**GeoChat**

Created by: Kyle Deak and John Delaney

CMSC 495

Design Plan

**July 16, 2017**

# Overview

The basic construct of GeoChat is a web based infrastructure. For this project we will be using the XAMPP stack which includes a preconfigured Apache Web Server, PHP, MySQL, Mercury Mail. The goal is accessibility from any browser on mobile or desktop device. The browser will establish an http connection on 80 and connection to the Apache Web Server. Using php as the server-side language to communicate to MySQL, the html will utilize JavaScript and AJAX calls to pass client information to and from the php methods. Using JavaScript the information can be updated without the need to refresh the page.

# Goals

1. **Geolocate Chat services:** Produce an application that allows users to chat and be displayed on a map so others around them can see their thoughts.
2. **Provide an intuitive GUI for application usage :** Produce a Graphical User Interface that users can navigate without direction or much help from a user guide.

# System Architecture

## User Registration

The user registration page is provided to allow the registering of users for the web-based program. The passwords will be hashed and salted and unrecoverable.

|  |  |  |  |
| --- | --- | --- | --- |
| Requirement Number | SRS | Developer | Suspense Date |
| 1.1 | Web interface for user information | Kyle Deak | 7/23/17 |
| 1.1.1 | Create HTML front end with userform |  |  |
| 1.1.2 | First Name textarea id=”firstname” |  |  |
| 1.1.3 | Last Name textarea id=”lastname” |  |  |
| 1.1.4 | Password textarea id=”whatever” |  |  |
| 1.1.5 | Confirm Password textarea id=”whatever2” |  |  |
| 1.1.6 | JavaScript validation method for password comparison |  |  |
| 1.1.7 | Email textarea id=”email” |  |  |
| 1.1.8 | JavaScript email validation method |  |  |
| 1.2 | MySQL table built to house username, password hash, email, first name, last name, last login date. Named User | John Delaney | 7/17/2017 |
| 1.2.1 | Create MySQL table |  |  |
| 1.2.2 | Add Column First\_Name VarChar 100 |  |  |
| 1.2.3 | Add Column Last\_Name VarChar 100 |  |  |
| 1.2.4 | Add Column Username VarChar 100 |  |  |
| 1.2.5 | Add Column Email VarChar 150 |  |  |
| 1.2.6 | Add Column Password VarChar 512 |  |  |
| 1.2.7 | Add Column Last\_Login Date |  |  |
| 1.2.8 | Add Column UserId VarChar 20 |  |  |
| 1.2.9 | Add Column Icon LONGBLOB |  |  |
| 1.3 | Php script to interact with web page and database to validate and place data in database. | John Delaney | 7/23/2017 |
| 1.3.1 | Php method for connecting to DB |  |  |
| 1.3.2 | Php method for data validation (size) |  |  |
| 1.3.3 | Php method for posting to DB catch error/success and throw message back to Web |  |  |
| 1.3.4 | PHP method to add generic image to DB upon registration |  |  |
| 1.4 | Web modal to catch returned information from a successful or unsuccessful registration. |  |  |

## Login Page

The login page will be the user’s ability to authenticate and enter the main application. The user will enter a username and password and credentials will be passed through PHP and compared to the information stored in the database.

|  |  |  |  |
| --- | --- | --- | --- |
| Requirement Number | SRS | Developer | Suspense |
| 2.1 | Create User Login Page | Kyle Deak | 7/23/17 |
| 2.1.1 | Create HTML with userform |  |  |
| 2.1.2 | Username textarea id=”username” |  |  |
| 2.1.3 | Password password field id=”pword” |  |  |
| 2.1.4 | Create modal to catch login error to display |  |  |
| 2.2 | Create PHP authenticate.php | John Delaney | 7/23/2017 |
| 2.2.1 | Create php method to select user info |  |  |
| 2.2.2 | Create php method to compare credentials |  |  |
| 2.2.3 | Create method to report bad login and pass message back to web. |  |  |
| 2.2.4 | Create method to grant access to main application |  |  |

## Main Web Application

The main web application will house the map and chat box for users to interact with the application. Users once successfully log in they will be able to type into the chat area and their message will be displayed in a message box above their icon.

|  |  |  |  |
| --- | --- | --- | --- |
| Requirement Number | SRS | Developer | Suspense |
| 3.1 | Create main GUI | Kyle Deak | 7/30/17 |
| 3.1.1 | Build map using Leaflet.js place in <div> |  |  |
| 3.1.2 | Build chat area textarea id=”message” |  |  |
| 3.1.3 | Build chat submit button (works with enter as well) |  |  |
| 3.1.4 | Build upload icon (avatar) select png, jpg, gif from local drive. |  |  |
| 3.2 | Create Table in MySQL named Messages will house messages, latitude, longitude, and userid | John Delaney | 7/17/2017 |
| 3.2.1 | Add Column UserId  Match UUID constraints |  |  |
| 3.2.2 | Add Column Messages VarChar 250 |  |  |
| 3.2.3 | Add Column Lat Numeric |  |  |
| 3.2.4 | Add Column Lon Numeric |  |  |
| 3.3 | Create PHP for main application chatGeo.php | John Delaney | 7/30/2017 |
| 3.3.1 | Create method to DB messages on submit |  |  |
| 3.3.2 | Create method to Retrieve messages and coordinates |  |  |
| 3.3.3 | Create method to DB image (avatar) |  |  |
| 3.3.4 | Create method to retrieve new image |  |  |

## 4. Server Side Security

|  |  |  |  |
| --- | --- | --- | --- |
| Requirement Numer | SRS | Developer | Suspense |
| 4.1 | All php calls to the DB will utilize best practices for secure transactions | John Delaney | As Needed |

# Processing Design

# Prototype Design for the Map Page

# RDBMS Design - MySQL

