Campusnaut

ECE 1778 Final Report
12/04/12
Word Count: 2499

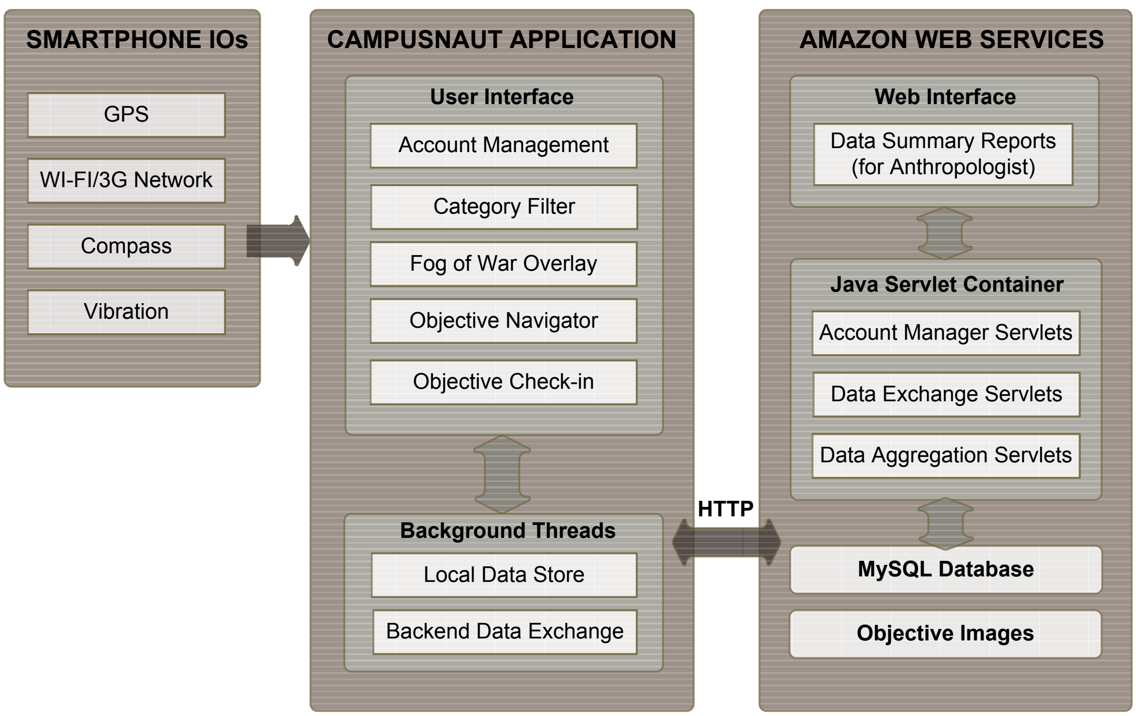
Steve Chun-Hao Hu (992566996)
Chenliang Man (999418170)
Graham Candy (990856356)

Introduction

The motivation behind *Campusnaut* was to design an interesting Android application for new students to explore the University of Toronto campus that simultaneously generates real-time demographic data for investigation and interpretation by social scientists. More specifically, this demographic data correlates students' gender to their particular areas of interest on campus. Social scientists may use this data to make tentative propositions regarding new students' gendered interest in particular areas of the campus, providing new perspectives on why university enrollment rates are increasingly imbalanced, currently at a 60/40 female to male ratio.

Overall Design

The following block diagram outlines the major components and interactions that make up the Campusnaut application.



Our app relies on four key components of smart phones, including the GPS function, WIFI/3G connectivity, the compass and the vibration function.

Campusnaut Application

Account Management

The account management component includes the login screen and account creation screen. When user creates a new account, the account information is uploaded to MySQL database and the login credential is saved using SharedPreferences. After first login, the app will automatically sync user's profile from the SharedPreferences and the Category filter component will be loaded directly.

Category Filter

The category filter interacts with the backend database to sync a list of objectives data from MySQL database into SQLite database. The filter employs inflater to generate a two level dropdown list where 1st level is sorted into 5 general categories, and 2nd level is a list of 27 locations. Objectives are synced with the backend whenever a user checks-in.

Fog of War Overlay

The fog of war effect utilizes the CPU-based *canvas* function, which is normally used by drawing/paint applications. Essentially, we created a second, black layer on top of Google Maps. Once the user travels over 0.0004 GPS coordinate distance, the layer will be triggered to draw a transparent circle with blurred edge to on the canvas to reveal the map underneath.

