



Final Report
ECE1778h: Creative Applications of Mobile Devices
University of Toronto

Presented to:
Dr. Jonathan Rose

Presented by:
Abbas Attarwala
Abhishek Das
D'André Wilson

1.0) Introduction: What and Why?

Whether you are starting a business, starting a new strategy in your business or simply trying to keep your business viable, you need to do market research. Proper market research is the key between the success and failure of any business, but it can be quite tedious. How do you get users and customers to give feedback on your business ideas and strategies? In the past, market research companies would mail page long surveys to customer's homes to be filled out and mailed back. Some companies also ask users to fill out forms in-store, and nowadays, many use emails and online surveys to collect data. Many of these companies have raffle-like incentives, however the chances of users actually gaining The idea of this project is to leverage the increasing popularity of the Smartphone and create a mobile application that can collect market research from users right on their phone. Users are surveyed based on location and topic interests, questions appear directly to the user and they are given rewards or points/coupons as soon as the question is answered. This mobile app will benefit market researchers by providing an easy way to distribute surveys, while also being able to collect trends in market data as well as provide information on what motivates consumers to answer surveys.

2.0) Overall Design

As seen in figure 5, the architecture of this application is essentially divided into two main components i.e. client and server. The client side consists of the Android and the Web user. The Android user submits and answers market research questions and the web user creates questions and queries on market research. HTTP GET request/response is mode of communication between the client and the server (Google App Engine).

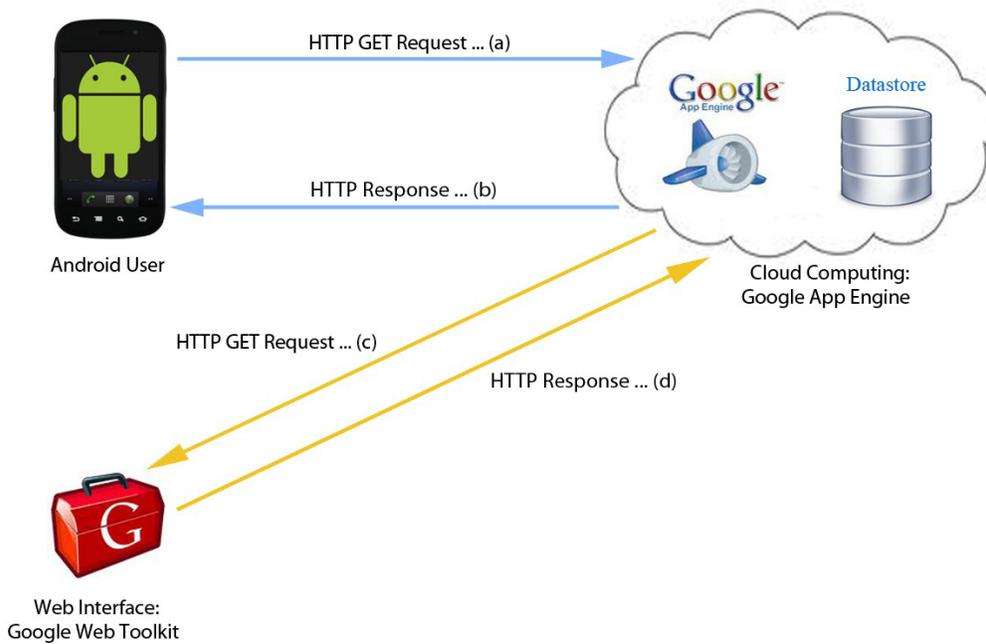


Figure 1: Top-level architecture diagram and interaction between client and server.

More explicitly, the communication between the Android client and Google App Engine server i.e. (a) and (b) from figure 1 (blue arrows) is shown in figure 2.

