COSC 499 Team 6
Timetable Visualizer

Hannah Jin
Ilerioluwa Oyedele
Ross Stevenson
Kevin Van Kessel

Project Charter

Background:

Dr. Sylvie Desjardins requires a timetable tool that displays information regarding course schedules for each semester at the University of British Columbia - Okanagan Campus. Dr. Desjardins wants to be able to visualize and breakdown the education plan so that she can analyze conflicts for different programs, year level, professors' schedules, core courses, labs, seminars and activities. Currently Dr. Desjardins manages the schedule using excel spreadsheets and visualizing the schedule using drawings and sticky notes on a white board. We intend to create a tool to improve this process and make it more efficient and simple for Dr. Desjardins.

Goals:

- Deliver a product that addresses and clearly answers all the clients needs laid out in the scope statement document and project charter.
- Create a robust and useful database which is fully compatible with the education plan supplied by Dr. Sylvie Desjardins.
- Create an effective user interface that allows the user to manipulate the selected classes to different timeslots and days and save those changes into the database.
- Have our product integrated into the UBC system with CWL login and following all security regulations and processes.
- Document and instructions for client.
- Testing for frontend, backend, features and automated testing

Scope Statement:

Deliver to our client a web based application that enables the client and unit heads to visualize different timetables based upon course level, subject matter, professors schedule and unit by importing an excel file containing an education plan.

Key Stakeholders:

Client	Dr. Sylvie Desjardins
Project team members	Ilerioluwa Oyedele, Ross Stevenson, Kevin Van Kessel, Hayun Jin
Other	UBC, UBCO IT Department
Users	UBCO unit heads

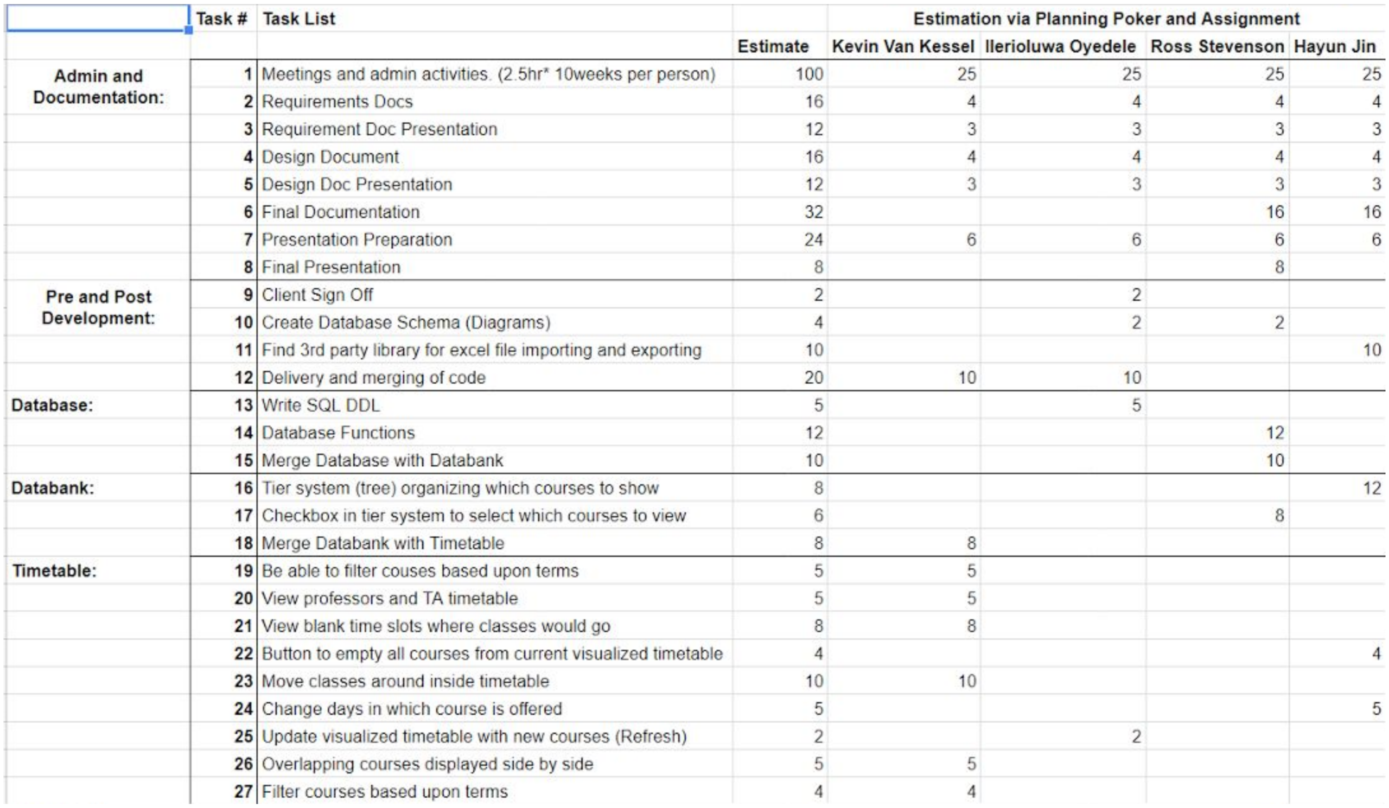
Project Milestones:

- Design document and client approval **June 23rd**
- Have a server operating with the most basic site available **June 30th**
- Establish backend database **July 8th**
- Establish excel import and export system **July 14th**
- Develop dynamic schedule board with functionality on the selected classes **20th**
- Have product hosted on UBC server with correct security and authentication **26th**
- Finish development **July 31st**
- Finish testing **August 10th**
- Deliver packaged product **August 14th**

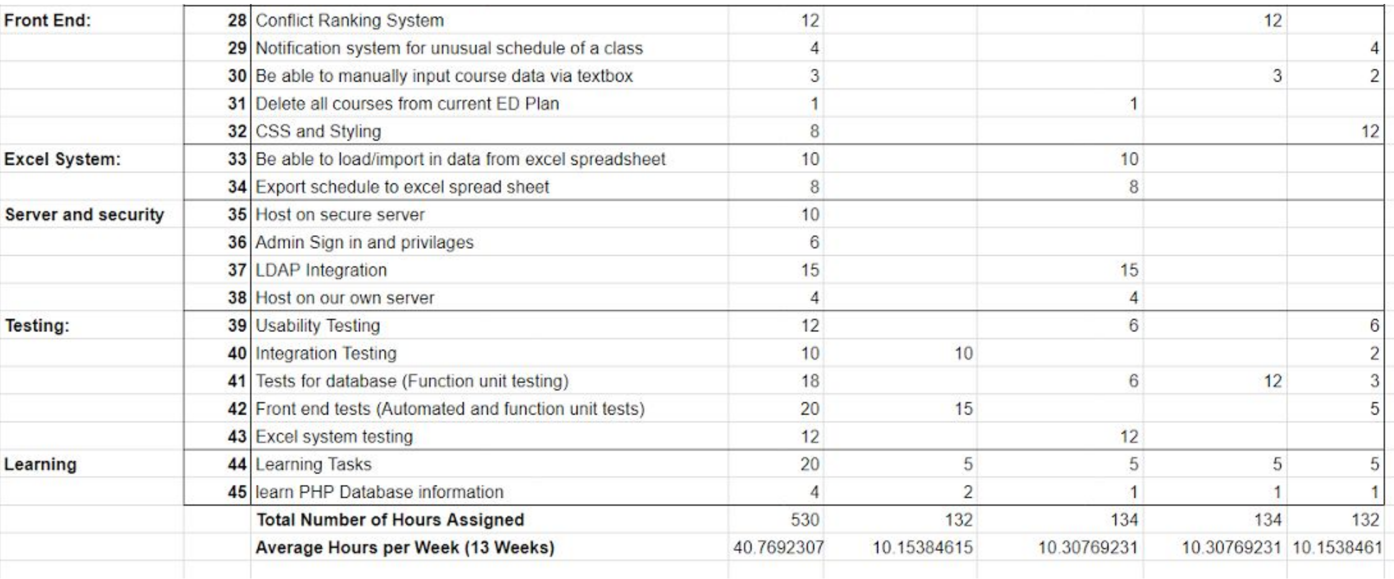
Constraints, Assumptions, Risks and Dependencies:

Constraints	Develop and deliver project by Mid August 2017. We must abide by UBC protocols if we are to host our product on their server. Any technology, libraries and licenses we use must be in accordance to UBC's policies. We have little to no financial cost. Learning new technologies
Assumptions	We will be given access to a UBC server to deliver our product to the client. We will use an external server running PHP and a MySQL database to start off with
Risks and Dependencies	We are dependent on a client server provided by the UBC IT department. The provided UBCO server may not be running the correct PHP version. The provided UBCO server may not running a MySQL database. User authentication Team members don't contribute or drop class

