Whoop-Txt Report

Final Report Prepared By:

Anthony Balmeo, Jessica Kain, Zorayr Khalapyan, Wade Norris, Zachary Policzer, Parth Shah

**Preliminary: About Our Project**

Whoop-Txt is a geo-aware mobile web application which allows users to send messages to others in their group list or in their vicinity. This application offers a new way of sharing information that combines social network with the simplicity of texting. Our project operates with Facebook, one of the most popular social websites, in order to create an engaging and entertaining text-sharing environment. In addition, the geo-location feature in this text-sending medium introduces a new way of sharing social media.

**1. Project Goals**

We wanted to create an application that would offer users a new and exciting way of sending messages. Essentially, we wanted to expand on the now popular SMS messaging system with the following feature goals:

First, we wanted to make mass messaging a simple process. Within the application, a user should be able to easily create a new group of friends and then have an ongoing steam of messages that pertain to that group. This way, a user can easily mass message a group of people by whatever criteria they wish. Also, the ability to organize these messaging groups is not isolated to the phone application; users should also be able to organize these groups on their PC.

Second, we wanted to give users the ability to send messages based on their current location. Users may find themselves in situations where they only want to send messages to people who are near them at the time of authoring a message. The idea here is that it’s possible for users to get in contact and meet up with friends where it’s possible, instead of sending needless messages to acquaintances that are too far to be considered available.

Finally, we also wanted to give users the ability to virally share information based on their current location. A mass text about free Di-Di Reese cookies isn’t terribly relevant to someone who is not in Westwood. With the geo-locational aspect of the application, users can send messages to anyone within a certain radius, regardless of knowing them or not. If the message is popular, users can re-share the message, and the message’s radius of recipients grows, in effect, creating a viral message.

**2. Notable Accomplishments**

**3. System Architecture and Design**

**3.1 Architecture Overview**

**3.2 Architecture Design Decisions**

**3.3 Component Reuse**

An underlying goal when developing our software was to reuse components in order to save time and write efficient code. Different component reuse techniques were used when implementing the Whoop-Txt desktop and mobile versions.

For the desktop version, open source code from both Facebook and Google were heavily used. In particular, the Facebook JavaScript SDK as well as the Facebook PHP SDK was used as a framework in order to manipulate the social plugins available. From this code, Facebook profile pictures, user ID’s, and Facebook friends were obtained and stored into our Whoop-Txt database. This code was obtained from the Facebook Developer website.

In addition, Google Analytics and Google Adsense were reused for the desktop version as well. Components from Google Analytics were used so that we could track the usage of our Whoop-Txt application. Code was also used from Google Adsense to show targeted advertisements on our application with the hope of obtaining future profits.

For the mobile version, code obtained from the Mobile-Web Framework (MWF) was used. With this framework at our disposal, we were able to create several utilities that could be reused throughout our implementation. These utilities include the Communication utility, MWF utility, and the Whoop-Txt utility. The Communication utility parses URL parameters and is a cross-platform implementation of AJAX calls. The MWF utility generates DOM elements that are MWF formatted to offer a nice user interface display. Lastly, the Whoop-Txt utility provides a JavaScript interface with the backend API and uses the Communication utility. All of these utilities/components were reused throughout the mobile-side implementation which efficiently organized the structure of our software.

For both the desktop and mobile versions of our application, the code for the tag auto-complete boxes were reused. For the desktop version, the code was taken from the following website:

<http://loopj.com/jquery-tokeninput/>.

For the mobile version, the code was taken from the following website:

<http://www.codeproject.com/KB/scripting/jsactb.aspx>

**4. Process Design Decisions**

**4.1 Process Design Model Considerations**

**4.2 Team Organization and Process Decisions**

All process decisions and assigned implementation tasks were made as a team. Much time and effort was spent organizing project progress and the following methods were used to accomplish this.

