)

✓ Tuesday Thursday, 08:35-09:55 (Gabriele Contessa)

PSYC 1001
Intorduction to Psychology I

PSYC 1002
Introduction to Psychology II

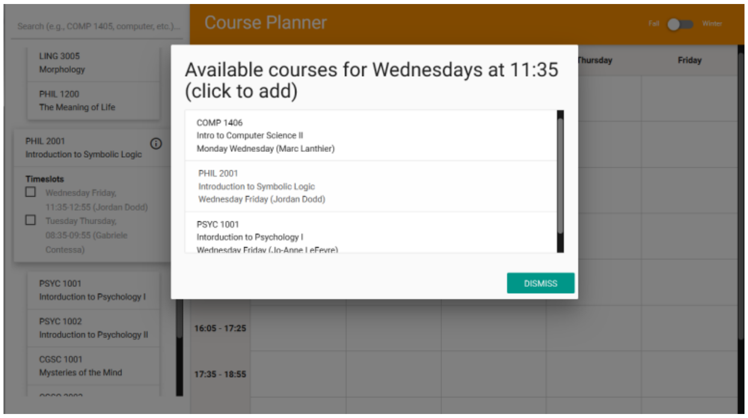
CGSC 1001
Mysteries of the Mind

Time	Monday	Tuesday	Wednesday	Thursday	Friday
08:35 - 09:55		PHIL 2001 Gabriele Contessa SA THB To be added		PHIL 2001 Gabriele Contessa SA THB To be added	
10:05 - 11:25					
11:35 - 12:55	COMP 1406 Marc Lanthier MC 2000 To be added		COMP 1406 Marc Lanthier MC 2000 To be added		
13:05 - 14:25					
14:35 - 15:55					
16:05 - 17:25					
17:35 - 18:55					

Two courses have been queued for addition to the timetable by selecting the appropriate checkbox. Note that the checkbox for the first timeslot of PHIL 2001 is disabled due to the conflicting course COMP 1406.

Queueing course additions via timeslots

It may be desirable to add courses to the timetable based on timeslot (e.g., if there is a gap in the day that you would like to fill). Clicking on an empty timeslot will display a list of available courses which occur at that time and clicking on one will queue it for addition.



LING 3005
Morphology

PHIL 1200
The Meaning of Life

PHIL 2001
Introduction to Symbolic Logic

Timeslots

□ Wednesday Friday, 11:35-12:55 (Jordan Dodd)

□ Tuesday Thursday, 08:35-09:55 (Gabriele Contessa)

PSYC 1001
Intorduction to Psychology I

PSYC 1002
Introduction to Psychology II

CGSC 1001
Mysteries of the Mind

Available courses for Wednesdays at 11:35 (click to add)

COMP 1406
Intro to Computer Science II
Monday Wednesday (Marc Lanthier)

PHIL 2001
Introduction to Symbolic Logic
Wednesday Friday (Jordan Dodd)

PSYC 1001
Intorduction to Psychology I
Wednesday Friday (Jo-Anne LeFevre)

After clicking on the empty Wednesday 11:35 timeslot, a list of courses which can be taken then is displayed.

Adding queued courses to the timetable

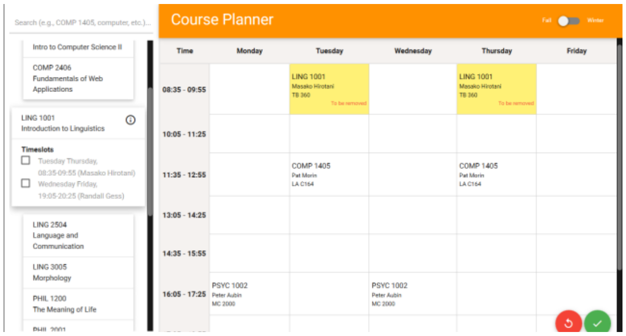
Temporarily queued courses are confirmed and added by clicking the green apply button at the bottom right of the screen. Multiple courses can be queued before applying your changes. They can be dequeued by clicking the red undo button, which is located to the immediate left of the apply button. These buttons are only visible when there are unapplied changes.