Work Breakdown Structure:



	Task #	Task List		Estimation via Planning Poker and Assignment			
			Estimate	Kevin Van Kessel	Ilerioluwa Oyedele	Ross Stevenson	Hayun Jin
Admin and Documentation:	1	Meetings and admin activities. (2.5hr* 10weeks per person)	100	25	25	25	25
	2	Requirements Docs	16	4	4	4	4
	3	Requirement Doc Presentation	12	3	3	3	3
	4	Design Document	16	4	4	4	4
	5	Design Doc Presentation	12	3	3	3	3
	6	Final Documentation	32			16	16
	7	Presentation Preparation	24	6	6	6	6
	8	Final Presentation	8			8	
Pre and Post Development:	9	Client Sign Off	2		2		
	10	Create Database Schema (Diagrams)	4		2	2	
	11	Find 3rd party library for excel file importing and exporting	10				10
	12	Delivery and merging of code	20	10	10		
Database:	13	Write SQL DDL	5		5		
	14	Database Functions	12			12	
	15	Merge Database with Databank	10			10	
Databank:	16	Tier system (tree) organizing which courses to show	8				12
	17	Checkbox in tier system to select which courses to view	6			8	
	18	Merge Databank with Timetable	8	8			
Timetable:	19	Be able to filter couses based upon terms	5	5			
	20	View professors and TA timetable	5	5			
	21	View blank time slots where classes would go	8	8			
	22	Button to empty all courses from current visualized timetable	4				4
	23	Move classes around inside timetable	10	10			
	24	Change days in which course is offered	5				5
	25	Update visualized timetable with new courses (Refresh)	2		2		
	26	Overlapping courses displayed side by side	5	5			
	27	Filter courses based upon terms	4	4			



Front End:	28	Conflict Ranking System	12			12	
	29	Notification system for unusual schedule of a class	4				4
	30	Be able to manually input course data via textbox	3			3	2
	31	Delete all courses from current ED Plan	1		1		
	32	CSS and Styling	8				12
Excel System:	33	Be able to load/import in data from excel spreadsheet	10		10		
	34	Export schedule to excel spread sheet	8		8		
Server and security	35	Host on secure server	10				
	36	Admin Sign in and privilages	6				
	37	LDAP Integration	15		15		
	38	Host on our own server	4		4		
Testing:	39	Usability Testing	12		6		6
	40	Integration Testing	10	10			2
	41	Tests for database (Function unit testing)	18		6	12	3
	42	Front end tests (Automated and function unit tests)	20	15			5
	43	Excel system testing	12		12		
Learning	44	Learning Tasks	20	5	5	5	5
	45	learn PHP Database information	4	2	1	1	1
		Total Number of Hours Assigned	530	132	134	134	132
		Average Hours per Week (13 Weeks)	40.7692307	10.15384615	10.30769231	10.30769231	10.1538461

Scope Statement Document:

Background/ Description:

Our client, Dr. Sylvie Desjardins, wants us to produce a system that will allow her to visualize timetables when she arranges courses for upcoming years. By visualizing timetables, she will be able to arrange classes as to minimize conflicts in classes students are likely to take at the same time. The conflicted classes will also be shown on the timetable visually, to help her to organize classes more effectively. For example, computer science students can takes computer science classes in their specific year level without having many conflicts while meeting their requirements and having options for different electives. Under the admin account, the admin will be able to import and export the education plan using a excel spreadsheet file safely. The system gives the admin better visualization of program schedules, be user friendly, and they can also be able to make any changes they wish to make and have those reflected in their files.

Functional Requirements:

- Allow users to be able to login and logout.
- Be able to load/import .xls file containing education plan into database
- Be able to make changes to the default timetable, by expanding time, adding and deleting courses, and moving courses around at different times.
- Be able to add course through manual input
- Be able to view a timetable from Monday to Friday containing time slots for classes
- Be able to filter timetable by different terms
- Be able to select which courses to view in the schedule by unit, subject, and by year level
- Be able to manually select individual courses to be displayed in timetable
- Be able to see timetables for specific professors
- Be able to unselect all courses from time table visualizer
- Be able to move classes inside timetable visualizer and have those changes saved in database
- Be able to export the changes to the education plan into a .xls file
- User have the ability to change days courses are offered in the timetable
- User will be able to click a button which refreshes timetable based upon changes in selection bar
- Generate a back end database to hold information about courses
- Conflict ranking system to display conflicts to user
- Administrative sign in where only the Admin can make changes to the timetable
- Changes in time of a course can be reflected on every day of the week for that course
- Generate a notification if amount of lecture time excels a certain threshold (3hrs)
- Visualization of timetable is colour coordinated so it is easy to understand
- Courses which overlap will be displayed side by side in that block
- Load in new Education plan without duplicating courses
- All informations are stored into database
- Users have ability to control their account
- Develop schema for database based upon Dr. Desjardins’ education plan

- Admin has ability to backup database changes into another table.

Non Functional Requirements:

- The project will be a web based project
- Friendly to Macintosh and PC users
- Website operates effectively and functions properly on desktop computers
- The project will be based on agile process
- The project will be written in HTML, CSS, and PHP.
- The project will make use of the MVC design pattern.
- The project will make use of an SQL database to store data.
- Importing and exporting will be handled using an external PHP library that can read and write Excel data.
- Host website on UBC server with reliable uptime.
- Security will be handled by means of LDAP integration, for legal purposes and requirements, and client issues.
- Legal requirements are protected via under school security protection and their IT services.
- For Frontend, responsiveness, backups
- The project is built with the codeigniter framework
- The team functions effectively and completes tasks using 1 week iterative sprints.
- The team follows an agile workflow plan.
- Client stays informed by following the communications plan laid out prior to the project commencing

Technical Requirements:

In this project, we will be using HTML, CSS, Javascript client side, PHP server side, and MySQL for handling databases, PHPExcel Library and Microsoft Excel (export and import files). We will be using Github as our version control. We will be using TravisCI for continuous integration. Trello for our team collaboration tool to build the project, social network applications such as facebook, slack and skype to communicate each other and toggl for time tracking. Security configured using LDAP system for both web and database to ensure protected information.

User Requirements:

- Be able to login and logout as administrator or user. Administration will have privileges to edit the education plan, move around sections, create and delete sections and make changes to the database. While users will only have viewing privileges on courses and the current education plan.
- Administration can reupload education plan or work on the current education plan uploaded in the database.
- User can choose one or more courses that they want to see by clicking specific courses, number of levels, faculties, etc to minimize from seeing too many courses causing confusion
- Admin can save their work and export it from our system
- Admin can allow access (system status)/ deny access to system.

Environmental Constraints:

- Our web based application is only accessible with an internet access with proper VPN. Users also need computer to access our website.
- IT support availability (data formats, Resource availability)

Risks:

- Login security
- Issues acquiring a server from UBC IT services
- Team members leave project for any reason

Design Document:

Description of software

We are building a web based application integrated into the UBC system via LDAP (Lightweight Directory Access Protocol). The application will assist in the process of creating the schedule for each semester at the University of British Columbia - Okanagan Campus. The administrator will have the ability to load in the education plan, the courses to be offered in the upcoming session, to the database and manipulate the time and day of each of the sections. As the administrator does this they have the ability to choose which courses and sections they see in the schedule at a time, giving them the ability to organize courses as to meet their own criteria. Once changes are made the administrator can export the schedule to an excel file so that any changes made are reflected in their education plan. Both users and the administrator have the ability to select which sections are displayed using the database bank which organizes courses by unit followed by subject, year level then course number and lastly by section. All users also have the ability to view the schedule of each professor in the system. To assist with the schedule making process the system will have a conflict ranking system to prioritize possible schedule issues and subtly notify the administrator of possible issues in the plan.

List of User Groups

Administrator - Associate Dean of the University of British Columbia Okanagan Campus
Superuser - The Unit Heads of the University of British Columbia Okanagan Campus

Usage Scenarios

Scenario #1: Administrator logs in and loads in a new Ed-plan.

1. Administrator clicks the login button and logs in as an administrator.
2. Administrator clicks the Import ED Plan button and imports the excel file containing the Ed-plan.
3. The data is processed, stored in the database, then displayed in the timetable.
4. On the sidebar, courses are organized hierarchically.
5. On other sidebar, Admin can manually input course code and choose professor's name that they want to view in the timetable.

Scenario #2: Administrator moves a course to a different day to resolve a conflict

1. Administrator clicks the login button and logs in as an administrator.
2. Administrator clicks unit 5 , COSC, 300, she and clicks the specific courses that she wants to deal with, in this situation, 304 and 341.
3. Administrator wants to move COSC 341 on Tuesday and Thursday, to avoid conflict with COSC 304, which is scheduled on Monday/Wednesday/Friday.
4. Administrator clicks the edit button in the course box, and clicks Tu/T, and sets the time.