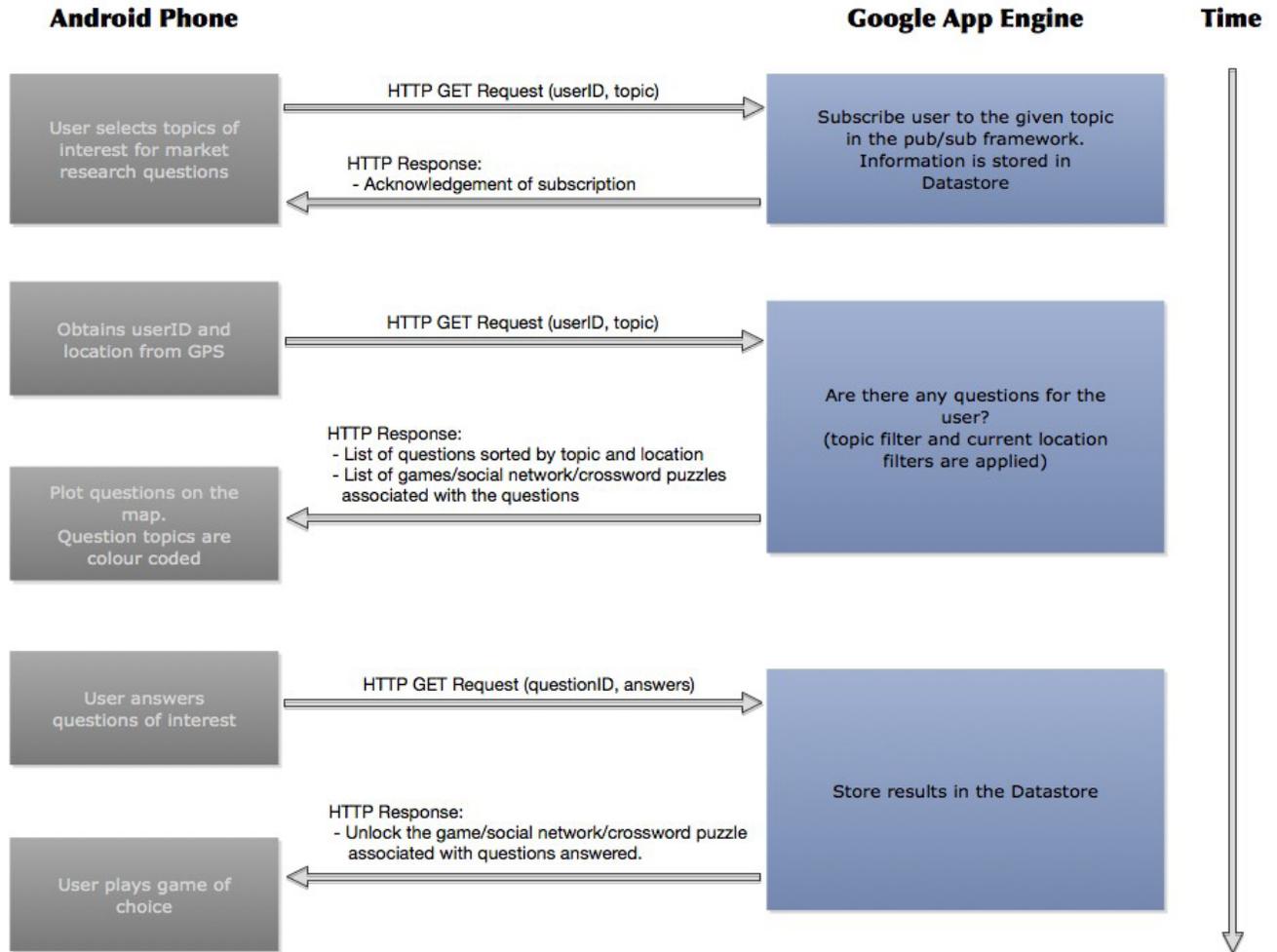


Figure 2: Sequence Diagram of Android and Google App Engine interaction. Time moves down. Each block represents a function i.e. red block for Android and blue block for Google App Engine.

The communication between the web interface client and Google App Engine server i.e. (c) and (d) from figure 1 is shown in detail in figure 3.

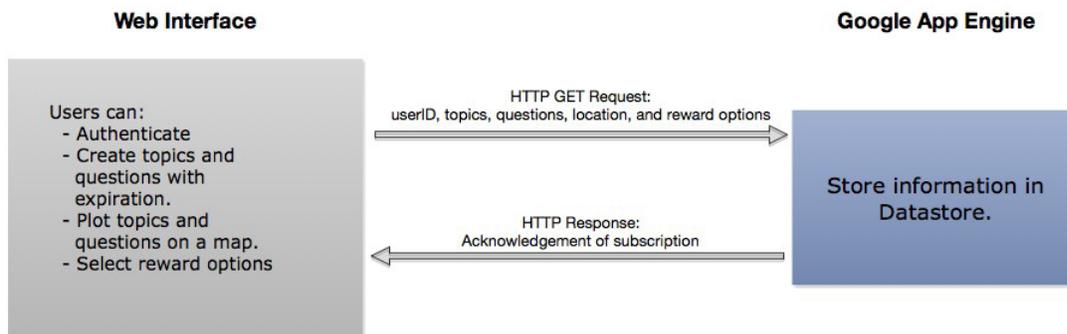


Figure 3: Sequence Diagram of Google Web toolkit (web interface) and Google App Engine interaction. Time moves down. Each block represents a function i.e. red block for web interface and blue block for Google App Engine.

Our architecture implements two major frameworks in the design. 1) Publish/Subscribe architecture and 2) Model View Controller design.

Publish/Subscribe Architecture: The pub/sub architecture that is deployed on the Google App Engine allows Android users to subscribe to topics of their interest. This subscription is important because it filters away questions that are of no interest. The market researcher is publisher that creates questions and assigns topics to them. The system automatically pushes question to the node that contains list of subscribers for that topic.

Model View Controller Architecture: The model view architecture implemented on the web UI and deployed on the Google App Engine allows a coherent dataset (model) to maintain multiple different views. As seen later in Section 3 of the report, the four different views supported are 1) Creation of markers 2) Information Visualization on Map 3) Information Visualization using pie chart and 4) Information Visualization using bar chart.

3.0) Statement of Functionality and Screenshots from App

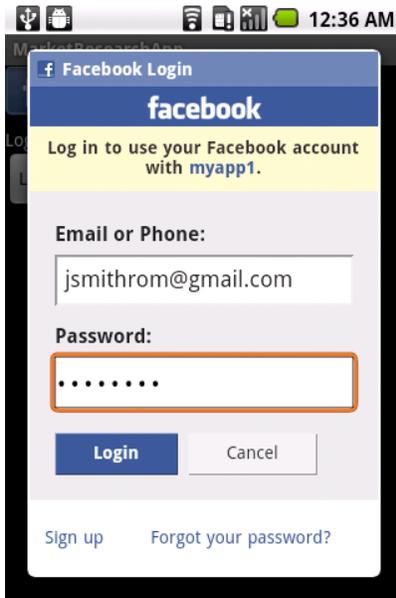


Figure 4: Facebook login page on Android. We use Facebook login to authenticate users on our Android application.



Figure 5: After authentication, the user is presented with an easy to use home screen. The icon on (1,1) is for viewing questions on the map. Icon on (1,2) shows questions in a list view form. Icon on (2,1) allows users to subscribe to topics of their interests such as (food/drink and games). Icon on (2,2) allows user to view their incentive for answering questions. Icon on (3,1) allows for synchronization between client and server. Icon on (3,2) allows for QR code scanning.



Figure 6: The map view displays questions using different marker icon (food/drink and games). Using different icons for segregation across question topics, it becomes easy and intuitive for users to answer them.