Removing courses from the timetable

Courses are removed using a similar queueing process as course addition. Changes must first be queued, and then applied. This allows you to see the outcome of your actions before they are carried out. There are multiple ways to remove courses from the timetable:

Removing courses via search results

After clicking a search result to expand it, a timeslot's corresponding checkbox can be unchecked to mark the course for removal from the timetable in that timeslot. This will cause the course to temporarily turn yellow on the timetable.

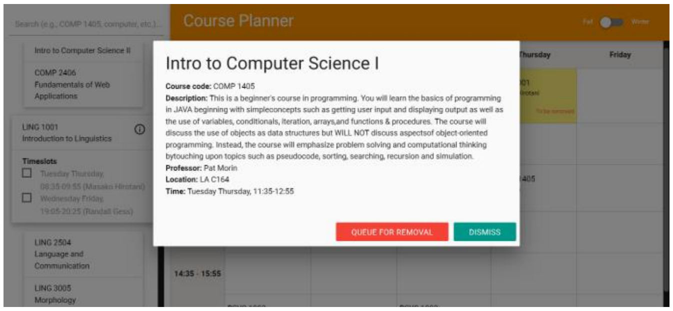

Search (e.g., COMP 1405, computer, etc.)...
Intro to Computer Science II
COMP 2406 Fundamentals of Web Applications
LING 1001 Introduction to Linguistics
Timeslots
□ Tuesday Thursday, 08:35-09:55 (Masako Hirotani)
□ Wednesday Friday, 19:05-20:25 (Randall Gess)
LING 2504 Language and Communication
LING 3005 Morphology
PHIL 1200 The Meaning of Life
PHIL 2001
Course Planner
Fall
Winter

Time	Monday	Tuesday	Wednesday	Thursday	Friday
08:35 - 09:55		LING 1001 Masako Hirotani TB 360 To be removed		LING 1001 Masako Hirotani TB 360 To be removed	
10:05 - 11:25					
11:35 - 12:55		COMP 1405 Pat Morin LA C164		COMP 1405 Pat Morin LA C164	
13:05 - 14:25					
14:35 - 15:55					
16:05 - 17:25	PSYC 1002 Peter Aubin MC 2000		PSYC 1002 Peter Aubin MC 2000		

The checkbox for LING 1001's Tuesday/Thursday timeslot has been unchecked, causing it to be marked for removal from the timetable.

Removing courses via info dialogs

Courses which have been added to the timetable can be clicked to view more information about them. This dialog also features a button to queue the selected course for removal.



Intro to Computer Science I

Course code: COMP 1405
Description: This is a beginner's course in programming. You will learn the basics of programming in JAVA beginning with simpleconcepts such as getting user input and displaying output as well as the use of variables, conditionals, iteration, arrays,and functions & procedures. The course will discuss the use of objects as data structures but WILL NOT discuss aspectsof object-oriented programming. Instead, the course will emphasize problem solving and computational thinking bytouching upon topics such as pseudocode, sorting, searching, recursion and simulation.
Professor: Pat Morin
Location: LA C164
Time: Tuesday Thursday, 11:35-12:55

A course on the timetable has been clicked, causing an information dialog to be shown. The red "queue for removal" button can be used to mark the course for removal from the timetable.

Removing queued courses to the timetable

Courses marked for removal are confirmed and removed by clicking the green apply button at the bottom right of the screen. They can unmarked by clicking the red undo button, which is located to the immediate left of the apply button. These buttons only appear are only visible when changes are queued.

Section 5 – Usability Inspection Evaluation

5.1 – Heuristic Evaluation

After the high fidelity prototype was built, each team member conducted a heuristic evaluation. Each evaluator went through the interface considering Nielsen's ten suggested heuristics. Problems found were prioritized into the following four categories: 1– cosmetic problem, 2 – minor usability problem, 3 – major usability problem and 4 – usability catastrophe. The following are the results from the testers' evaluations.