Scenario #3: Superuser logs in and views their unit, edits a section, then logs out of the system.

1. Superuser clicks the Login Button and logs into the system
2. The Superuser views the blank timetable and browses the databank containing the courses in their unit.
3. The Superuser selects two sections in their unit, views them in the timetable
4. The Superuser edits when the sections are offered.
5. The Superuser logs out of the system.

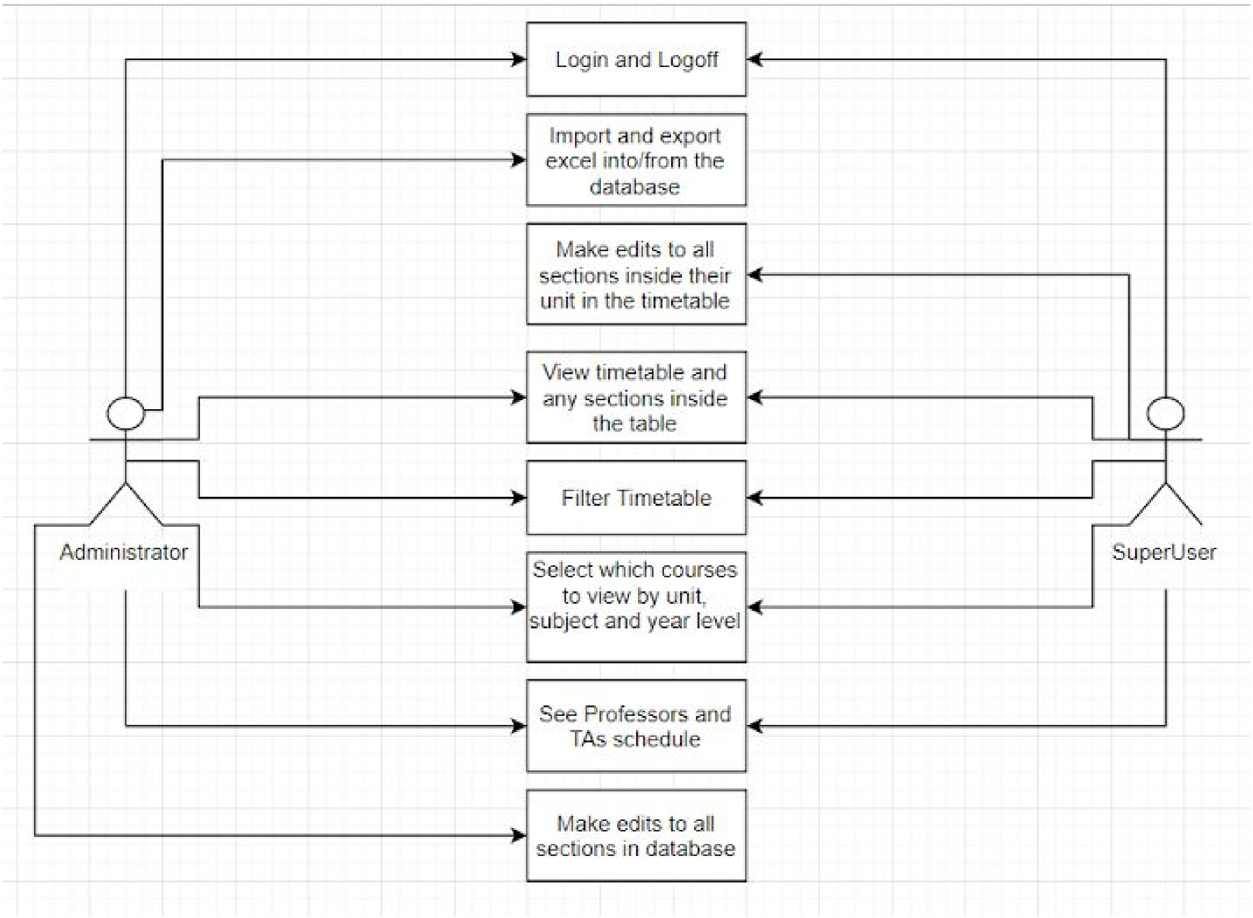
Scenario #4: Superuser logs in and edits a professor from their unit's schedule then logs out of the system

1. Superuser clicks the Login Button and logs into the system
2. The Superuser selects a name in the professor section and views that professors schedule in the timetable.
3. The Superuser selects a section from the timetable.
4. The Superuser edits the information for this section
5. The Superuser logs out of the system.

Scenario #5: Admin Logs in and exports education plan and loads in a new education plan.

1. Administrator clicks the login button and logs in as an administrator.
2. Administrator clicks the Export ED Plan button and exports the excel file containing the Ed-plan.
3. Administrator clicks the Import ED Plan button and imports the new excel file containing the Ed-plan.
4. The data is processed, stored in the database, then displayed in the timetable.
5. On the sidebar, courses are organized hierarchically.
6. On other sidebar, user can manually input course codes and choose professor names that they wants to view in the timetable.

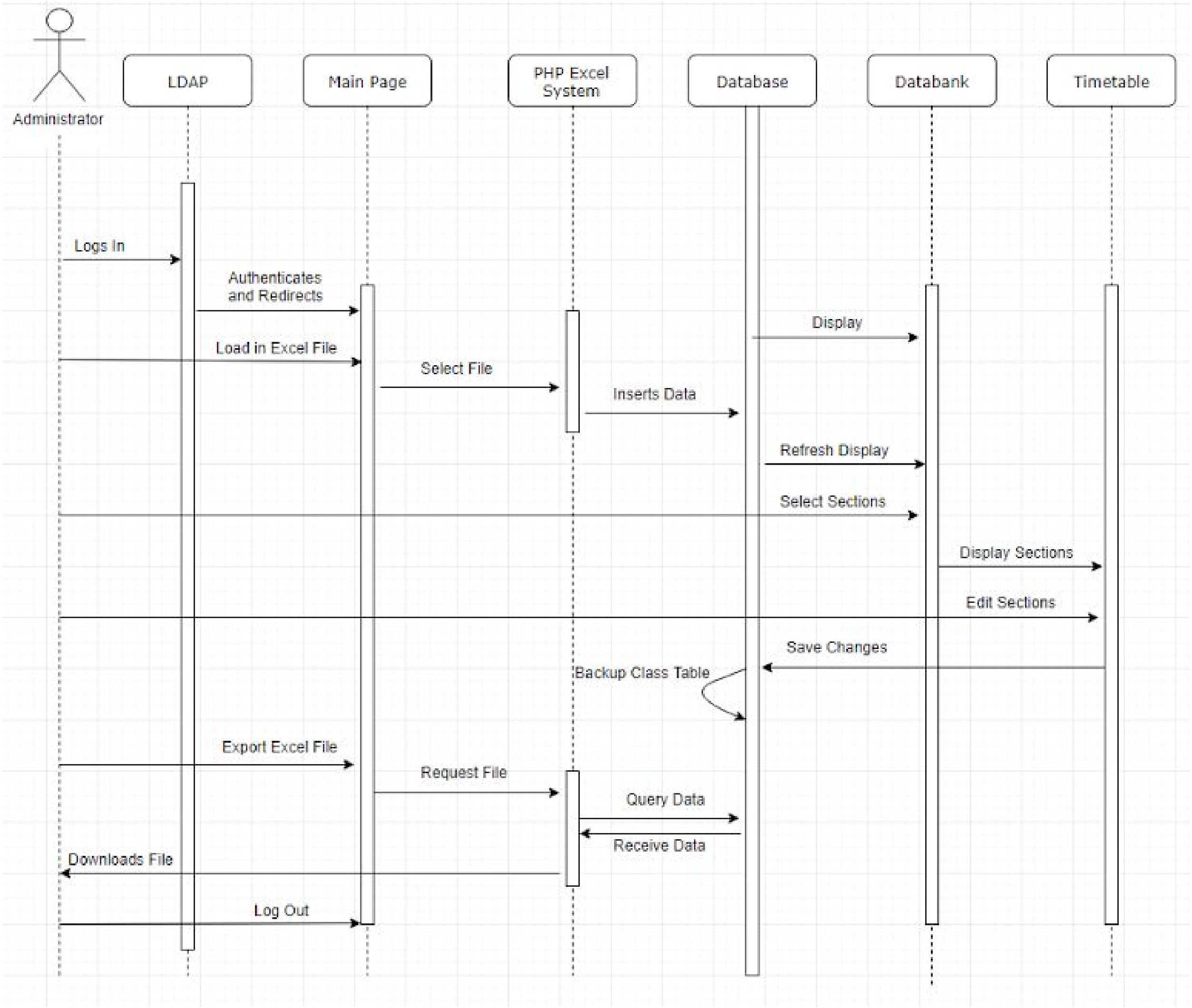
Use Case Diagram:


Login and Logoff
Import and export excel into/from the database
Make edits to all sections inside their unit in the timetable
View timetable and any sections inside the table
Filter Timetable
Select which courses to view by unit, subject and year level
See Professors and TAs schedule
Make edits to all sections in database
Administrator
SuperUser

Use case diagram displaying how the 2 sets of users, the administrator and the SuperUsers (unit heads) interact differently with the functions of the product.