**4.2.1 Division of Tasks/Roles within the group**

Initially, we split our team into two general development groups, one group handling the front end and the other implementing the back end of the application. However, after the development preliminaries were completed (aka requirements and design), we had a better understanding of the work that needed to be done. Since we wanted Whoop-Txt to be a mobile-web application, two different front ends needed to be implemented. As a result, we decided to have three development groups: one for the mobile front end, another for the desktop front end, and the other for implementing the back end’s database.

Anthony, Jessica, and Parth were in charge of implementing the front end of the desktop version. Anthony implemented the Message page and the corresponding functionality. Parth was in charge of the Groups page which consisted of handling the invitations and group lists as well as the group creation functionality. Jessica wrote the Help, Privacy Policy, and Report/Contact this Application pages.

Wade’s role was to implement the front end of the mobile version. This consisted of both designing the user interface as well as implementing the functionality of sending/organizing messages, sending/accepting invitations, and group creation. The authentication page for the mobile side, however, was designed by Anthony and the Facebook login was implemented by Jessica.

Zorayr and Zachary implemented Whoop-Txt’s database. In addition to this, they wrote an API documentation which consisted of the functions and example calls. This was done in order to save both of the front ends team’s time in trying to understand the back end code.

**4.2.2 Team Communication**

Our team felt that communication was imperative in making this application a success. Every Friday during discussion, we would meet to discuss tasks for the upcoming week as well as the progress of the current week. In addition to this, our team posted a weekly schedule on our Github repository so that the tasks assigned to each member were clear and concise.

In addition, the team would also communicate frequently via email, Facebook, or the Issues page located on the Github repository. Our conversation threads consisted of several topics including encountered problems, unclear tasks, deadlines, etc. These mediums significantly helped with team communication particularly when the team could not physically meet due to schedule conflicts.

**4.2.3 Other processes/Tools/methods used by group to promote efficiency**

Some additional measures were taken in order to promote efficient development. In addition to the team communication addressed in the previous section, we also attended a weekly teleconference with our IBM clients. We received helpful client suggestions and guidance from these conversations which contributed to the project organization.

For writing code, the team decided to install NetBeans IDE 7.0.1 on each developer’s machine. NetBeans was then set up to point to our FTP host so that we could make updates directly to our Whoop-Txt application on Facebook. This tool then allowed us to implement the desired functionality and easily merge/test the code. Using NetBeans, thus, promoted implementation efficiency and cohesion.

**5. Team Participation**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | % Weight | Anthony | Jessica | Zorayr | Wade | Zachary | Parth |
| Requirements Specification | 10% |  |  |  |  |  |  |
| Design Document | 10% |  |  |  |  |  |  |
| Presentations | 10% |  |  |  |  |  |  |
| Code Construction | 40% |  |  |  |  |  |  |
| Testing and Debugging | 20% |  |  |  |  |  |  |
| Overall (Weighted contribution) | 100% |  |  |  |  |  |  |

**Brief Description**

Although each member was assigned his/her own implementation task, all members contributed in writing the application documentations.

**6. Potential Future Functionality**

In the future we would like to integrate the application with google maps. We think it would be valuable and pretty neat to see how far a message traveled by getting re-shared, and seeing how many people downloaded and re-shared a message.

We’d also like to make it add friendly in an innovative way. It’d be nice if the project could use it’s geo-location to determine when users are near certain places of business, and then be able to get adds based on those businesses. Imagine walking around the Westside pavilion, and getting informed on all the deals a certain store has as you walk past it? Advertisements are instantly more valuable, as they are targeted at people who can physically take advantage of whatever offer is contained within the advert.

**7. Project Challenges/Lessons Learned**

As expected, our team encountered many challenges while developing the Whoop-Txt application. One of our first challenges was creating a project to implement. After being assigned to the IBM project, we learned that our only requirement was to incorporate some form of internet mash up in our assignment. As a result, we were given more freedom to choose our own idea. Several project ideas were formulated, but eventually we decided to create a mobile-web application that would offer users a new way of sharing information via texting.