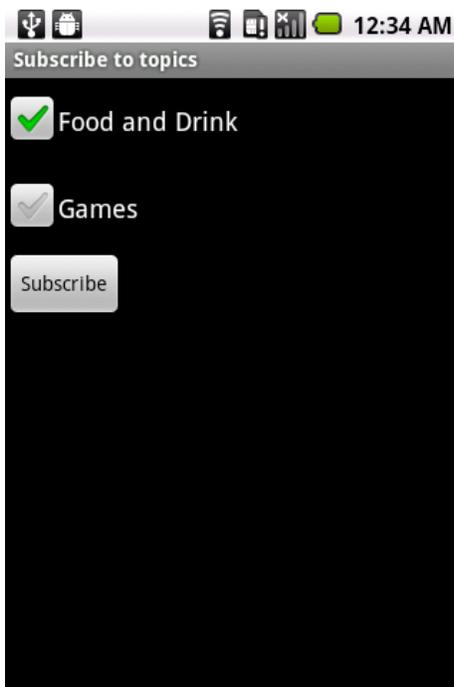


Figure 7: The publish/subscribe architecture on the client side (subscribe), allows users to subscribe to topics of their interest. Here the user is subscribing to Food/Drink topic.



Figure 8: After subscription, questions of food and drink are only displayed. This can be compared with figure 3, where markers of all topics are shown.



Figure 9: Clicking on a marker, opens an easy to use dialog box allowing the user to view the question and answer appropriately.

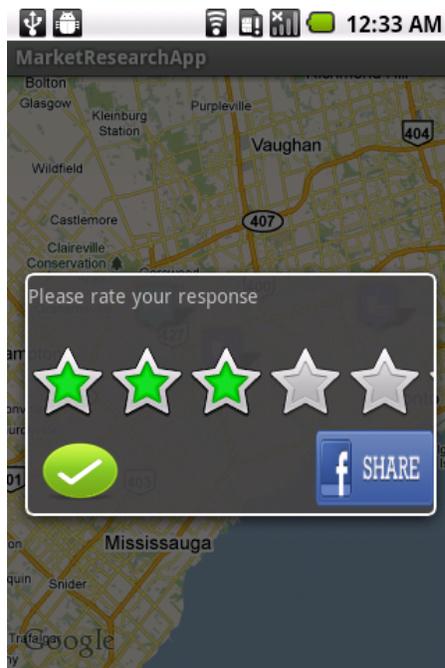


Figure 10: The user can rate the question and post the answer on his/her Facebook wall. By posting it on Facebook wall, we can leverage other interested members from social network to join and start using our Market research framework.

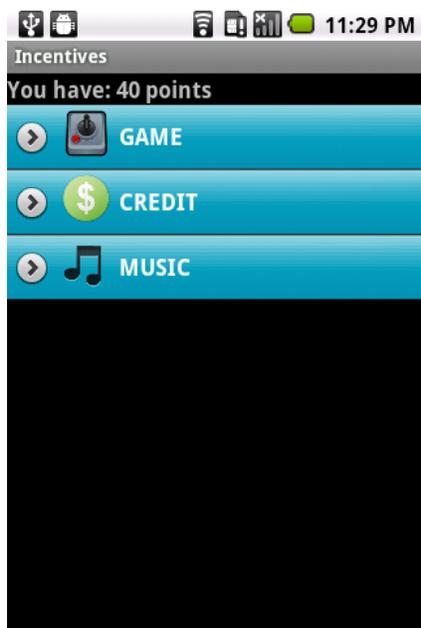


Figure 11: At this time, we support three incentive categories. 1) Games, 2) Credit for Amazon and 3) Music. These incentives are dependent on the number of points accrued on answering questions. As the number of points increases, incentives in the three categories are unlocked.

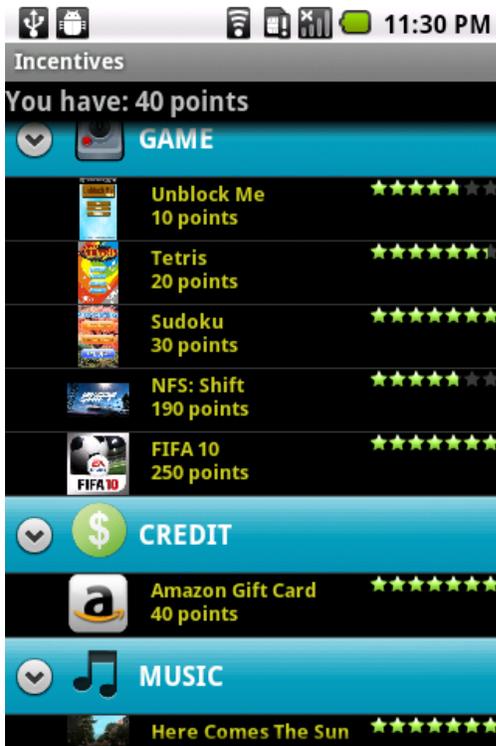


Figure 12: Shows the different games, and music supported for incentive.

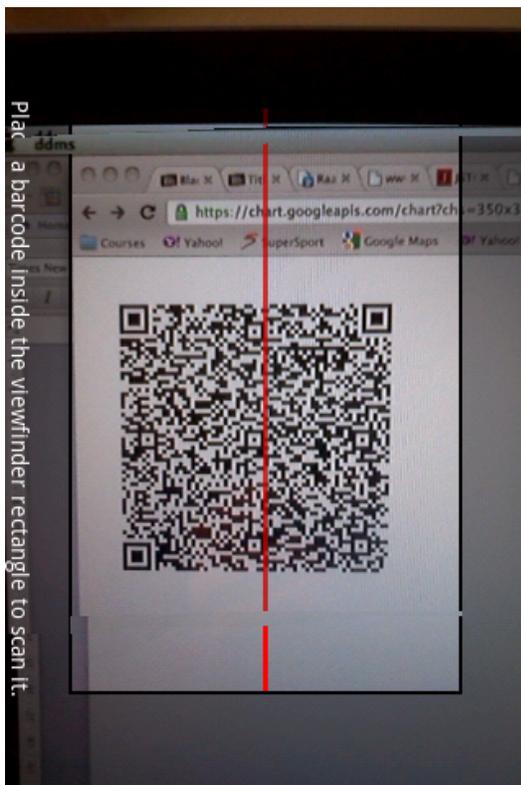


Figure 13: The android user can also scan a QR code and get the appropriate question on the phone for answering.