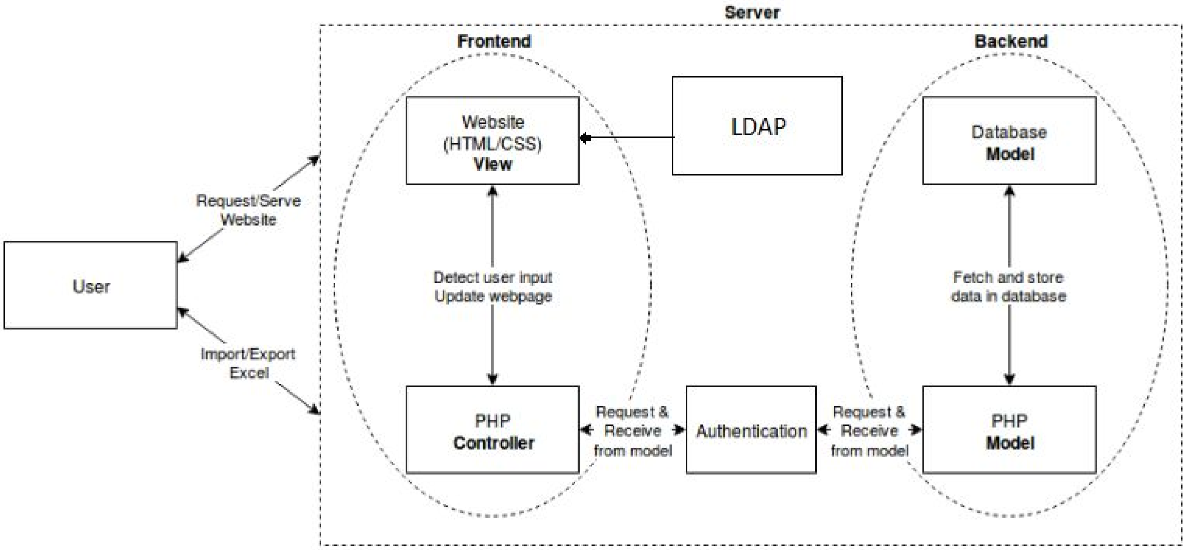
	PRECONDITIONS	**POSTCONDITIONS**
1. Login	Display signin page	Load timetable page
2. Import excel file into Database	Display import button; timetable is in default mode	Show file upload screen
3. Make edits to sections inside their unit	Display given timetable	Be able to edit selection
4. View timetable from Monday to Friday containing class schedule	Display blank table	Display table with courses
5. Filter timetable	Display timetable with schedules	Display empty timetable
6. Select which course to view by unit, subject, and year level	Checkboxes are unselected, weekly time schedule checkboxes are also unselected. Timetable is empty	Display which course that is selected in checkbox databank in the timetable.
7. See professor and TA timetable	Timetable is empty. Display dropdown menu on sidebar, where he/she can search professor or TA names.	Display whichever professor's name on timetable he/she selected in dropdown menu.
8. Export from the database	Timetables are all scheduled (all courses are scheduled)	Convert to xls. File and gets downloaded in computer
9. Make edits to all sections in the database	Monday to Friday Schedule checkbox is displayed. Course box is placed in wrong time in a week	Course schedule has been changed by shifting the course horizontally across the timetable
10. Log out	Currently logged in	Redirected to login page

Sequence Diagram



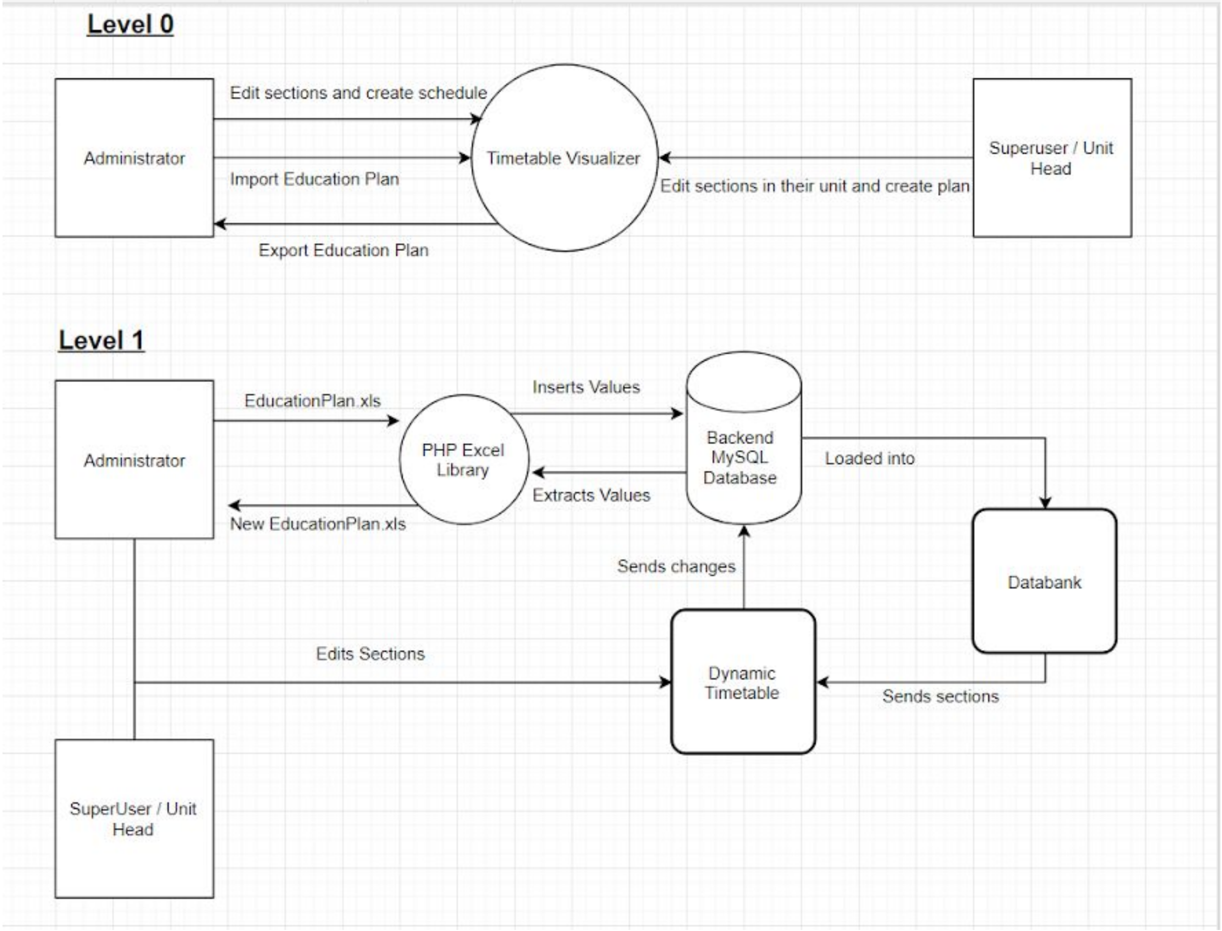
The sequence diagram shows how the user performs tasks and how those tasks cause interaction between the different pieces of the system.