After finding our project idea, the next task was to choose a domain name for our application. Unfortunately, this was more difficult than we had anticipated. Many of the names we thought suitable for our project were already taken by other websites. Eventually, we narrowed down our search and selected “Whoop-Txt” to be our application name.

Before we could develop our idea, we were expected to compose a project schedule, write a specification requirements document, and design the architectural overview of our application. This proved to be difficult since, as a team, we came up with our own project idea. Our original features and constraints were not very concrete in the early stages of the project and we knew these would undergo changes during implementation. However, it was paramount that our design and requirements be as accurate as possible so that the development stage and the design would not drastically change. We struggled with defining clear cut requirements since the clients’ expectations were limited. Thus, writing these documents took more time and effort than anticipated, but in the end it significantly helped define our application idea.

Task delegation, however, proved to be challenging in the early stages of development. In particular, without a fully implemented database, both the desktop and mobile front end functionalities could not be coded. Thus, during this time, both front end teams could only create UI prototypes that would have to be updated at a later point. In addition, it was challenging deciding on a UI design that suited each team member’s tastes. We wanted the UI interface to be easily understood, visually pleasing, and readable. Several UI layouts were suggested and it took several attempts before finding our final version.

The most difficult task that we encountered for this project was integrating the database with the front end skeletons. Before accomplishing this task, the two front end teams had to understand how to communicate with the database and what functions were provided. Luckily, this was easily done since the back end team wrote API documentation as well as a breakdown of the database structure which was available on the project Github repository. This documentation consisted of all the functions implemented, their necessary parameters, and example AJAX requests and JSON responses. This was extremely helpful since we were able to treat the database as a black box without going into the intricate implementation details. While implementing the front end functionality, however, we discovered that the database needed to be updated and functions needed to be added. Thus, our team had to integrate and develop simultaneously in order to successfully meet our scheduled deadline. However, it was better to integrate while still implementing the front end functionality since errors were found and resolved well before deployment.

**8. Conclusion**

In the end, our team successfully accomplished our original goal in creating an application that introduces a new way of sending text messages. Although we experienced difficulties in getting started, we were able to overcome the many challenges faced by staying strong as a team. Through careful planning, project management, and weekly status meetings, we were able to delegate tasks in order to satisfy our scheduled deliverables.

We would like to thank the IBM team for all of their help and support. We hope you enjoy using this Whoop-Txt application.

**Project URLs**

The source code for the Whoop-Txt application is loaded in the team’s Github repository. Here you can also find the team’s weekly schedule, the API documentation, as well as the database structure under “Wiki.”

Github URL: <https://github.com/zkhalapyan/whoop-txt>

Weekly Schedule URL: <https://github.com/zkhalapyan/whoop-txt/wiki/Weekly-Schedule>

API Documentation URL: <https://github.com/zkhalapyan/whoop-txt/wiki/API-Documentation>

Database Structure URL: <https://github.com/zkhalapyan/whoop-txt/wiki/Database-Structure>

The desktop and mobile versions of the Whoop-Txt application are located at:

App Website URL: <https://apps.facebook.com/whoop_txt/>

App Mobile Website URL: <https://rocking-apps.com/whooptxt/mobile/>

**Contacting the Developers**

If there are any issues with the Whoop-Txt application or implementation questions, the developers can be contacted at the following emails. We will be happy to assist you.

Anthony Balmeo………………………………………………………………………..[anthonybalmeo@ucla.edu](mailto:anthonybalmeo@ucla.edu)

Jessica Kain jkain@ucla.edu

Zorayr Khalapyan……………………………………………………………………….[zkhalapyan@ats.ucla.edu](mailto:zkhalapyan@ats.ucla.edu)

Wade Norris [wade.norris@ucla.edu](mailto:wade.norris@ucla.edu)

Zachary Policzer………………………………………………………………………...[zacattackftw@gmail.com](mailto:zacattackftw@gmail.com)

Parth Shah [parthms@ucla.edu](mailto:parthms@ucla.edu)