Objective Navigator

The user is represented by a cartoon beaver[1], located at the center of the screen, positioned by GPS through 3G/WIFI connectivity. We use matrix to rotate the icon based on the angle between the user and nearest objective. The distance between the user and each objective is continually calculated. The hand of the beaver functions as a compass. When the user is within 50 meters of an objective the phone will vibrate to alert them that they are within 'check-in' distance.

Objective Check-in

[1] A beaver is the mascot of the University of Toronto and is prominently displayed on its coat of arms.

The check-in component loads the objective image stored on Amazon S3 and loads the objective description and check-in status from SQLite. After check-in, the check in button will be disabled; the check-in status will be uploaded to MySQL database and updated on the SQLite database.

Local Data Store

The SQLite database caches the objective information such as GPS coordinate, detailed description, image ID on Amazon S3, and current user's check-in status. The SharedPreferences is used to cache the user login credential and status. The internal file I/O is used to keep track of a list of the circle center being drawn on the canvas inside the fog of war layer.

Backend Data Exchange

Several background threads are used to communicate with the backend server through HTTP request in order to upload and retrieve both user data and objectives data to and from MySQL database.

Amazon Web Services

The backend is build on top of the Amazon Web Services free usage tier to provide better scalability if the project is expanded in the future. A remotable Amazon EC2 instance is created to host the Tomcat server and MySQL database. An Amazon S3 bucket is used to store all the objectives' images.

Java Servlet Container

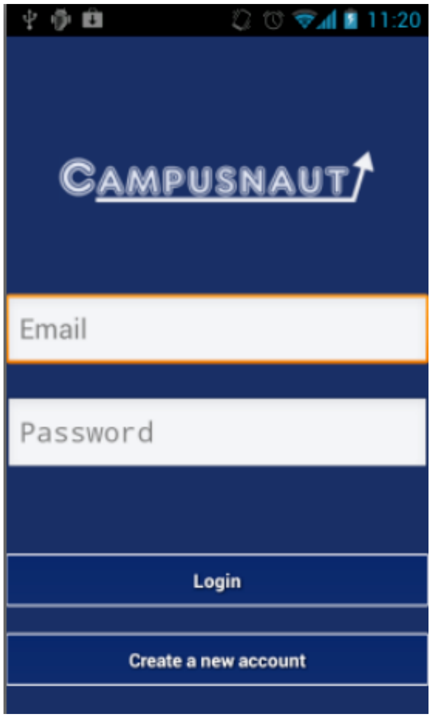
Several Java servlets were implemented on top of the Tomcat server to listen to the data exchange request from the Campusnaut application. The servlets use JDBC to read data from MySQL database and perform data aggregation to provide data feed to the JavaScript RGraph to generate Data Summary Reports.

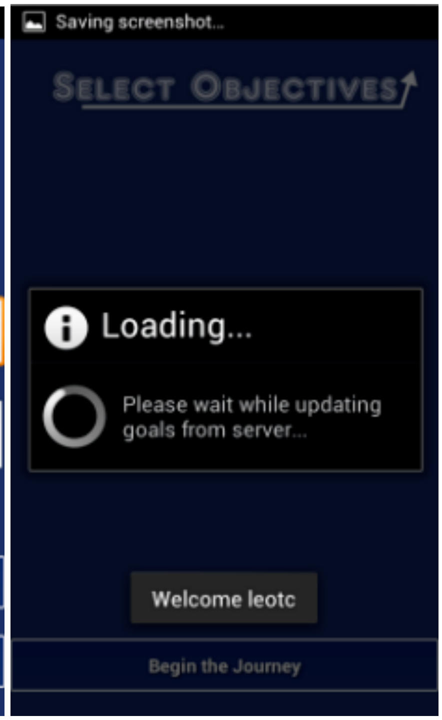
MySQL Database

All the detailed user account information and objective information are stored in the MySQL database.