System Architecture Diagram



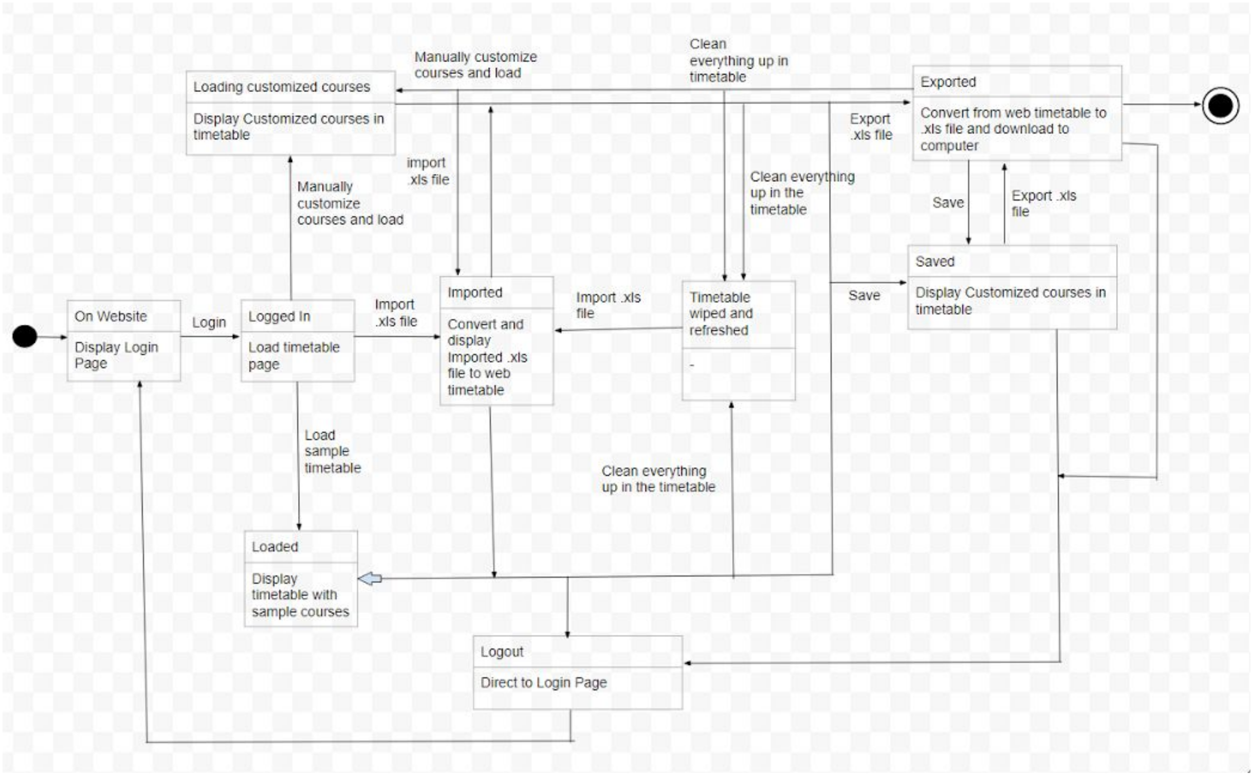
The system architecture diagram shows the structure of the system and how it behaves and interacts with the different components.

Data-flow Diagram RS



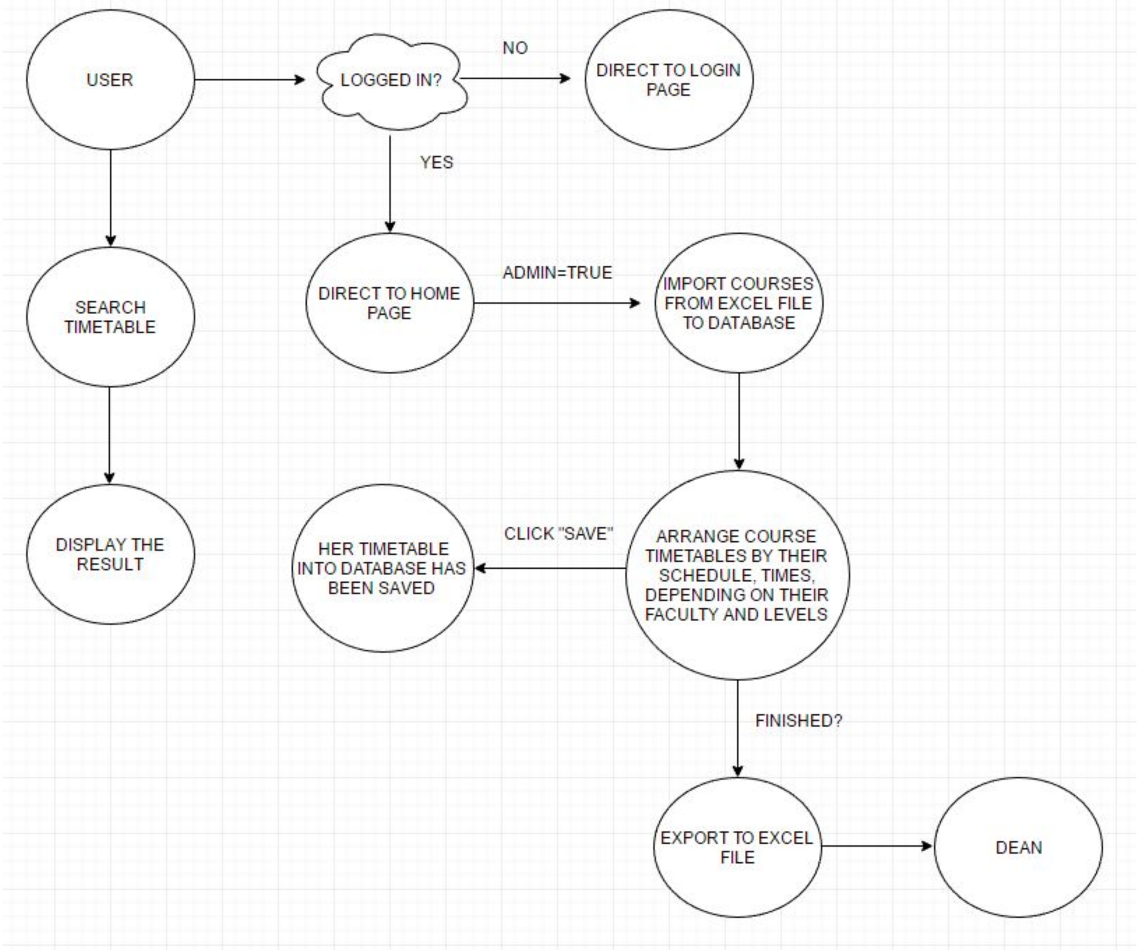
The 0 level depicts how the two types of users will use the system. The 1 level shows how the parts inside the system interact with one another and how the information flows throughout the system.

State Diagram:


Loading customized courses
Display Customized courses in timetable
Manually customize courses and load
Clean everything up in timetable
Exported
Convert from web timetable to .xls file and download to computer
Export .xls file
import .xls file
Manually customize courses and load
Clean everything up in the timetable
Save
Export .xls file
Saved
Display Customized courses in timetable
Imported
Convert and display Imported .xls file to web timetable
Import .xls file
Timetable wiped and refreshed
-
Save
On Website
Display Login Page
Login
Logged In
Load timetable page
Import .xls file
Load sample timetable
Clean everything up in the timetable
Loaded
Display timetable with sample courses
Logout
Direct to Login Page

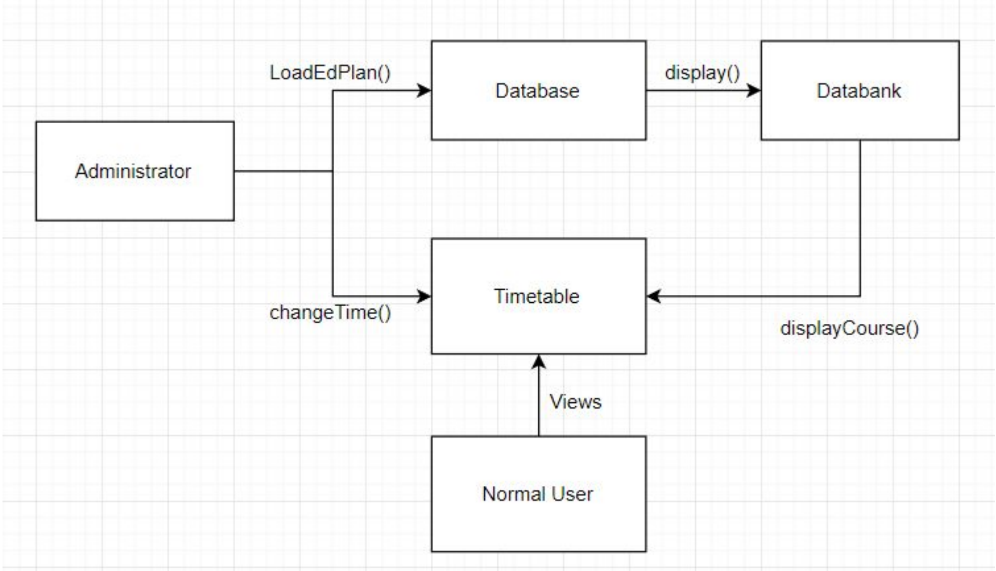
How the system interacts and moves from one state to the next as the user navigates and operates the program.

