Intro

Our team is planning on creating an online schedule-sharing social network.

This web application will allow friends to share their course schedule with each other, as well as subscribing to their professor and teaching assistants’ office hours. Schedules will be uploaded as an iCalendar file. Once uploaded, the schedule can be edited and friends of the user will see the newly updated information instantly. The goal of the site is so that the user of the website will be able to see all the schedules in one place and won’t have to look for them individually. Hopefully, this app will help eliminate such questions among friends:

To enhance the user’s experience, the list of friends and professors/TAs the user has subscribed to will be pinpointed on an interactive map, in real-time geolocation if the user allows it, which will have precedence over their scheduled time. To represent the individual’s friends, markers will be used and pinpointed on the map to indicate the friend’s location. Additionally, highlighted route from one building to another will be implemented when a friend is clicked on as she is currently walking over to her next class. Along with the Google Map API, we will also use the Google Calendar API to model the uploaded schedule. Doing so will provide a nice visual to display the user’s daily schedule in order to enhance the web experience. Additionally, the Google Plus API can be used to extract the user’s information for a more personal experience.

User Stories, Timeline, and Acceptance Criteria

Functional Requirements:

Tian/Gerry - As a student user, I want to send friend requests/subscribe to other users/events, so that I can see their updates. – Oct 31st

Acceptance criteria:

* Being able to search for friends through their email addresses.
* Ability to add friends based on their gmail addresses.
* Able to see friends in the *Friend* tab.
* Can see friends notification through the *Notification* tab.
* See notification when friends accepted requests.
* User will be able to subscribe/unsubscribe to public services
* Can click on a marker indicating an event and subscribe to it through the info window
* See the event added to the user’s schedule
* Unsubscribe from the event from the calendar tab by deleting it or from the info window attached to the marker

Tian - As a student user, I want to see the status of my friends, for example, how many friends are available now, or how many friends are in a class, etc. – Oct 31st

Acceptance criteria:

* In the friends, there is an icon next to each friend’s profile indicating the personal status.
* If we click onto the icon, there shall be more detailed information about the friend.

Del - As a student user, I would like to download my course schedule from UBC courses and then upload it to the website, so that my friends can see it on a map. – Oct 31st

Acceptance criteria:

* Ability to upload the schedule .ics file into the system and have it automatically parsed into the calendar widget, confirming that the schedule is properly populated.

Del - As a student user, I would like to enter my own personal items on my schedule (i.e. personal study time, do not disturb; or party time, join me!) so that my friends know if I were busy or not. – Nov 17th

Acceptance criteria:

* Being able to add personal arrangements into the schedule, and have all the events and activities properly persisted in the database.
* Being able to remove events from the schedule.
* Friends can see your updated schedule for new events.

Del - As a student user, I would like to see my friends’ course schedules and personal schedules so that I know whether they are free for meet-ups. – Nov 22th

Acceptance criteria:

* Ability to click on a friend and view his/her schedule, but not being able to modify it.

Raymond - As a student user, I want to see my friend’s current up-to-date positions on the map. – Oct 31st

Acceptance criteria:

* Ability to pinpoint friends’ current positions on the map.
* See more details when I click on their locations, such as the time the update took place.
* Only see friends who are on the map, not strangers.
* Ability to confirm that there is a timestamp for the person’s last known geolocation.
* Able to see for friends and current users’ the markers moving as the users relocate and log into the system.

Raymond - As a student user, I want to see the route and estimated travel time from my current position to the destination on my map. – Nov 17th

Acceptance criteria:

* Student users will see a turn-by-turn navigation suggestion from their current location to their destination.
* Student can see the estimated time it will take to get to their next destination/class.
* A route will be highlighted on the map to indicate the user to going to another class.

Raymond - As a student user, I want to have conversations with my friends in real time. – Nov 23rd

Acceptance criteria:

* User will be able to chat with friends in real time with our web application.
* Both user can see the messages sent near instant time
* User can double click on user picture to start a conversation

Gerry - As a student user, I would like to subscribe/unsubscribe to certain public services to get their servicing schedules and event notifications. – Nov 17th

Acceptance criteria:

Gerry - As a general user, I want to keep my privacy to the maximum level, so that only friends but no strangers will get to know my personal info. – Nov 17th

Acceptance criteria:

* User is able to set his/her exposure (private or public) of personal information.
* If the profile is public, everyone can see their current location on the map.
* If the profile is private, only friends of the user can see his/her location.

Gerry – (optional, may not do if not enough time) As a student user, I want to be able to see a list of popular YouTube videos in the nearby area so that I am entertained and come back often to the website to see new information, along with checking out my friend’s schedule. – Nov 23rd

Acceptance criteria:

* User can search for video and see relevant results
* User can click on the video to watch it
* User can like/recommend the video so that other users in the location will be able to see it
* A list will be repopulated as user log in from different locations, or go to different location while logged into the website

Gerry - As a general user, I want to be able to log in with my Google+ account so that I can post about my activities on this web application, such as a plan to attend an employer info session tonight. – Nov 17th

Acceptance criteria:

* Can click on a Google+ log in button and prompt to log in with Google+ credentials
* Able to post on Google+ news feed about events and meetings scheduled in web application

Non-functional Requirements (Everyone) –Ongoing till final project demo

As a general user, I want to the site to be secure and stable so that I know my personal information is safe and that my user experience is enjoyable.