Statement of Functionality & Screen Shots from App

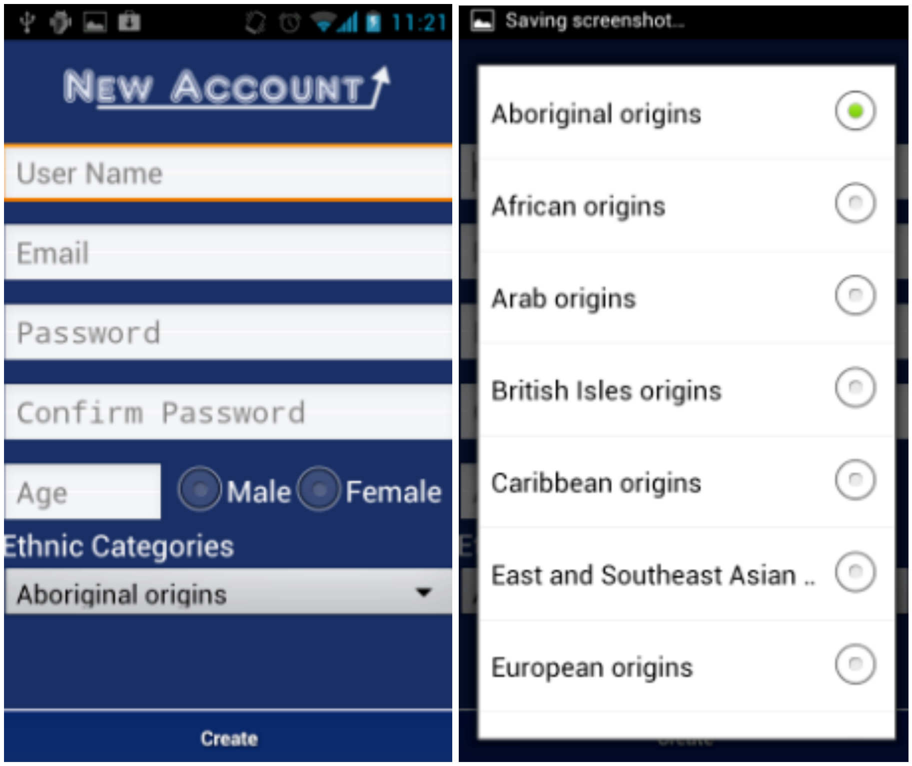
Account Management





Account Creation and Validation

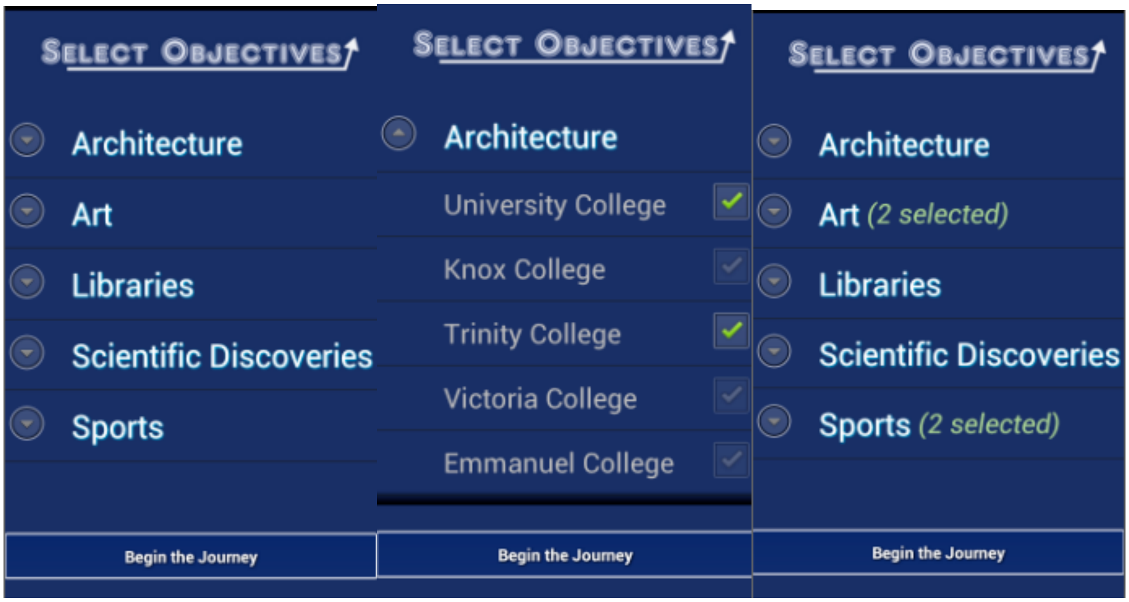
When CampusNaut is first launched, the user must register their account information, including e-mail, password, age, gender and ethnic group. If a user has created an account already they can enter their details and the account will be validated on the server.