Activity Diagram


USER
LOGGED IN?
NO
DIRECT TO LOGIN PAGE
YES
SEARCH TIMETABLE
DIRECT TO HOME PAGE
ADMIN=TRUE
IMPORT COURSES FROM EXCEL FILE TO DATABASE
DISPLAY THE RESULT
HER TIMETABLE INTO DATABASE HAS BEEN SAVED
CLICK "SAVE"
ARRANGE COURSE TIMETABLES BY THEIR SCHEDULE, TIMES, DEPENDING ON THEIR FACULTY AND LEVELS
FINISHED?
EXPORT TO EXCEL FILE
DEAN

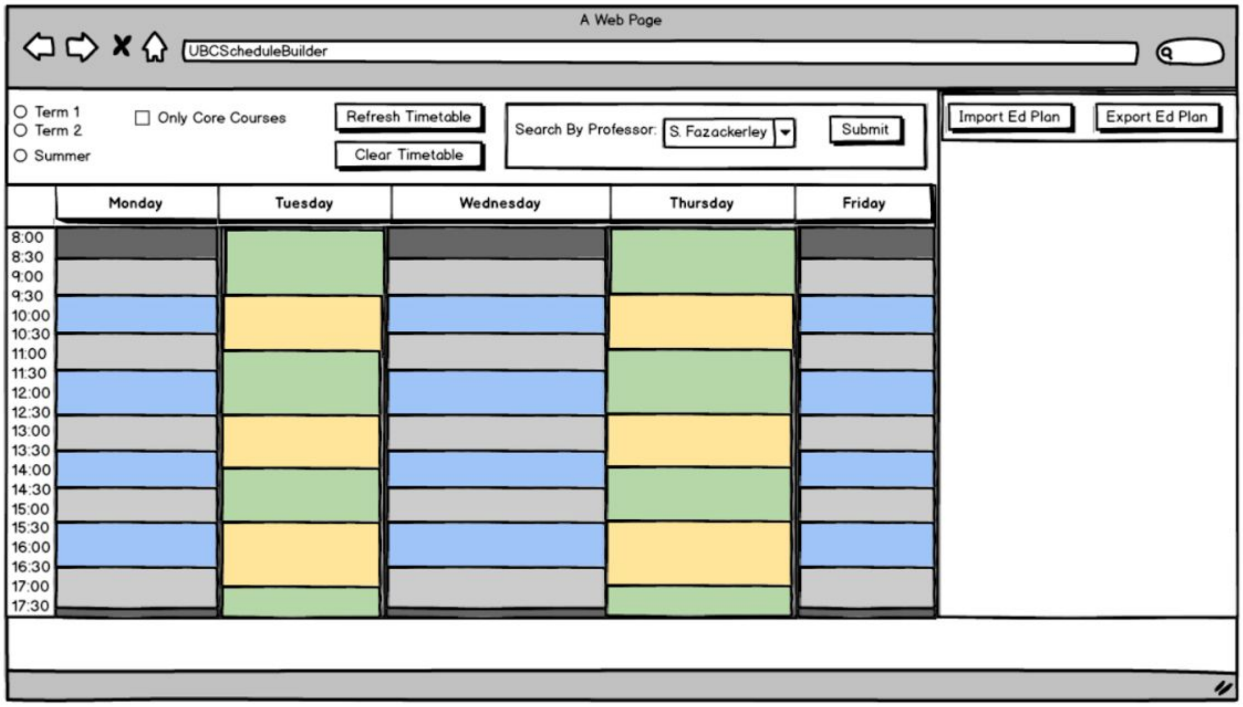
How the system flows from one activity to the next.

Collaboration Diagram

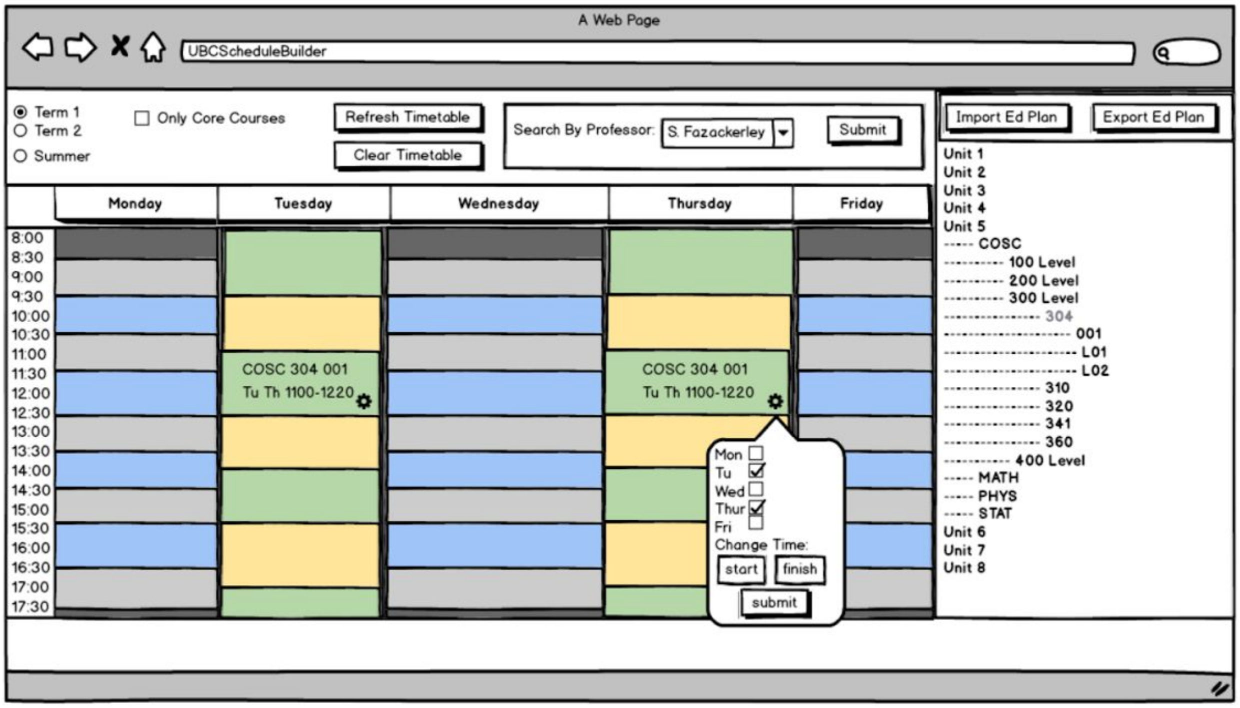


How the software interacts and communicates in order to function correctly.

Mockups Version 2.0



Search By Professor: S. Fazackerley



Search By Professor: S. Fazackerley

Mockups depicting the first concept of the project. Mockup missing logoff button and login screen. Mock Up contains main dynamic timetable and databank and shows all components necessary to perform functionality such as editing sections and searching my professor.

Tech Specifications

The technical specifications for the proposed product will be similar to most web applications, as in it will be hosted on a web server, use a backend database and have reliable performance and a understandable user interface. The product will be built using the CodeIgniter framework. We chose CodeIgniter because it is a popular, well-documented framework for building MVC based web applications. The model will consist of a mysql database and PHP to interface with it. The view will be built from the html templating capabilities that Codeigniter provides. The controller will also be handled by the CodeIgniter framework and written in PHP. A server provided by UBC's IT services will be needed to host the product to allow remote access. Security will be needed as well since this product will be handling important information for the university.