Matthew Penny

Visibility of system status

The system status is consistently visible throughout the application since everything is updated in real-time as actions are performed. It may be desirable to show more contextual information regarding the user (i.e., their degree program), however this is probably inessential (1) since users will likely know this information and can already filter compulsory courses.

Match between the system and the real world

The system uses familiar phrases rather than codes to convey information to the user, and as such generally mirrors the real world. A calendar similar to a conventional one is employed along with clearly labelled weekdays and timeslots.

User control and freedom

Users are able to undo the intermediate course queueing step, but once they have done so, they cannot redo it. Also, applied changes cannot be undone. These can be considered minor usability problems (2), since queueing courses is relatively fast and meant to be temporary.

Consistency and standards

Standard icons, colors, and fields are used (e.g., green check mark to apply, red arrow to undo). However, the course information dialog's content may differ depending on where it is invoked from. This may confuse users briefly, but can be considered a minor, aesthetic issue (1) as the variance is not large (the omission of timeslot information). It is also possible to queue courses for removal from their detail view, but not queue them for addition. This is a minor usability problem (2) because it is not what users are expecting and does not conform to expectations, but there are other obvious ways to accomplish the same outcome.

Error prevention

The system makes use of an intermediate queueing step when adding and removing courses. This makes adding and removing courses very deliberate actions which are unlikely be performed accidentally. Additionally, queued changes are indicated visually in real-time. One issue with this behavior is that it is possible to undo queued actions but not redo them. If a user accidentally undoes the queuing step, they must repeat each of their prior actions again. Applied changes also cannot be undone. These are minor usability problems (2) since the queueing of courses is relatively fast.

Recognition rather than recall

Direct real-time visual feedback is used to indicate actions and their results. The interface displays information only when it is contextually relevant (e.g., an info button appears when a search result is expanded), indicating applicability to previous actions. Tooltips and labels are used to indicate functionality which may not otherwise be obvious but this labelling absent for course conflicts (only a red warning symbol and rectangle are used to indicate them). Tooltips or labelling could be added to make conflicts more clear. This is a minor aesthetic issue (1) since there are already two visual elements that indicate course conflicts.

Flexibility and efficiency of use

The system features accelerators such as filtering and batch additions and removals of courses. There are also multiple ways to add and remove courses. Although it is possible to remove courses from their detail view, it is not possible to add them this way. Courses also cannot be queued for removal until they have been added, requiring an extra step in the process. Both of these issues are minor (2) since the actions can still be completed in other ways.

Aesthetic and minimalist design

Information is generally only shown when it is contextually relevant. The interface shows more or less information depending on the task the user is engaged in (e.g., detailed course information is not shown until the user opens the detail view).

Help users recognize, diagnose, and recover from errors

Courses are added and removed using an intermediate queuing process, indicating what will happen before it does. In the event a mistake does occur in course queuing, users can undo their actions. However, this can only be done prior to the application of changes. Users must manually undo their actions once they have been applied. This can be seen as a minor usability problem (2) because the queuing step indicates what will happen if the user applies their changes, making mistakes unlikely.

Help and documentation

The help document is thorough but new users may not know where to start which could be classified as a minor usability problem (2). However, the application provides feedback whenever actions are performed, making their effects fairly apparent and encouraging experimentation.

Seena Rowhani

Visibility of system status

No issues could be found in this regard. Every action results in some sort of feedback, making it very easy for the user to understand the effects of each action.

Match between system and the real world

The application uses common culturally known terms to describe the intent to actions. Focus works left to right, indicating to the user that their first action will most likely be filtering the courses that they'll need to take. Then when it comes to adding courses, the actions are described at each step, making it clear what needs to be done.

User control and freedom

Users have the ability to revert actions at any point of time. Searching is reactive, meaning that to go back to a search result all that needs to be done is re-enter what was entered in certain fields. Actions have states, so there are intermediate steps when adding or removing a course. This makes it possible to undo actions. However, doing so is limited to the intermediate queuing step, which I would classify as a minor usability problem (2).