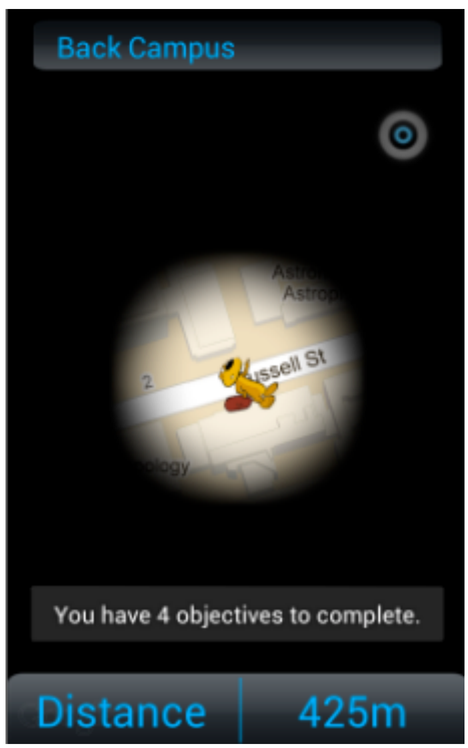
Category Filter

Objective selection

After login, the category filter screen is displayed, generated from the server in real-time. The user clicks on a category to expand a list of objectives. Multiple objectives can be selected from multiple lists. The number of selected objectives will be shown after selection. To 'begin the journey', user should choose at least one objective whereby the selected objectives will be loaded into a current exploration list.



Map View



Fog of war

An overlay is added to black out Google map and this creates a "fog of war" effect which is peeled back as the user moves in the real world at a range of 60 meters in all directions.

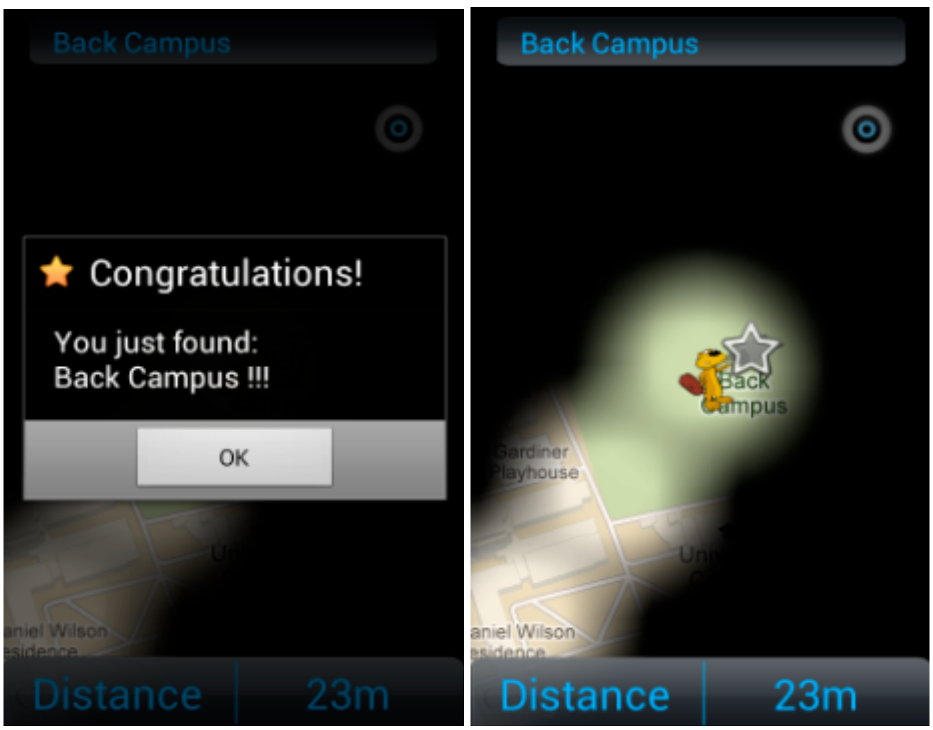
Re-center

On the top right of the map view, a button is provided to re-center the user location. This button is valuable if the user is taking a meta-view of the map and wants to quickly return to the street level view.

GPS location

The user is located on the map based on the GPS network.