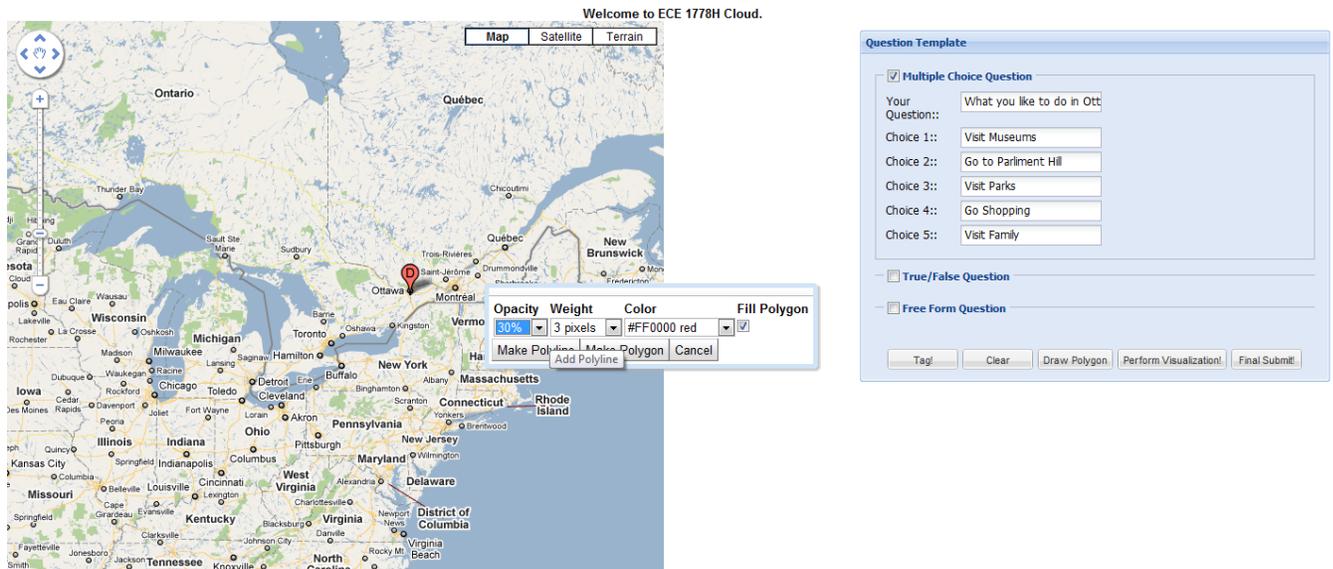


Figure 14: Question can be created by clicking on the map and a marker (market research question) is inserted there. The question template box on the right allows the user to fill in attributes for the marker such as multiple choice question, true/false question or free form question. The 'Draw Polygon' button allows the user to draw a polygon around the marker