Consistency and standards

In terms of stylistic design, all elements appear to follow same font sizing, color scheme, padding, and margins. Phrasing is consistent as well.

Error prevention

Every action has immediate feedback, making constant interactions a feasible way of using the application. At any point between filtering, adding, or removing the application state is quite apparent, so the user should know where they stand based on this. No need for user to remember chain of actions because the only relevant action is the current one.

Recognition rather than recall

The user will not need to manage action history. Options are always apparent, and the application is minimal in such a way that everything needed to be manipulated is less than one or two actions away.

Flexibility and efficiency of use

As for accelerators, filtering is limited to a certain degree (it is not able to search on specific fields). Other ways of adding or filtering courses is by clicking on a time slot, which is more intuitive to some. The good thing about the accelerators implemented is that feedback is instant, due to the reactive nature of filtering on key press.

Aesthetic and minimalist design

Aesthetically, the application holds its own. Feedback is only shown when necessary; the application is as minimal as possible. Material design follows the minimalist design ideology, so overall nothing is wrong here.

Help users recognize, diagnose, and recover from errors

Error messages are lacking in some regard, as the application focuses more error prevention than error handling. I would classify this as a minor usability problem (2), as it wouldn't affect the outcome of using the application, only the way some users would potentially use it.

Help and documentation

Documentation is at minimal, therefore this can be argued as a minimal usability issue (2). Although it is intuitive what actions are needed to be completed, actions need to be performed to observe causal effects.

Alexander Maxwell

Visibility of system status

The visibility of the system status keeps the user informed by providing visual feedback such as filtering in real time, and the state of the timetable being displayed in the week view. There could be improvements with other forms of feedback, such an auditory feedback. Using the appropriate sounds would reconfirm the results of actions, but is not considered necessary for the average user of the application. I would classify auditory feedback as a minor usability problem (2).

User control and freedom

The application provides users with a lot of control and freedom when completing tasks. Features such as undo are provided to clear the apply queue. Removing a single course from the apply queue is a common issue that is difficult to perform. Currently, the user would have to find the course in the course list, to unselect it. This is a problem if the user has moved on to another course. Being able to remove a course from the apply queue using the week view, would be a way around this. I would classify this issue as a minor usability problem (2).

Consistency and standards

The consistency of how tasks are preformed in the application is important to the usability. One component of the interface that lacks consistency is the information dialog. The information fields being displayed depend on the context the dialog was opened from. Also, selecting a course from the week view provides a removal option, but selecting a course from the course list does not provide a method for adding the course. I would classify this inconsistency as a minor usability problem (2).

Error prevention

The application does a good job at managing error prevention. An example would be the inability to add two courses that have conflict times to the apply queue. Visual feedback is given for actions in real-time through the course list and week view, which also helps. This allows users to confirm that the results of their actions are what they intend.

Recognition rather than recall

Courses displayed on the week view only use course codes to label the course tile. This forces users to recall the actual name of the course. To address this issue, the user can click the course tile to view the course information dialog, but this process would have to occur multiple times if many courses were on the timetable. I would classify the problem as a minor usability problem (2).

Flexibility and efficiency of use

The Course Planner provides a lot of flexibility for advanced users to accelerate the timetable creation process. Examples of accelerators include the ability to add and remove many courses at once.

Aesthetic and minimalist design

Most of the application follows a minimalist design philosophy. One component that could use improvement is the course timeslot tiles in the course list. The tile provides the instructor next to the day and time the course occurs. Although the instructor may be unique to each timeslot, it is not pertinent information when selecting a timeslot and makes the timeslot tile seem cluttered. I would classify this issue as a cosmetic problem (1).

Help users recognize, diagnose, and recover from errors

When trying to diagnose a course conflict, the application provides two symbolic representations. Although the symbols suggest there is an issue, there is no indication that a course conflict is the problem. I would classify the issue as a major usability issue (3) because if the problem cannot be diagnosed the user will not know how to proceed.