Objective Navigator



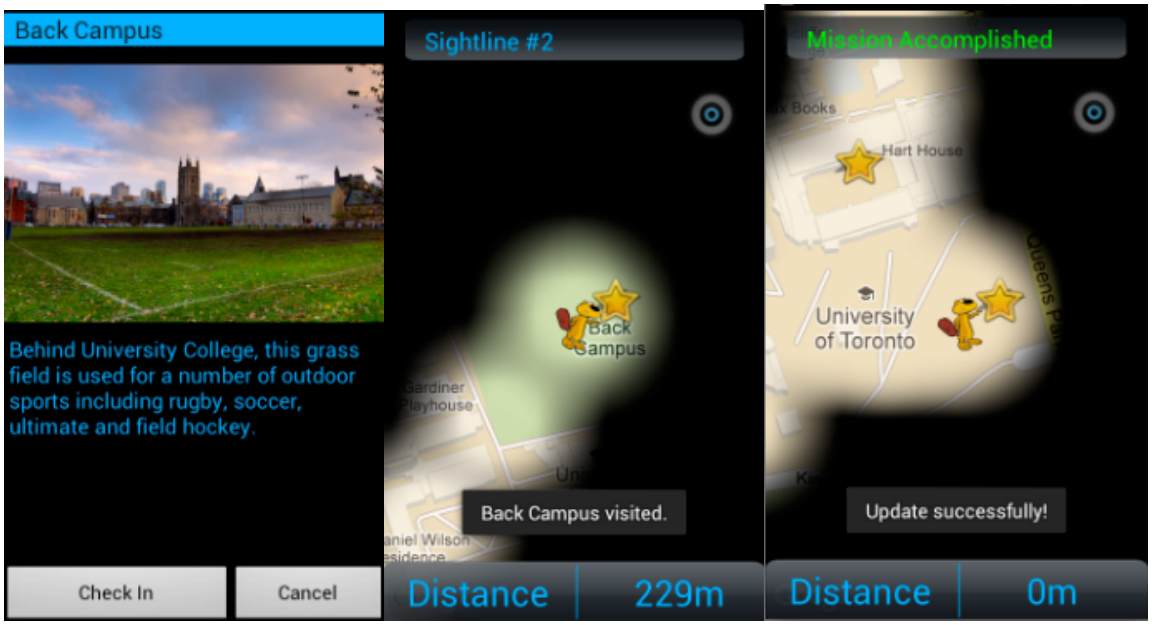
Objective navigation

The objective navigator is activated when user starts the map view. The nearest unchecked-in objective is set as the current objective. The bottom bar displays the distance to the objective.

Objective detection

When the user is within 50 meters of an objective the objective is marked by a silver star on the map, the phone will also vibrate.

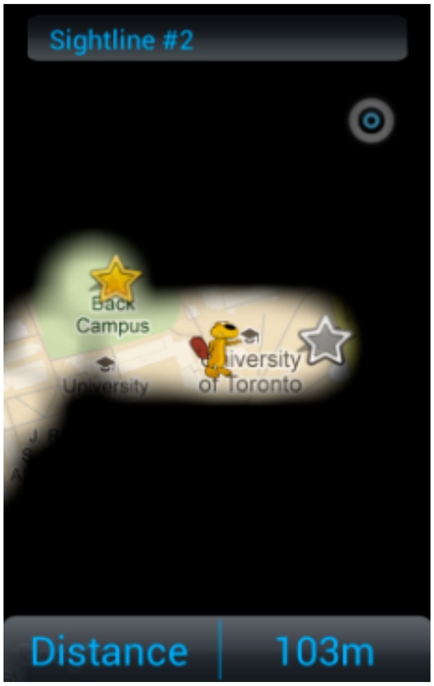
Objective Check-in



Check-in and Mission Completion

When an objective is discovered the user is within 50 meters, it can be checked-into by tapping the silver start and clicking the 'Check In' button. A notification message will then be shown to user and the star will turn gold. Once checked-in, the objective navigator & detector is reset to the nearest, next objective. After all the objectives in are checked-in, the top bar in map view will notify user by showing bright 'mission accomplished' message. In this case, user can continue the mission by adding more objectives in 'objective editor'.

