Software Requirements and Design Document

For

Group 14

Version 1.0

Authors:
Conner F
Ryan R
James H
Davion M
DJ S

1. Overview

Gustbuddy will be a functional web application that will deliver a myriad of weather information to the user which can be shared through social media. We are proposing to develop the web app with a variety of APIs and custom algorithms to refine and personalize weather data for users accessing the app.

The system aims to deliver relevant, personalized, location-based weather information to users. Florida specific data related to weather and hurricane forecasts, hurricane data and statistics, and

2. Functional Requirements

1. Day/Week/2 Week weather forecast (HIGH)
2. On-this day past weather report breakdown (HIGH)
3. Florida Hurricane Damage Analysis Breakdown (HIGH)
4. Hurricanes in Florida Analysis breakdown (HIGH)
5. API data utilization (HIGH)
6. User login/notifications (MEDIUM)
7. Florida Focus (MEDIUM)
8. Social media sharing implementation (LOW)
9. Hurricane Probability (LOW)

3. Non-functional Requirements

1. Securing user's usernames and keeping their passwords anonymous
2. Writing efficient code
3. Attention to minimize performance lag
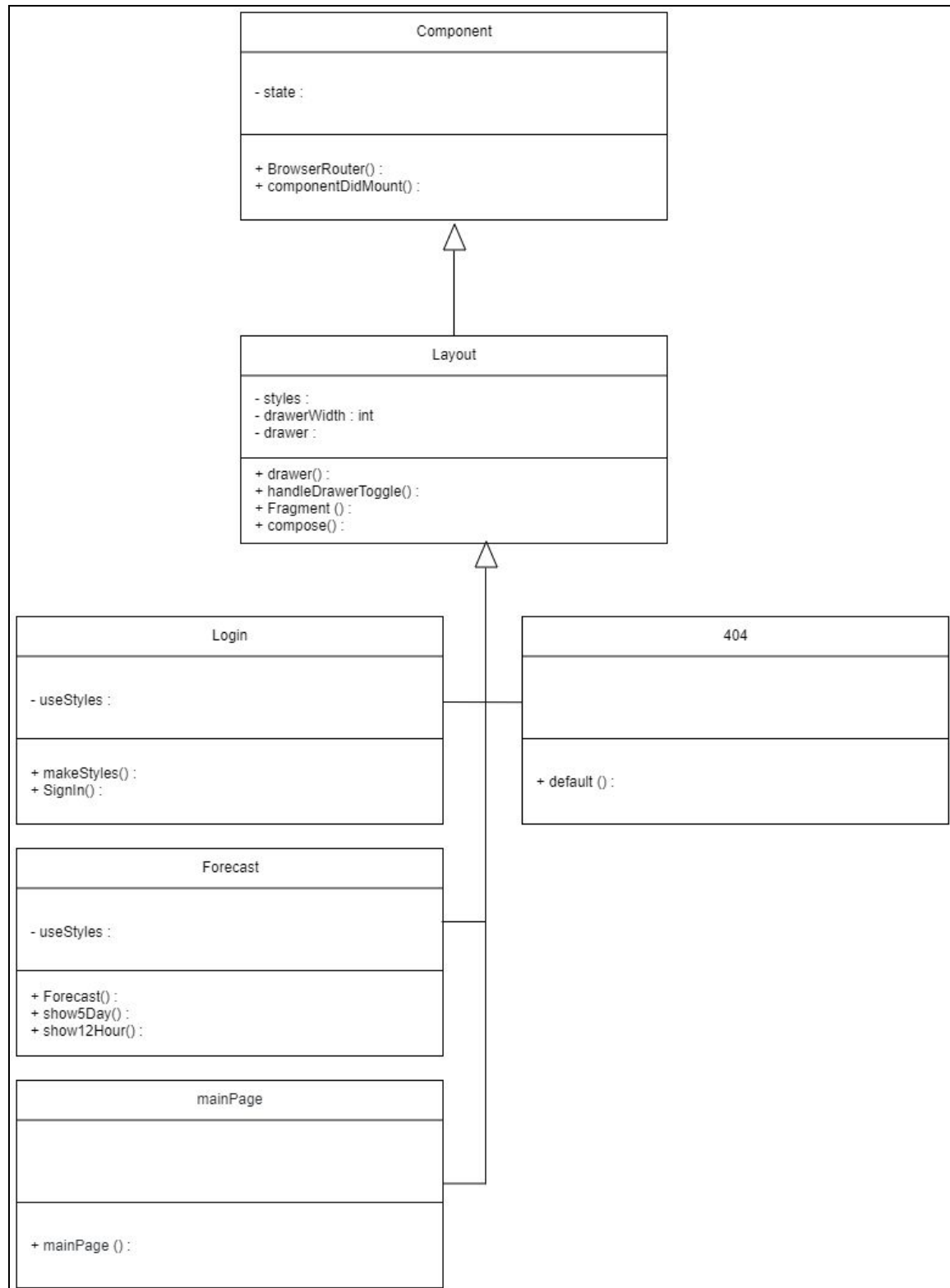4. Consistent weather updates while site is open
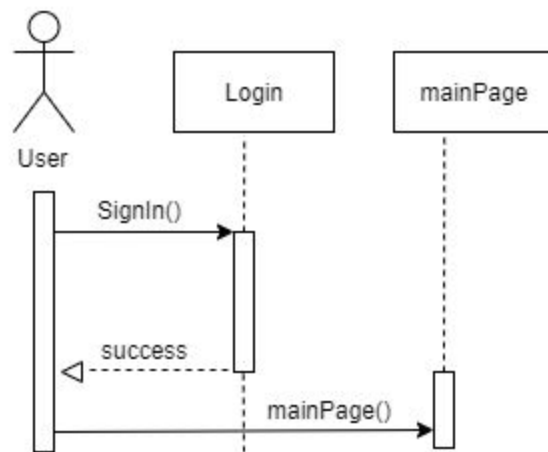
4. Use Case Diagram