Help and documentation

The help documentation instructs users how to complete basic tasks and highlights some features while doing so. The help documentation could have provided some information on resolving errors. An example would be how to address course conflicts. I would classify this issue as a minor usability error (2).

Consolidation

Every evaluator will have different results for the heuristic evaluation. There are an endless number of ways in which an application can be improved. From the results of the heuristic evaluations there were a few key problems that stood out to the evaluators.

The first feature the evaluators mentioned was the timeslot conflict feature. The timeslot conflict feature prevents the user from adding a course that overlaps with the time of another. Although the application provides an indication that the user cannot add the course, there is no explicit feedback that states a course conflict is the problem. If the user needs to add the course, they will not know how to proceed if they can’t determine the error is a course conflict. It was indicated by an evaluator that the user must recall the fact that courses can not conflict times. One evaluator suggested the issue could be resolved by providing users with tooltips or other error indicators to help make the problem more apparent. The evaluators came to the consensus that the issue is a major usability problem (3) because when the error occurs, it could potentially prevent the user from creating their timetable with the needed course.

The second problem area evaluators agreed on was the undo feature for the apply queue. Two problems raised about this feature were the inability to individually remove courses from the queue and the lack of recoverability once the undo button is clicked. If a user has multiple courses in the apply queue, it is not possible to remove a single course from the queue. The user must undo the entire queue or apply the changes in the queue and then remove the course(s) of interest. The second issue with the undo feature is the inability to recover the apply queue once the undo button has been clicked. This will likely result in frustrated users in the event the button is accidentally clicked, as all changes in the queue will have to be done again. The evaluators agreed to classify the issue as a minor usability issue (2), as the user can recover from the problem manually, even though it may be inconvenient.

The third issue raised by evaluators was the lack of help documentation. The evaluators looked at help provided in the application as well as in the help document. All evaluators agreed that the help documentation was rather sparse and could be improved by providing information on how to get started and how to recover from error situations. It was also suggested that the application could provide users with important information for getting started when using the application for the first time. The evaluators agreed that the help documentation is a minor usability problem (2), since the application is fairly intuitive to use.

The fourth issue identified was the inconsistency of information and functionality of the course information dialog across different contexts. When the course information dialog is opened from the course list, the context of the information in dialog is the course as a whole and thus it does not show information about the timeslot, location, or professor. When the user opens the information dialog from the week view, the context of the information is for that particular timeslot and corresponding information is shown in addition to the standard course information. This may result in user confusion due to unexpected information. Additionally, courses can be queued for removal when the information dialog is opened from the week view, but the functionality for course additions is not provided when it is opened from the course list. Although this behavior may not be what users initially expect, it is fairly easy to understand and adapt to. For this reason, the evaluators classified the issue as a minor usability issue (2).

Although small, these changes will improve the usability of the application. By providing feedback and documentation on recovering from errors, users will be better prepared to handle problems when they arise. Having preventative measures to stop users from undoing all changes in the apply queue will prevent headaches if the undo button is wrongfully selected. Finally, having consistency amongst common user interface components will help align the behavior of the application with user’s expectations.

Section 6 – Workload Distribution Summary

Alexander Maxwell

- Project idea
- User interview + grounded theory
- Affinity diagram
- Low fidelity prototype
- Cognitive Walkthrough
- Overview
- Design document
- Heuristic evaluation + consolidation
- Workload Distribution Summary
- Report consolidation
- Report editing

Matthew Penny

- Project idea
- User interview + grounded theory
- Low fidelity prototype
- Storyboards
- Cognitive walkthrough
- Iteration two
- High fidelity prototype
- Help documentation
- Heuristic evaluation
- Workload distribution summary
- Report editing

Seena Rowhani

- User interview + grounded theory
- Low fidelity prototype
- Cognitive walkthrough
- Iteration one
- High fidelity prototype
- Source code documentation
- Heuristic evaluation
- Workload distribution summary

Appendix

Interviews

Interviewee 1

Person interviewed: Student, female, competes on varsity team.

What platform are you most likely to use (e.g., desktop, mobile, web)?
Web, on a computer, need the screen space.