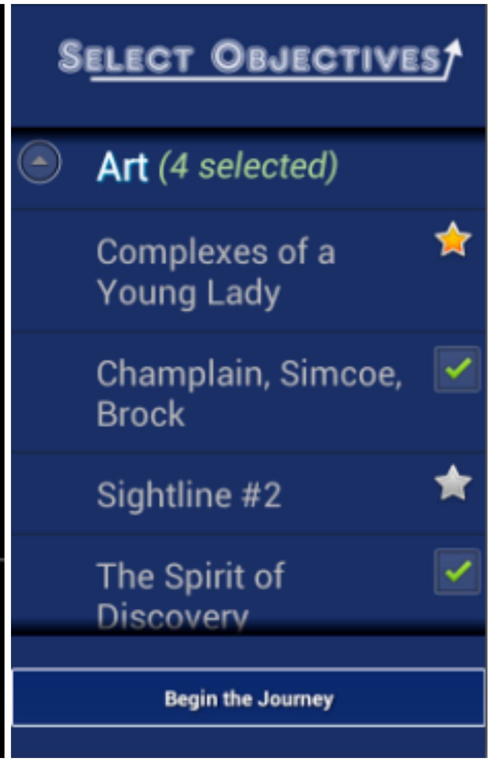
Objective review



Review

By zooming out and tapping on any of stars shown on the map, the user can review the text and image descriptions of discovered objectives. The Check-in functionality is disabled in this situation.

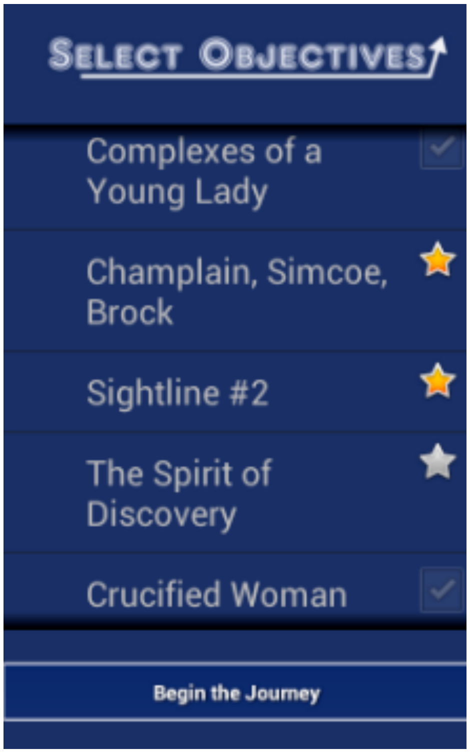
Objective Editor

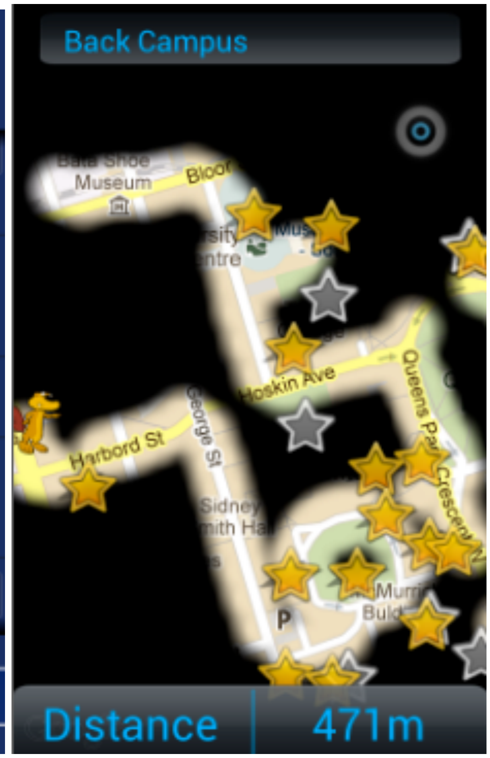


Add more objectives

The 'objectives editor' is launched from the menu. Here the user is able to review their discovered (silver start) and checked-in (gold star) objectives as well as add new objectives.

