**Survenator, the Survey Creation Machine**

**Summary of Project**

**Members:**

Travis Bangs – Project Manager/PHP Backend Developer

Colby Troutt – Database Manager

Samuel Roman – Web Developer

Andrew Vance – Android App Developer

**Technology:**

Server OS: Ubuntu Server 14.04.2 running Apache Web Server

Database: MySQL

Languages: HTML, PHP5, Javascript, Java (Android App)

GitHub: <http://github.com/Tarvis451/Survinator>

URL: <http://travis-webserver.dyndns.org:81/>

Hosted at home

**Summary:**

Despite the years of computer science course experience the group has, none of us have worked with a web server before, however we learned very much along the way. We used an Agile approach, as with little knowledge about what would be required beforehand, it would be impossible to create specifications in advance. The project was designed before we learned much about design patterns, therefore it does not use a strict object-based approach. However, the behavior of the application most closely resembles heavy use of the Factory pattern. Almost every action is done by passing some parameters to PHP functions, which then takes care of validating then adding and requesting information from the database. Therefore, the information within the database could be considered as the ‘objects’ that the factory functions create.

One main goal that we wanted was to share as much code between the android app and web app as much as possible, to eliminate a lot of duplicate work. Therefore, the PHP scripts that do all of the backend work were implemented as functions, with small wrapper scripts that the Android app can call, which then calls these functions. That way, the functions themselves are shared between the web app and the android app. In the end, this means that the website and the android app are little more than frontends that implement their own user interfaces.

The following are features the project implements: user registration and login, survey creation, dynamic multiple choice questions allowing any number of responses, survey taking, live results viewing. All of these features are implemented both in the web and android apps.

Given more time, we would have liked to implement: survey and question editing after publication, and the ability to delete unwanted surveys and questions.