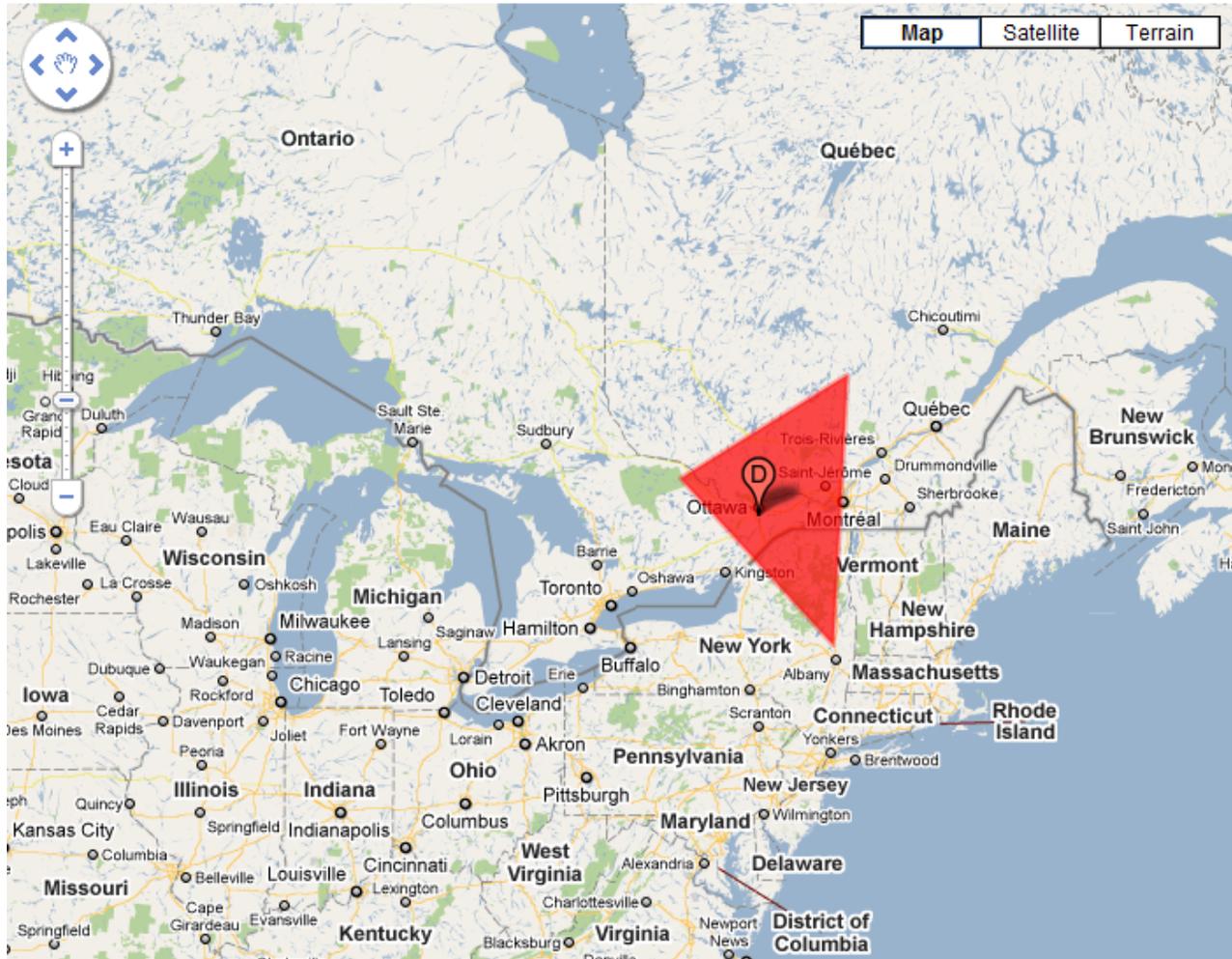


Figure 15: The polygon sets the bounds around the question for the Android user. Only when the Android user falls within the bounds, the question is sent from the server to the client(Android user).

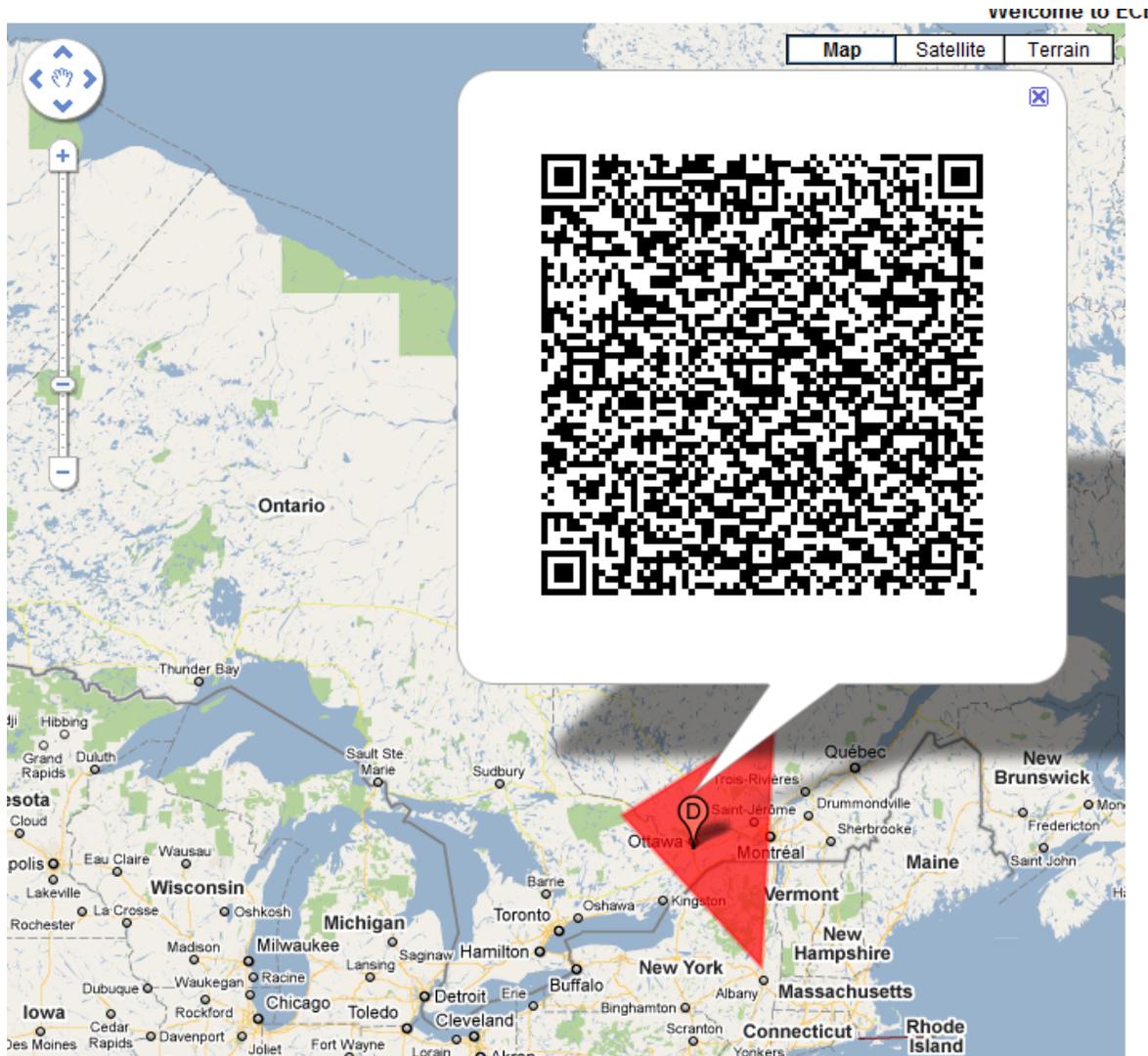


Figure 16: QR code is generated for each question entered by the user. QR code allows the market researcher to post flyers/posters that Android users can scan and answer the question.