Actors: User, Administrators
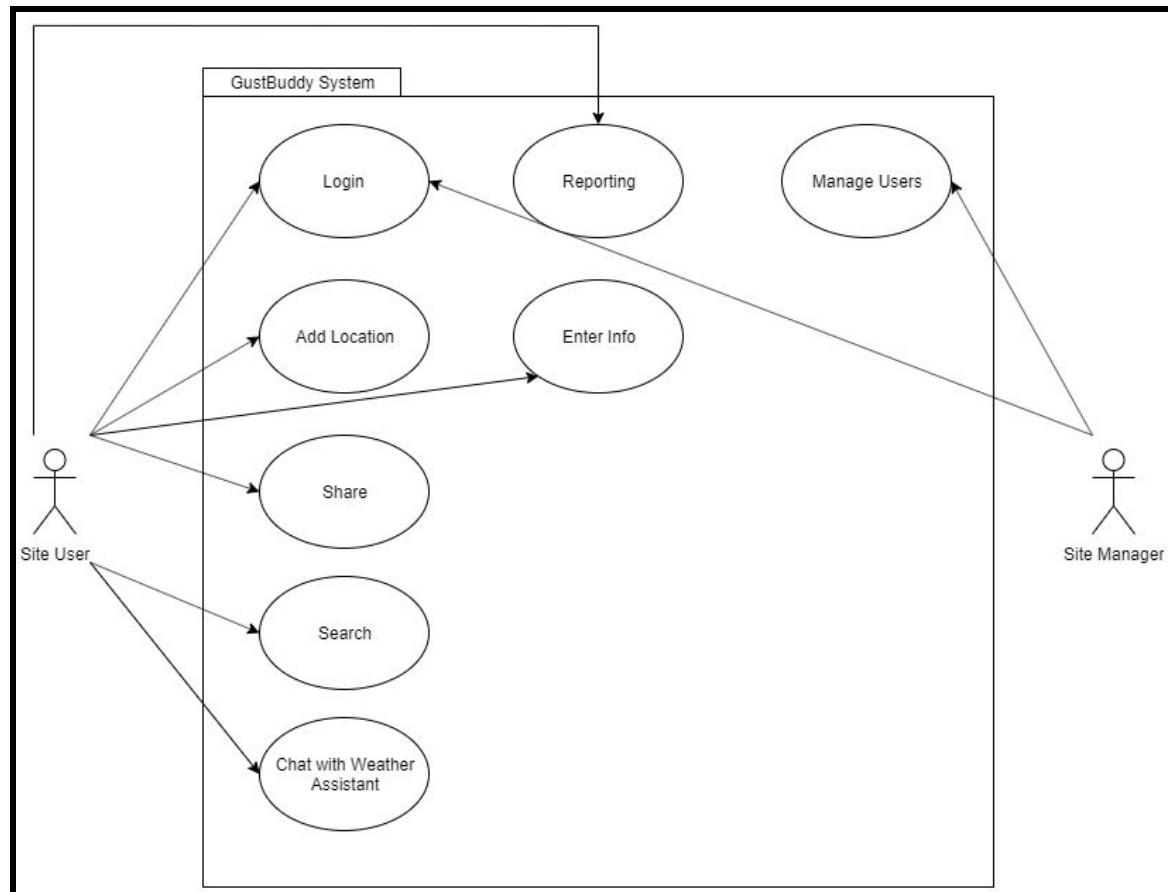
5. Class Diagram and/or Sequence Diagrams