Acceptance criteria:

* Information about the user can only be accessed if the user provides the correct credentials and is currently logged in
* When user logs out, the information cannot be accessed
* UI is smooth and not glitchy or show unexpected behaviour

As a user, I want to be able to the website to be accessible so that I can access it from any browser, on any device that is connected to the internet.

Acceptance criteria:

* User can access the website from desktop or mobile device with Safari, Firefox, Chrome, IE

As a developer, I want the code to be maintainable so that it is modular and can easily be changed and understood by other developers.

Acceptance criteria:

* Classes, methods, and variables should be commented using Javadoc standards
* Code style should be agreed upon and followed through
* Appropriate architectural styles should be used when possible
* Appropriate design patterns should be used when possible

As a user, I want the user experience to be fast so that I can see all my friends’ location on the map in real-time and not see delay in the activities I perform on the website.

Acceptance criteria:

* The website is scalable to when a lot of users are logged in and using our website concurrently
* The performance of the map showing the friends’ geolocation in near real-time is good

Web Services in our Application

Google Map Web Service

To enhance the user’s experience, we will be using Google Maps to show the approximate geolocations of a user’s list of friends in real-time, or according to their scheduled time if the geolocation is not available. To represent the individual’s friends, markers will be used and pinpointed on the map to indicate the friend’s location. Additionally, highlighted route from one building to another will be implemented when a friend is clicked on as she is currently walking over to her next class.

YouTube Web Service (optional, may only do Google+ and Map Web Service)

Furthermore, to provide an interesting and entertaining dynamic environment in our application, a list of YouTube videos watched or shared in the nearby area will be auto populated as the user go from one place to another. This way, users can be introduced to various videos that may be related to a certain area on the map, and tie in with the social network aspect of sharing information. As well, doing so may persuade the user to come back often, and log in from different locations to see what type of videos may show up.

Google+ Web Service

In order to utilize existing social networking tools, we can integrate the Google+ web service into our application. Doing so can provide the user with the flexibility to use their existing account to log into our web app, as well as sharing content from our site easily with their existing social network. Such contents can include sharing a newly created lunch meeting with a friend or an employee info event they are currently attending.

Who Doing What

Tian Cai (Own profile modifications/tasks and friend requests)

Searching and sending out friend requests, as well as accepting them

Ability to upload a user picture attached to profile

Displaying list of friends and friend tab related tasks

Display details about friends, such as their current status, location, and picture

Ensure slick UI

Gerry Cao (Notifications and Web Service Integration)

Settings tab to allow user to select a public or private profile

Ability to subscribe and unsubscribe from friends

Immediate notifications from friend updates

Dynamically populating a list of YouTube videos

Share/like/contribute a YouTube video that will be added to the list and shared with nearby users

Data is never stale and list repopulated upon new data

Google+ plus login and posting functionality

Raymond Lei (Instant messaging and map tasks)

Messaging and instant messaging functionality

Map overlays and directions marker/time to next class

Populating map with friend locations in a feasible way

Map the user’s location in near real-time efficiently

Zeyu Zhong (Calendar and schedules)

Parsing .ics schedule file and populating the calendar view with the correct events

Allow user to modify their schedule in the calendar view

Be able to see friend’s schedules when friends are selected

Everyone

Implement functional tests, such as JUnit test, as you code

Run the tests and make sure they consistently pass

Make sure feature passes acceptance criteria

Continually check code does not contain any of the top 25 security concerns (<http://www.sans.org/top25-software-errors/#cat1>)

Add comment and use nice code style as you code

No concurrency bugs experienced

Data Storage and Location-Awareness

Data Storage

We will be using the App Engine Datastore on Google App Engine as our data storage. Furthermore, will be using Java Data Objects (JDO) as the standard interface to store objects with data. Afterwards, by using a PersistenceManager, we can interact with JDOs and perform common tasks such as adding, updating, deleting and querying database objects that meet a set of criteria. Each service will access/modify the data storage via asynchronous callbacks in order to retrieve information from the client side. All the heavy process and logic will occur in the server side, while the client side only provides parameters and display what is retrieved most of the time. Doing so will improve our web app and computation performances.

Location-Awareness

We will be using the GWT geolocation API that accesses the HTML5 geolocation API in order to implement location-awareness and near real-time geolocation in our web app. This solution is compatible with many browsers, such as Firefox, Chrome, Safari and Internet Explorer. Hence, providing good accessibility to our website. Furthermore, to obtain the user’s real-time location, an async call every 1 second to getCurrentPosition() will return latitude and longitude values of the logged in user. With the information, we can correspondingly update the overlay marker on the map if the user’s location changes. Additionally, this will also be done for any of the currently logged in friends of the user. As a result, the map will also update the user’s friends location in real-time. This will be a concern and needs to be done in an efficient way.

Sprint 1 – October 31st, 2012 (Basic UI)

Sprint 2 – November 17th, 2012 (Each Function works by itself)

Sprint 3 (Code-freeze) – November 23rd, 2012 (Functions work interactively)

Bug fixes and final touch-ups – November 24th to 26th, 2012