How long do you spend creating your schedule?
Normally spend a good day making all the different arrangements, and then pick the best one.

How do you decide which courses to select?
Normally I would look it up a head of time (calendar), and figure out which classes I should take for the year. Then I pick the term (fall/winter) to take it in.

What type of information do you look for when enrolling in a class?
I like to know if there are any online versions of the class, if not, the time is most important. Instructor is important too, or if there are labs that need to be taken.

What is your biggest problem when picking classes?
Ability to coordinating class to be grouped together, as time is essential. Also blocking off times when practice occurs.

Would you like to see all classes, or just what's available for you?
I wouldn't mind seeing all the courses available, because sometimes there are interesting courses you could take as electives in the future, so you take the prerequisites. But if they weren't there it would be harder picking electives. Sometime you can override to take it.

Is there an order in which you select your classes?
Normally I would select my electives last. But it's hard to know which classes I should take in the fall and in the winter. Some classes in the winter require classes in the fall. So its takes some time sorting through which ones have to be taken in the fall.

How do you balance your time table?
Normally I would just take courses based on prerequisites. Most courses have the same workload.

Interviewee 2

Person interviewed: Student, male, third year computer science major.

What platform are you most likely to use (e.g., desktop, mobile, web)?
Web is probably best because you can sit down and work at the desk rather than sitting around on your phone. I'm more focuses that way because it feels more like sitting down to get work down. I'd get lazy on my phone. Phone screens are also smaller. You can't see as much or multitask as well.

How long do you spend creating your schedule?
Around 2.5 hours. I usually sit down and look at everything, make a draft timetable, then come back a day or two later and make the final one. "I putt in what I think I need first, and then fine tune it". I make two drafts: one rough, one final. Revise the rough one. "Throw in what I think I need, and go from there". Do this to have something to start with and work from.

How do you decide which courses to select?
Time (how it fits in my schedule), the prof teaching the course, whether or not it's necessary. This is in order to ensure I'll do better in the class. I use pretty much the same process for core courses and electives.

What type of information do you look for when enrolling in a class?
Prerequisites, whether or not it's mandatory, the professor teaching the course, the time, and if it conflicts with anything else I'm taking or need to take.

What is your biggest problem when picking classes?
Sometimes the timeslots don't work together (conflicting classes, having gaps between classes (possibly as a result of the conflicts)). "It's annoying if only one time is offered because it disrupts my day".

Would you like to see all classes, or just what's available for you?
I would want to see the ones that I still need to take, then what I am able to take for electives, and then how many first years I can take, then any I might have failed. Prioritize the list. It could also be good to show what I've already taken (good for verifying I've taken certain courses).

How do you balance your timetable?
I try to balance semesters (some easy, some hard classes), split up mandatory courses in each semester and then add in electives each semester to balance it out. Long weekends and no gaps are a bonus, not necessarily a priority (I won't it if it conflicts with my other organization strategies). I try to avoid night classes and unnecessary travel in between (i.e., gaps). I try to balance in order to make life easier and maximize time for doing work.

Interviewee 3

Person interviewed: Student, female.

What platform are you most likely to use (e.g., desktop, mobile, web)?
Pretty much I can only see myself doing it on the website.

How long do you spend creating your schedule?
Overall, I spend a little over five hours getting it done.

What type of information do you look for when enrolling in a class?
I look for the professor's name, and I usually look him up. Check if people like them or not. I look at what I need to take, and what looks easy or fun. And timing, I check if timing works well with everything else.

What is your biggest problem when picking classes?
I'd say the hardest part is getting my schedule organized. It's hard to know what courses to choose, especially since you have to use your audit to figure it out, and it's something entirely different.

Would you like to see all classes, or just what's available for you?
It'd be nice to show all courses available, it would give me an idea of what's where and I have the option to move or change focus at any point of time. Also it'd be cool to know what I can take to fill gaps.

How do you balance your timetable?
I balance usually by giving small gaps between classes, and put them at times where it's not hard to get on campus, or times easy to leave campus (not rush hour).