Class Diagram:

```
┌─────────────────────────────────────────┐
│                 Component                 │
├─────────────────────────────────────────┤
│ - state :                                 │
│                                           │
├─────────────────────────────────────────┤
│ + BrowserRouter() :                       │
│ + componentDidMount() :                   │
└─────────────────────────────────────────┘
                     △
                     │
┌─────────────────────────────────────────┐
│                  Layout                   │
├─────────────────────────────────────────┤
│ - styles :                                │
│ - drawerWidth : int                       │
│ - drawer :                                │
├─────────────────────────────────────────┤
│ + drawer() :                              │
│ + handleDrawerToggle() :                  │
│ + Fragment () :                           │
│ + compose() :                             │
└─────────────────────────────────────────┘
                     △
                     │
┌───────────────────────────┐   ┌───────────────────────────────────┐
│           Login           │   │                404                │
├───────────────────────────┤   ├───────────────────────────────────┤
│ - useStyles :             │   │                                   │
│                           │   │                                   │
├───────────────────────────┤   ├───────────────────────────────────┤
│ + makeStyles() :          │   │ + default () :                    │
│ + SignIn() :              │   │                                   │
└───────────────────────────┘   └───────────────────────────────────┘

┌───────────────────────────┐
│         Forecast          │
├───────────────────────────┤
│ - useStyles :             │
│                           │
├───────────────────────────┤
│ + Forecast() :            │
│ + show5Day() :            │
│ + show12Hour() :          │
└───────────────────────────┘

┌───────────────────────────┐
│         mainPage          │
├───────────────────────────┤
│                           │
│                           │
├───────────────────────────┤
│ + mainPage () :           │
└───────────────────────────┘
```

Sequence Diagrams:

Increment 2 Sequence 1

```
     User            Login          mainPage

      |               |                |
      | SignIn()      |                |
      |-------------->|                |
      |               |                |
      |   success     |                |
      |<--------------|                |
      |            mainPage()          |
      |------------------------------->|
```

Increment 2 Sequence 2

```
     User          Login        mainPage       Forecast

      |              |              |              |
      | SignIn()     |              |              |
      |------------->|              |              |
      |              |              |              |
      |  success     |              |              |
      |<-------------|              |              |
      |          mainPage()         |              |
      |---------------------------->|  Forecast()  |
      |                             |------------->|
```

Increment 2 Sequence 3

```
     User          Login        mainPage       Weather

      |              |              |              |
      | SignIn()     |              |              |
      |------------->|              |              |
      |              |              |              |
      |  success     |              |              |
      |<-------------|              |              |
      |          mainPage()         |              |
      |---------------------------->|  Weather()   |
      |                             |------------->|
```
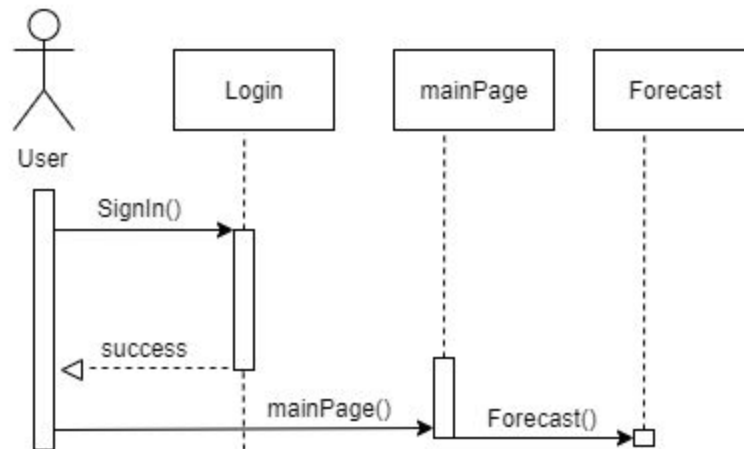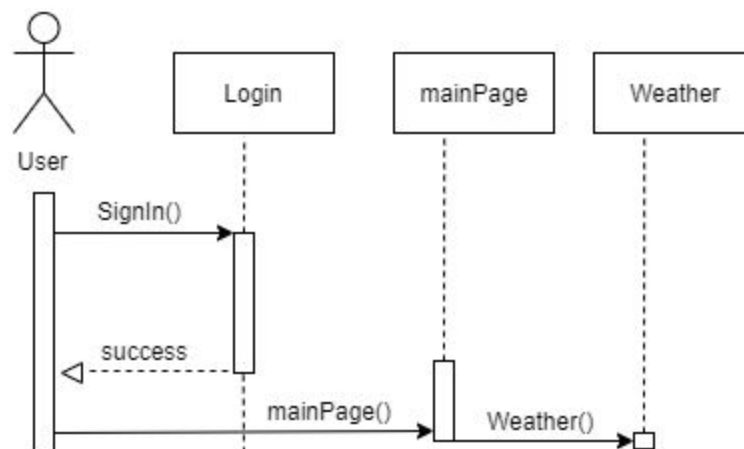
6. Operating Environment

The software currently operates as a locally hosted web application. The goal is for the web application to be published and be accessed from any remote host with an internet connection.

Hardware platform: Any
Operating system: Any

7. Assumptions and Dependencies

We plan to use React (A JavaScript library for building user interfaces), Mongo and/or SQL for our database. We will utilize a variety of APIs for various social media platforms (Facebook, Twitter, etc) as well as for incoming weather data (NOAA, etc). Various Material-UI templates will be used for the overall design of the site, including the Dashboard and various pages. The final project will also need to be hosted on a web server for access outside of development.