Exit & Continue

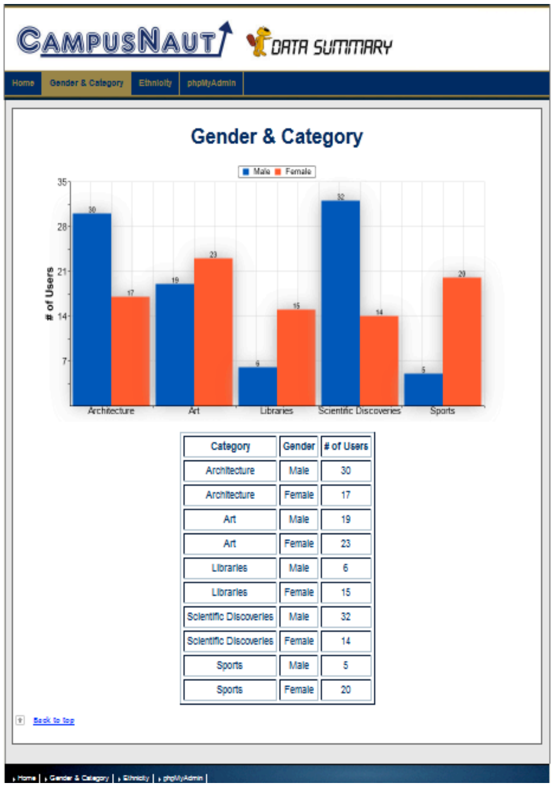


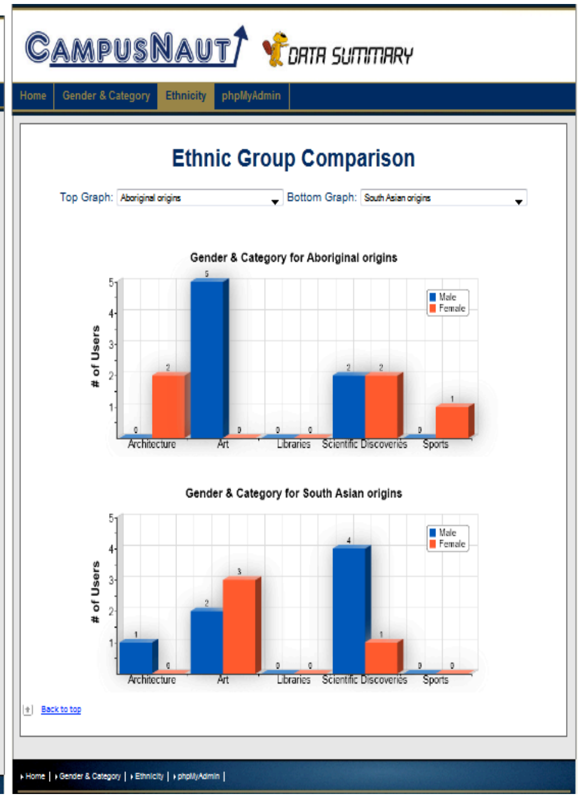


Continue exploration

If the user exits the application and loads it later, the entirety of the user's previous state is saved/loaded.

Backend Website





Data analysis

When a user uploads new objectives into their journey(s), the demographic data originally selected by the user is uploaded to a back end server. For social science researchers we have created a backend website which allows the scientists to observe real-time, graphed representations of this data.

What did you learn – what would you do differently?

We learned to use most of features supported by Google API, however while the API is very efficient, there are some limitations including the API forbidding caching the map offline and the overriding the default tile-based canvas. We should consider using an offline map such as OpenStreetMap API. We might have also considered the use of OpenGL which supports hardware acceleration to increase the efficiency of graphics rendering. The 'fog of war' layer is taxing on the Smartphone CPU after more than 50% of the campus map is uncovered.

Lastly, in terms of key learning, we also believe that the group could have better managed the process of inter-disciplinary collaboration. Rather than assuming we could just start 'collaborating' naturally, we would begin with foundations first. For example, although the group lesson where the apper learned technical aspects of electrical engineering and computer programming was an educational experience, we believe that we should have began with the most fundamental assumptions, techniques and terminology of each of our disciplines, and then built on those fundamentals up to a more complex form of collaboration. Regardless of early difficulties, however, we found this to be a rich interdisciplinary experience and we are happy to have developed this application together.

Contribution by Group Members

Steve Chun-Hao Hu

Key contributions from Steve include implementing the fog of war effect and the background calculations for objective navigator to draw the beaver to point at the nearest goal. He created the UI and local data store functionalities for both account management, and category filter. He also contributed in check-in screen in loading images and descriptions of objectives. Finally, he created the test harness, Mock GPS Controller for testing and demoing purposes. On the backend side, Steve setup the Amazon EC2 instance to run the MySQL database, and Tomcat server. He contributed in programming the servlets to handle data exchange requests. He wrote SQL queries and created the data model, which serve as data source for generating summary reports. Finally, he designed the CSS for Campusnaut website.