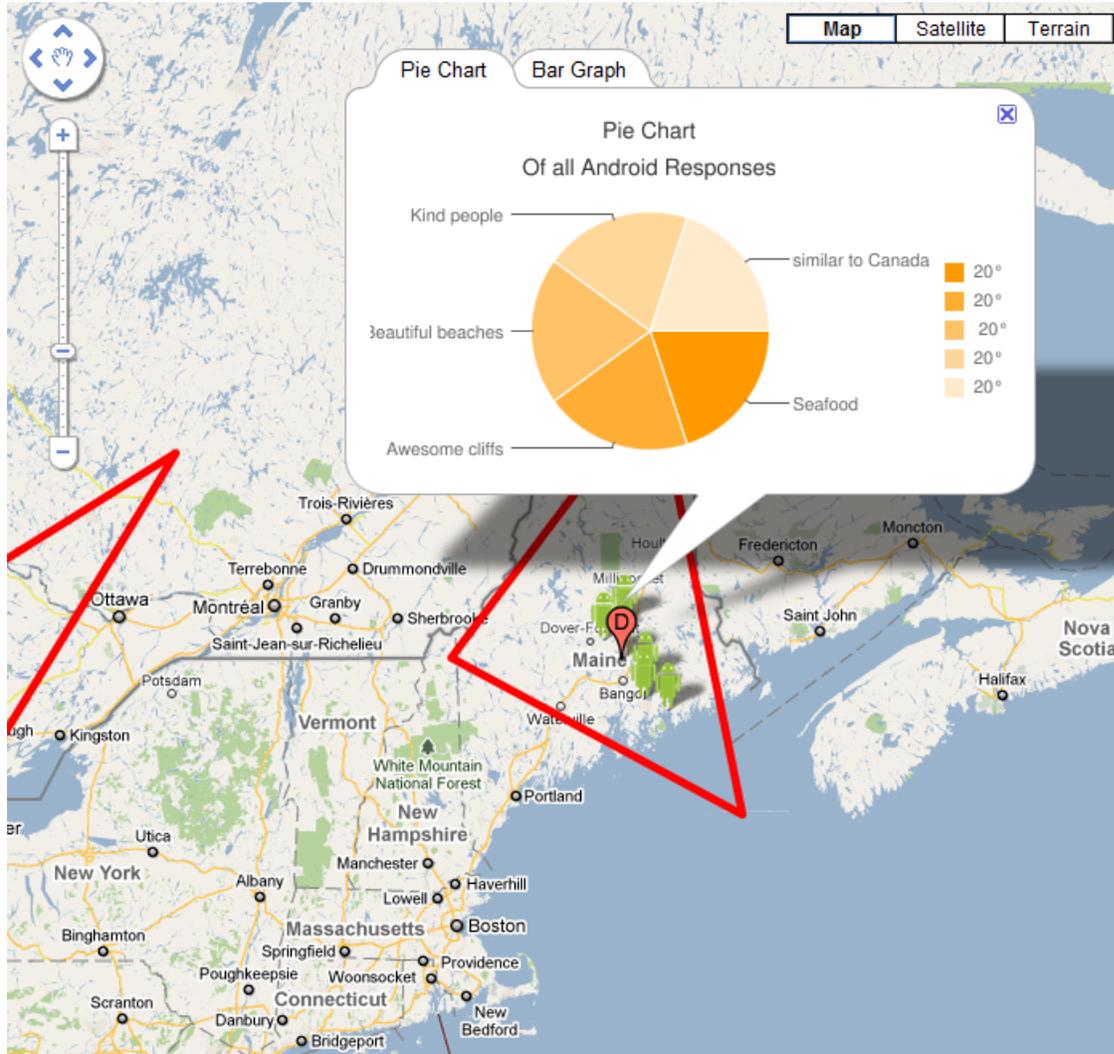


Figure 17: We provide the market researchers with two (Pie chart, Bar graph) information visualization tools on the raw data aggregated from all Android users. In this figure, a pie chart is shown with the distribution of the multiple choice answers. This aggregation of data is visible when the question (red marker) is clicked. The green markers on the map are individual Android responses.