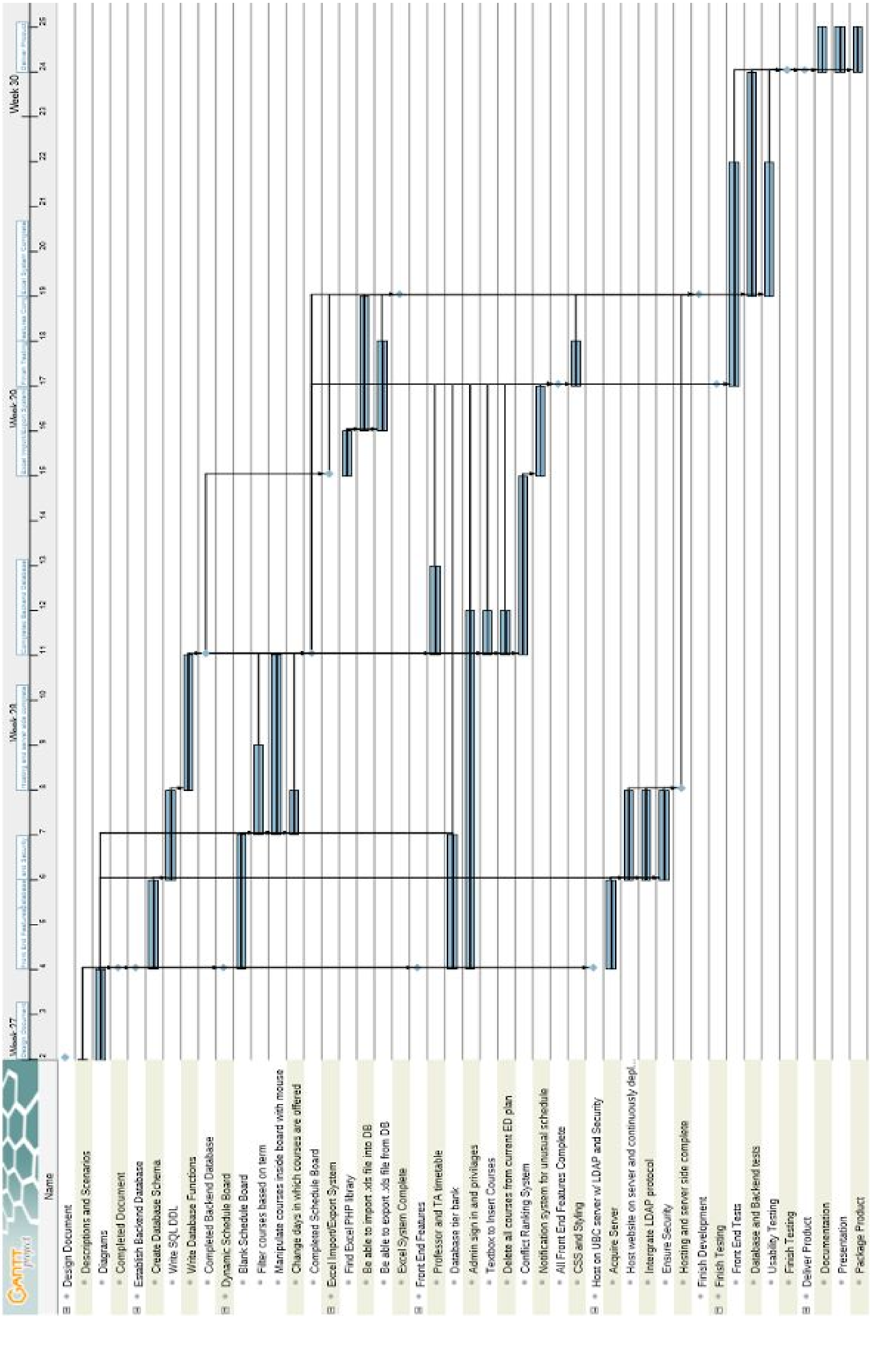
Test plan

The product will be put under a variety of tests, such as functional tests, performance tests, user tests, and acceptance tests. For frontend testing, we have gone through automated tests by using systems such as ghost inspector. For manual testing, we have used codeigniter and PHP storm.

We also have user tests, which is to let other users to There will be black-box style unit testing on methods, such as checking if exporting to Excel does in fact output a valid Excel file. Integration tests will be written to ensure all parts of the product interact correctly together, like whether or not a request from the application returns with the correct data from the database. Both of these are supported by the Codeigniter framework and can be automated to run every change using TravisCI. System testing should also be done to ensure that the application runs properly on every operating system and web browser we can support. Performance testings should be run as needed to ensure queries and server responses are not taking too long. Usability testing will be done to find ways to improve the user experience. To determine if the application meets the client's requirements, the client will be asked to review the product for acceptance testing.

	Planned Types of Testing
Front End Testing	Functional Testing, Performance Testing, Acceptance Testing
Excel System Testing	Usability Testing, Functional Testing
Database Testing	Usability Testing, Functional Testing
Conflict and Notification System	Unit testing, Functional testing,
Server Testing	Functional Testing, Acceptance Testing

Gantt Chart:

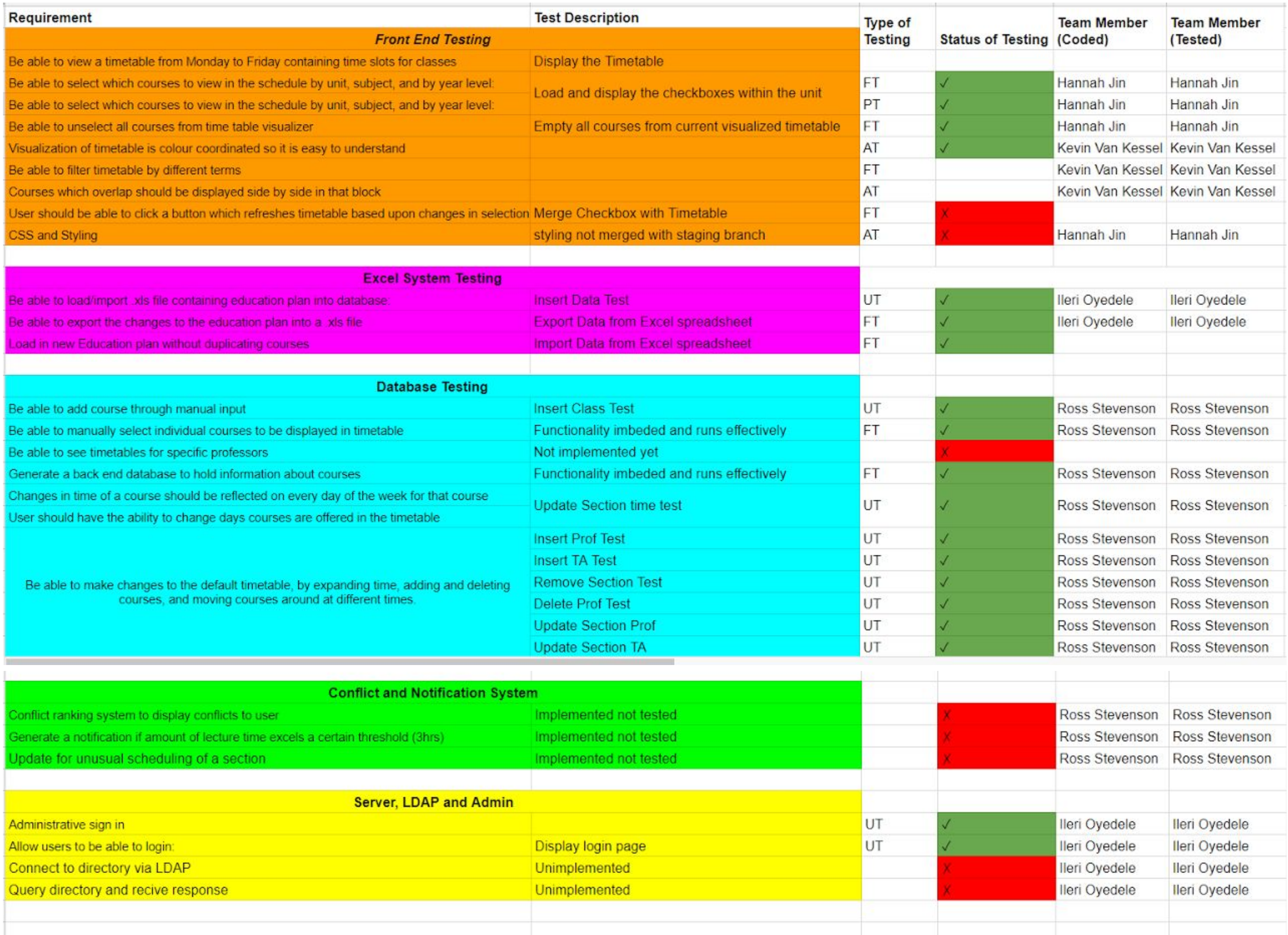


Name

- Design Document
  - Descriptions and Scenarios
  - Diagrams
  - Completed Document
- Establish Backend Database
  - Create Database Schema
  - Write SQL DDL
  - Write Database Functions
  - Completed Backend Database
- Dynamic Schedule Board
  - Blank Schedule Board
  - Filter courses based on term
  - Manipulate courses inside board with mouse
  - Change days in which courses are offered
  - Completed Schedule Board
- Excel Import/Export System
  - Find Excel PHP library
  - Be able to import .xls file into DB
  - Be able to export .xls file from DB
  - Excel System Complete
- Front End Features
  - Professor and TA timetable
  - Database tier bank
  - Admin sign in and privilages
  - Textbox to Insert Courses
  - Delete all courses from current ED plan
  - Conflict Ranking System
  - Notification system for unusual schedule
  - All Front End Features Complete
  - CSS and Styling
- Host on UBC server w/ LDAP and Security
  - Acquire Server
  - Host website on server and continuously depl...
  - Intergrate LDAP protocol
  - Ensure Security
  - Hosting and server side complete
- Finish Development
- Finish Testing
  - Front End Tests
  - Database and Backend tests
  - Usability Testing
  - Finish Testing
- Deliver Product
  - Documentation
  - Presentation
  - Package Product

Software Status:

Test Report:



Requirement	Test Description	Type of Testing	Status of Testing	Team Member (Coded)	Team Member (Tested)
Front End Testing					
Be able to view a timetable from Monday to Friday containing time slots for classes	Display the Timetable				
Be able to select which courses to view in the schedule by unit, subject, and by year level:	Load and display the checkboxes within the unit	FT	✓	Hannah Jin	Hannah Jin
Be able to select which courses to view in the schedule by unit, subject, and by year level:		PT	✓	Hannah Jin	Hannah Jin
Be able to unselect all courses from time table visualizer	Empty all courses from current visualized timetable	FT	✓	Hannah Jin	Hannah Jin
Visualization of timetable is colour coordinated so it is easy to understand		AT	✓	Kevin Van Kessel	Kevin Van Kessel
Be able to filter timetable by different terms		FT		Kevin Van Kessel	Kevin Van Kessel
Courses which overlap should be displayed side by side in that block		AT		Kevin Van Kessel	Kevin Van Kessel
User should be able to click a button which refreshes timetable based upon changes in selection	Merge Checkbox with Timetable	FT	X		
CSS and Styling	styling not merged with staging branch	AT	X	Hannah Jin	Hannah Jin
Excel System Testing					
Be able to load/import .xls file containing education plan into database.	Insert Data Test	UT	✓	Ileri Oyedele	Ileri Oyedele
Be able to export the changes to the education plan into a .xls file	Export Data from Excel spreadsheet	FT	✓	Ileri Oyedele	Ileri Oyedele
Load in new Education plan without duplicating courses	Import Data from Excel spreadsheet	FT	✓		
Database Testing					
Be able to add course through manual input	Insert Class Test	UT	✓	Ross Stevenson	Ross Stevenson
Be able to manually select individual courses to be displayed in timetable	Functionality imbeded and runs effectively	FT	✓	Ross Stevenson	Ross Stevenson
Be able to see timetables for specific professors	Not implemented yet		X		
Generate a back end database to hold information about courses	Functionality imbeded and runs effectively	FT	✓	Ross Stevenson	Ross Stevenson
Changes in time of a course should be reflected on every day of the week for that course	Update Section time test	UT	✓	Ross Stevenson	Ross Stevenson
User should have the ability to change days courses are offered in the timetable					
Be able to make changes to the default timetable, by expanding time, adding and deleting courses, and moving courses around at different times.	Insert Prof Test	UT	✓	Ross Stevenson	Ross Stevenson
	Insert TA Test	UT	✓	Ross Stevenson	Ross Stevenson
	Remove Section Test	UT	✓	Ross Stevenson	Ross Stevenson
	Delete Prof Test	UT	✓	Ross Stevenson	Ross Stevenson
	Update Section Prof	UT	✓	Ross Stevenson	Ross Stevenson
	Update Section TA	UT	✓	Ross Stevenson	Ross Stevenson
Conflict and Notification System					
Conflict ranking system to display conflicts to user	Implemented not tested		X	Ross Stevenson	Ross Stevenson
Generate a notification if amount of lecture time excels a certain threshold (3hrs)	Implemented not tested		X	Ross Stevenson	Ross Stevenson
Update for unusual scheduling of a section	Implemented not tested		X	Ross Stevenson	Ross Stevenson
Server, LDAP and Admin					
Administrative sign in		UT	✓	Ileri Oyedele	Ileri Oyedele
Allow users to be able to login:	Display login page	UT	✓	Ileri Oyedele	Ileri Oyedele
Connect to directory via LDAP	Unimplemented		X	Ileri Oyedele	Ileri Oyedele
Query directory and recive response	Unimplemented		X	Ileri Oyedele	Ileri Oyedele

Usability Test Report:
Survey results were positive in all categories. Listed below are observations and comments on negative aspects of usability testing.

Negatives:
Activity type unclear when manual input
Struggled for doing manual edit
am/pm (12 hour clock on windows was not intuitive) -> little struggle
Interface is a bit small
Time not grabbed for manual input of course when changing (This becomes annoying when inputting multiple times)
Mistook edit selection for manual entry
Not clear how to edit
Have trouble selecting terms (confusing)
selection is limited to one instructor and there doesn't seem to be an option to select particular courses
the import excel feature suggests that users could import a subject or multiple subjects at a time, but not sure how can they then select a group of courses or instructors to populate the map.
the manual entry forces to enter each time slot rather than being able to choose between the blocks (eg. MWF 9:30 am - 10:30 am). (tedious).

To do:
Courses overlap if time goes over other courses (FIXED)
autoload on selection by dropdown is easier to find professor
Manual input didn’t work (FIXED)
Error handling for time error data (FIXED)
Auto completion for teacher names and TA names (Currently it is very easy to enter wrong data)
Activity type? Scroll down when input entered lower than screen (too many hours)
Mon 30 min times (Third:00 doesn’t work?) (FIXED)
Greyed out example values in form (FIXED)
Visible changes to course on hover (FIXED)
Validate form fields
Change form input “Types”
differentiate between Summer and Winter schedule.
there are 3 terms with the label 3 used to indicate a course that run through both term 1 and 2. (true for both Winter and Summer Session.)

Unimplemented Requirements:

LDAP Integration: LDAP complete however unable to merge and test system.

Unit Head Privileges: All valid users have admin privileges. Only one admin login.

Automated testing: unable to implement.

Known Bugs: all known bugs have been fixed.

Delivering the Project:

Where the client can find the code:

Our code is hosted on our UBC server: http://s136.ok.ubc.ca/

The administrative sign in is **Login:** admin **Password:** admin1pass!

Installation details needed:

The site should be fully operational and accessible while on campus at UBC Okanagan or when connected to the ubc remote server at myvpn.ok.ubc.ca. To connect to the remote server follow the guide laid out by IT services at https://it.ubc.ca/services/email-voice-internet/myvpn

Unimplemented features:

Filter courses by term: almost complete. Unable to merge prior to deadline

Project Repository:

https://github.com/UBCO-COSC499-Summer2017/project-6-timetable-project-6-team

All of the layout for our Git repository is from Codeigniter, which is our framework used in this project. Depending on what source code the user may want to see, they are all organized into subfolders. For example, to view the source code of front end such as CSS design, users can go on an application folder -> views -> CSS Layouts. To view the backend source code like database, they are under ddl folder, which is within the "files" folder.

User Guide

i. Login/ Logout
ii. Import/ load .xls file
ii. Make changes to sections by moving around boxes, expanding time, adding and removing courses
iv. Add courses through manual input
v. Add courses through selecting sections in the checkbox databank
vi. Export timetable to excel file
vii. Refresh timetable

Login/ Logout

Purpose:

- gain access to the timetable system

Procedure:

- Type in username and password, and click sign in. The page will be directed to timetable page.

Import/ load excel file

Purpose:

- To add education plan to the database so users can edit sections in the timetable.

Procedure:

- On the right side of the timetable page, click “Import excel” button. Then click “Import” and “choose file”. You will then see the pop up where you can select .xls files. Click .xls file you want to upload, then press “open”. The .xls file is now uploaded and can be seen in the databank.

Make changes to sections by moving around boxes, expanding time, adding and removing courses

Purpose:

- Edit education plan visually so users can create an ideal schedule that can then be exported from our timetable website back to an .xls file.

Procedure:

- Click one of the sections in the timetable. On the right side of the timetable page, the information of the course selected will be displayed. In this area the user will be able to change the time, day, professor, TA and any other information about the section. The user will also be able to delete the section if they wish.

Add courses through manual input

Purpose:

- By being able to add sections through the manual input, users can customize course descriptions, including names, types, terms, and times.

Procedure:

- On right side of timetable page, there is a form called Manual Entry. Users must input valid information they choose, such as subject name, course number, section, activity type, Instructor and TA name, and term. Start and End time can also be controlled, and choose between am and pm. Then select any days to offer the section. Click submit. Then the section that has just been customized will appear in the databank.

Select section through selecting sections in the checkbox databank

Purpose:

- To provide users a organized chart, which allows them to choose and navigate courses easily through a hierarchical checkbox layout, the sections that are selected through this layout will be displayed on the timetable.

Procedure:

- On left side of the timetable page, there is the checkbox databank. Courses are grouped together based on which units they are associated with. In order to select certain sections, specific courses are organized and grouped hierarchically, starting from units to course sections. Check any unit checkboxes, then course names, levels, and sections. Multiple sections can be selected and will all appear in the timetable.

Export timetable to .xls file

Purpose:

- To save a copy of the current education plan as .xls file on the administrator's system.

Procedure:

- On right side of the timetable page, click “Export”. This will convert the edited timetable to an .xls file and will be saved to computer. If the user want to save their work instead of exporting their work to .xls, they may click “Save to DB”. This will save their work on the server for when they login back to the system again, without being converted to an .xls file.