Chenliang Man

Key contributions by Chenliang Man included implementing the UI of the map screen, check-in screen, check-in mechanism and their interactive functionality. He overrode the default class OverlayItem to allow app to load different markers onto the same overlay. He created the objective navigator in distance detection, message alert and vibration feature. Chenliang also programmed functions that exchange data with back-end server includes request for account creation, user login validation, user progress update and check-in request. On the back-end server he implemented the servlets which handle different type of user requests, created the scheme of the Campusnaut database and finished the ethnicity summary report page which generate real-time data with graph using Ajax.

Graham Candy

Key contributions from Graham include both background and primary app development. On the background side, he designed the laid out the overall scope of the app and facilitated the evolution of the application towards its final product, moving from a app that was trying to be a game, to an app that served a functional purpose for University of Toronto students while also generating important demographic data for social scientists. Graham determined the relevant demographic information required for interpretation in backend analysis based on core demographic categories for anthropological analysis. Graham then seeded the application with the 27 locations (objectives) on campus. He determined the GPS coordinates by retrieving the data from Google Maps, acquired an appropriate image and wrote or obtained a description of the 27 objectives. These 27 objectives were then sorted into categories (Art, Architecture, Scientific Discoveries, Sports and Libraries) given a sequential goal_id number by Graham. Finally, Graham conducted a mock analysis of the data which was presented in the final presentation.

Apper Context

As the social scientific disciplines are typically centered on gathering qualitative data, it has been a difficult challenge to envisage the *dynamic* use of smart phones for gathering this data, aside from their obvious utility as mobile versions of already existing devices, e.g. as microphone, video recorder, GPS etc. rather than develop this kind of app then, we attempted to create an app that would generate data not typically gathered by simple recording devices. Originally, I proposed that we generate 'movement maps' for analyzing what anthropologists call the "microgeographies" of daily life. In order to analyze this data our application was originally conceived to produce maps of user's GPS location over time, based on a free-exploration kind of model. Over time however, the scope of our project narrowed to a more specific question: "why does the sex ratio of new university students favour female students?" To answer this question it became apparent to me that our broad question of microgeographies placed too much strain on *compelling* users to '*freely explore'* the University of Toronto space, a contradiction which was aptly noted by the teaching assistants for the class.

Today, our application is focused on generating demographic data (gender, ethnicity, age) and correlating this to users' selected interest. This data set aids in theorizing about broad social-cultural conditions which may be discouraging men from attending university by providing a unique lens into understanding some of the potential factors behind this shift. As explained briefly in our final presentation, a recent study of Quebec university students, suggests one possible explanation. This study posits that male children growing up are typically raised to focus their energy on talents that are more associated with maleness, like competitive sports and technical skills, but that they tend to believe that success in these skills is something achievable outside of university, just by relying on their natural capacities as men. The data generated from our application could help challenge or support this argument. For example, if we found that a large amount of women were visiting multiple categories, while men excluded interest in sports but continually sought out scientific discoveries, this may reinforce the social dynamic suggested by the Quebec report. Using this analysis we could argue that a way of attracting more male students to university is, rather than representing universities as sites general economic/social betterment, they should be re-represented to male students as sites with historical significance and venues which have a long history of producing greatness/discovery and prestige. In the end I stress that these are all *hypothetical* arguments that could be challenged/supported using data from this app, and that they do not reflect my personal opinion or analysis.

As a final word, I argue that the discipline of Anthropology remains to be challenged to find dynamic ways to integrate new technologies into their research methodologies.

Future Work

First steps towards future work would include adding more categories and more objectives. There are a large number of sites at the University of Toronto that would not fall into any of the current categories including pubs, cafeterias and residences.
Also, we would add the input validation on Create Account screen to minimize the chance that user creating account with useless information. There are also a number of possible advanced features for future versions. These features could include a fully-fleshed out back end website for users where they are able to view the objectives they have completed across the entire map of the University of Toronto campus, and if the app was extend to the urban centre of Toronto, the entire city. Finally, it remains to be considered regarding the creation of an achievement system for users, a ladder system which allows users to compare their exploration may increase app uptake.

Business School

At present, we are not interested in having a business school team investigate the market potential of the app.