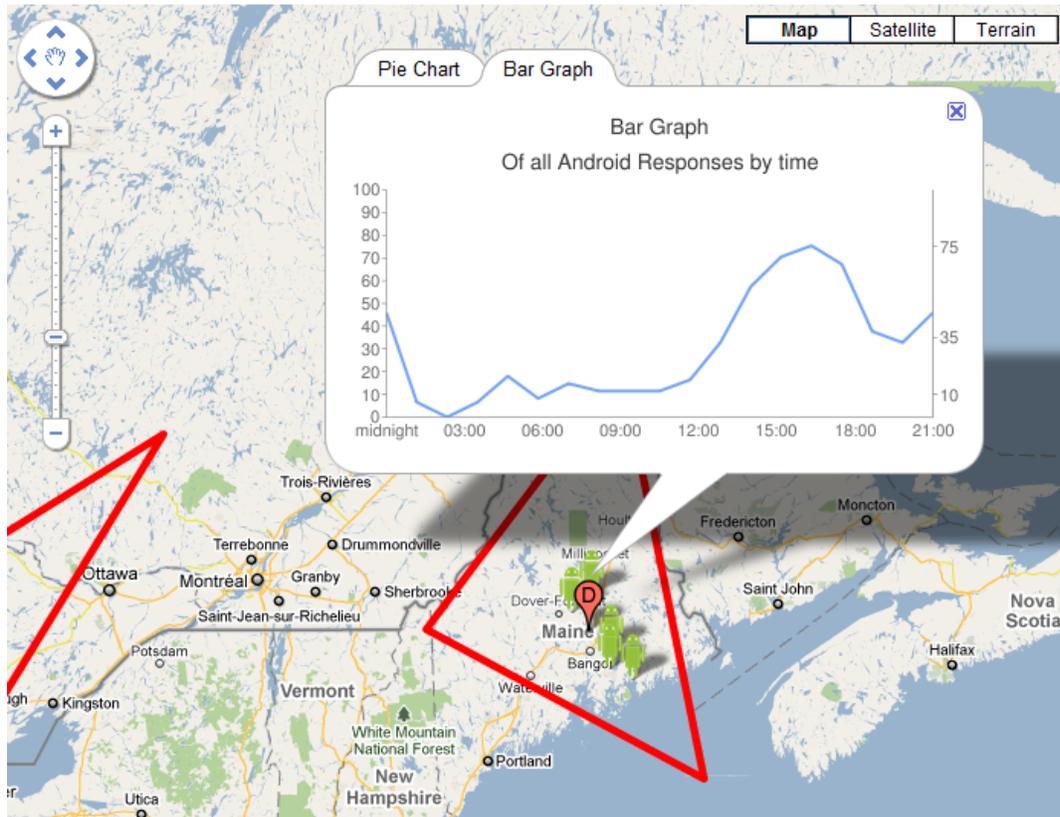


Figure 18: In the bar graph, the X axis is 'Time when the Android user responded' and Y axis is the 'Number of responses'. This allows the market researcher to discover the distribution of responses across time of day. This is particularly important if the question is located in office area or a university where this distribution may reveal interesting patterns.

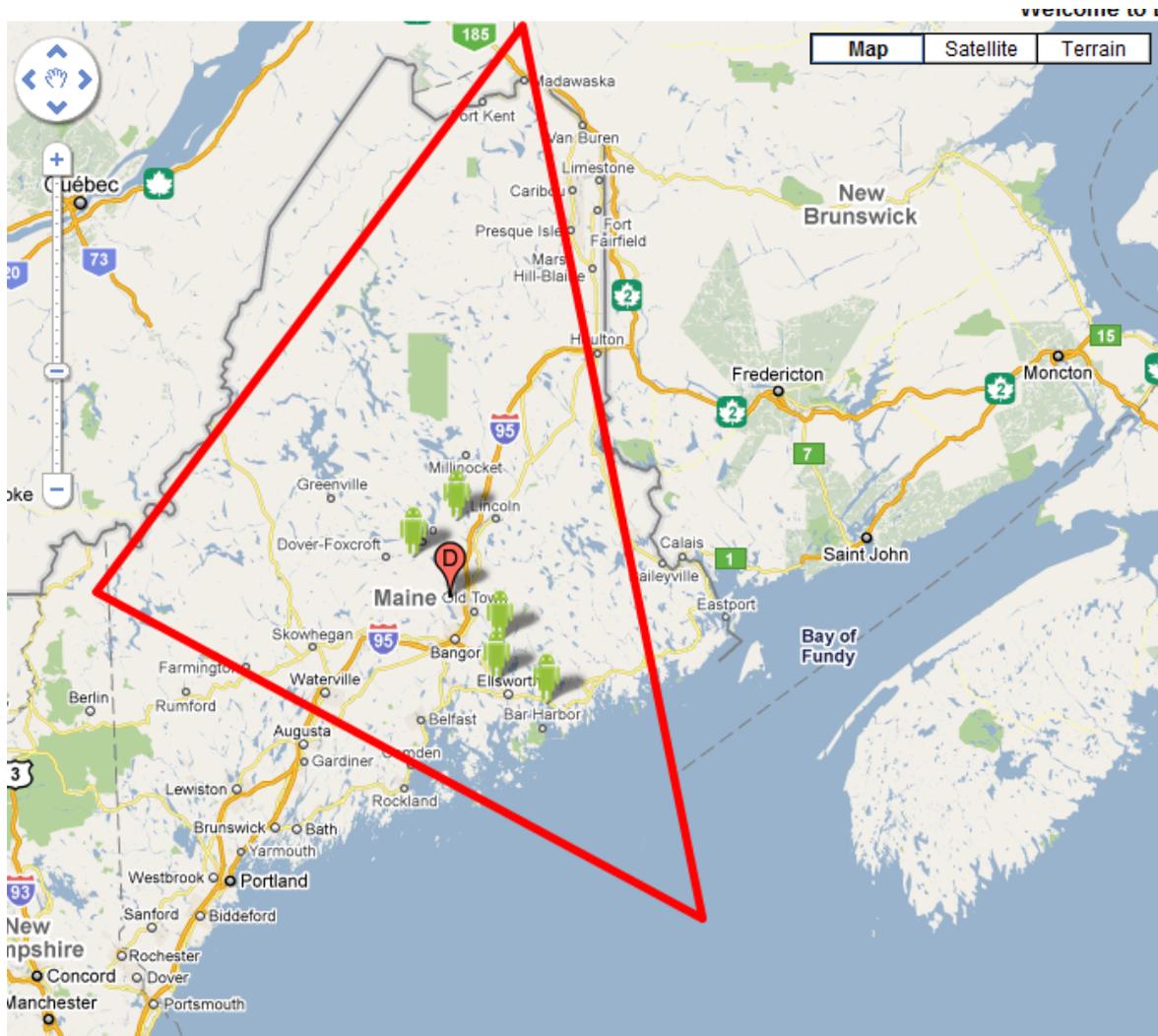


Figure 19: Clicking on each individual response i.e. the green marker, the market researcher can discover that answer. For instance, by clicking on the lower right green marker, the question template box updates to show Choice 5 is selected via the <Selected> string prepended.

We had originally proposed a server push mechanism for real time communication between server and client instead of polling. This was not implemented because applications hosted on the Google App Engine run in a limited sandbox execution environment. This sandbox does not allow creation or spawning of threads. This is a serious limitation as server push technology exclusively depends on spawning of threads at server side. This limitation can be overcome by porting our server side code over to Amazon EC2 or Windows Azure.

4.0) What would you do differently, what did you learn?

GAE has a number of shortcomings. It does not allow any creation of threads or making system calls. Applications hosted on GAE run in a limited sandbox execution environment. This is an obstacle moving forward in terms of scalability and providing memcache support on GAE. A better and reliable option is instead to port the server code over to Amazon EC2. Amazon EC2 unlike GAE allows developers complete freedom from the OS layer overcoming GAE's limitations.

On the server side we successfully deployed the publish/subscribe framework and learnt to use GWT in implementing the MVC design pattern for insertion of questions and information visualization on the responses from Android users. Two weeks spiral and continuous iterative design revision allowed us to follow the 'User-Centered Design'[1] principle on the Android device. UCD design principle allowed us to reuse most of our code for retrieval of questions (XML or QR code) and apply different views such as map view and table view on the raw data.

On the project management side, we learned how to get familiar with the spiral method of project management. One thing that could have been done differently is to specifically define all of the features and goals of the application at the beginning and to set each of the spirals then as well. This would prevent the project from going astray and adding additional features that aren't already in the scope. This would also allow for more detail to be put into the functionality of the features that are originally defined.

5.0) Contribution by Group Members

Abbas Attarwala:

Server Side (Google App Engine/ Google Web toolkit):

- a) Implement Publish/Subscribe framework
- b) Implement ModelViewController architecture on the frontend.
- c) Integrate GWT and GAE i.e. RPC method calls for deployment of server code on cloud.
- d) Generate QR code
- e) Information visualization on Android responses
- f) Design of UI on the frontend and integration with the datastore.

Client Side (Android):

- a) Map visualization on Android phone.
- b) Worked on first revision of Incentive screen.

Abhishek Das:

Server Side (Google App Engine/ Google Web toolkit):

- a) Worked on initial version of the Publish framework
- b) Information visualization on the WebUI

Client Side (Android):

- a) Design of UI on the Android.
- b) Map visualization on the Android
- c) Implement Subscribe framework from the Android side
- d) Implement the incentives: UI and framework
- e) Implement data sending/receiving between GAE and Android
- f) Implement QR functionality.
- g) Use Object Oriented Design patterns for code reuse.

D'André Wilson:

Apper Contribution

- a) Providing the main premise for the mobile app
- b) Ensure useful data for market research is collected and displayed
- c) Searching possible incentives (points/coupons/social media)
- d) Collect data on surveying formats/question styles
- e) Test user-friendliness of app

6.0) App Projects: App Context.

Though my research is within financial engineering, the research group is the Centre for Management of Technology and Entrepreneurship. A part of what we do is entrepreneurial studies. When starting a business, extensive market research needs to be done to determine whether your business venture has a market and whether that market is large enough. Once you determine that you will also need to find out what the demographic of your market is. Then you will need to determine what they are specifically looking for in a product, what features are more valuable to them, among a variety of other questions. A concern many entrepreneurs face is how to collect this data.

In the past data was collected, by mailing surveys to homes. There have also been businesses that ask consumers to complete on location. Now there are newer methods, such as website and email surveys. However, taking the survey to the consumer at anytime in any place hasn't been something that has been accomplished before. By using the technology of smartphones, these surveys can now be sent directly to a consumer. Consumers can answer market research questions at their convenience. This will hopefully allow for more market penetration and a higher response rate to questions.

The original idea for this app was to be a tool for entrepreneurs, giving start-up businesses direct access to a broad market in order to determine the size and demographic of the market they should target. By offering it the app free to consumers, entrepreneurs can poll anyone at anytime and gain data for their market research.

The key to this application is the incentives. Unlike many other surveys that may enter you for a raffle draw where, as many people know, your chances of winning are very slim. The idea for the app would be to use a point system that can be gained per question answered. The idea is that these points will then be redeemable for prizes of the consumers' choice, be it music or other apps, or even coupons from the businesses that post the questions.

The most intuitive thing about this app is that it essentially collects market data on itself. The app can be used to figure out what motivates which people to fill out surveys, when they usually fill out surveys, where they fill out surveys and who fills out which type of survey. The one flaw with this app is it is limited to the fact that only people with smartphones will have access to it. This will of course skew the data to reflect answers from the demographic that uses smartphones, for example the age range, salary and education level of smartphone users. But again this is beneficial due to the fact that we can then learn more about the demographic of smartphone holders. Upon the realization of this, this app can be very valuable, not only to entrepreneurial studies and market researchers but also to larger companies that would find this data extremely valuable.

7.0) What Next?

We plan to take this project beyond the classroom by implementing the following:

- 1) Incorporate OpenID authentication on the server side, allowing users to authenticate themselves using Google Account, Facebook, LinkedIn and Yahoo.
- 2) Improve on the incentive utility that is connecting our app with iTunes or Amazon Music for downloading of music to phone.
- 3) Allow business owners to enter custom incentive on the server side such as ‘free buffet lunch in Chinese restaurant’ that can be correctly validated and verified from the market research ‘person’ and Android user.
- 4) Show a larger variety of collected data, such as gender, age, education level, and be able to analyze the results to show specific data correlations. (e.g. correlation of answers to only female users or only users within a certain age range)
- 5) Convert to compatibility with other mobile platforms such as iOS, RIM, Windows etc.

We are considering the option of demoing this application to larger audience in ‘DemoCamp’ and to CS professors such as (Ron Becker) for feedback and future direction of the application. This application will be distributed for free to market researchers. The idea is to get them excited about the possibilities that our framework provides them in retrieving high quality responses. The feedback from DemoCamp will allow us as a team to reevaluate our application in terms of business model and the potential of starting a small company.

References:

[1] <http://web.mit.edu/21w.789/www/papers/p55-kangas.pdf>

[2] http://www.ehow.com/list_6510108_survey-incentive-ideas.html

Total Word Count: 2526

Apper Word Count: 499

Total Word Count minus Apper : 2027