**Whatever our idea name will be**

# **High Level Design**

**Hello World**

**November 16, 2020**

Team Leader: Matt Caponi 011161928

Ian Macabulos 016401240

Jakub Koziol 016523856

Shaan Shariff 015741750

Wonsuk Seo 018660874

# Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author** |
| 11/16/20 | <0.1> | Started Outline | Jakub Koziol, Matthew Capone, Shaan Shariff, Ian Macabulos, Wonsuk Seo |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

## **Table Of Contents**

[**High-Level Technical Design**](https://docs.google.com/document/d/1QhOAJkD8dhwjrYRQkVO7W8WoEjSqFgCr9fOMk6vYAgg/edit#heading=h.guq5q9p7g4s0)[**0**](https://docs.google.com/document/d/1QhOAJkD8dhwjrYRQkVO7W8WoEjSqFgCr9fOMk6vYAgg/edit#heading=h.guq5q9p7g4s0)

[**Table Of Contents**](#_vcgkuflpa1pt) **1**

[**Introduction**](#_laf66kg997qz) **2**

[Goals](#_qgpdxwuv3g4m) 2

[Purpose](#_qkyb3a36rzzt) 2

[Scope](#_6ydzhwxxsaxn) 3

[Required Elements](#_y5aj2ekvjcvq) 3

[Dependencies](#_6li0h2nyq1yb) 3

[**Policies**](#_3nrz9yfrn6ml) **4**

[Security](#_k0p12gqdds3h) 4

[Encryption](#_ot69w91lpj0h) 4

[Authentication](#_rkrwjw72323n) 4

[Logging](#_q28taiebh85x) 5

[Error Handling](#_febmv1qoz17l) 5

[Communication](#_thaoa85l6dnn) 6

[Architecture](#_eh33a52kynzh) 7

[Description](#_4g8kc0ryuei3) 7

## **Introduction**

### 

### **Goals**

### **Purpose**

### **Scope**

### **Required Elements**

### **Dependencies**

## **Policies**

### **Security**

### **Encryption**

Security is of utmost importance to the development team and to ensure that all user private information is secure and unreachable by internal and external threats. All network requests and transactions will be encrypted by forcing HTTPS SSL2 encryptions to prevent any outsiders from capturing information being sent over a local network. In addition, all secure user information will be encrypted using md6 encryptions so that in the case of database leaks from insider and outsider threats critical data stays secure.

### **Authentication**

User registration is the first step to accessing our website and all of it’s services. For added security, users must provide an email during sign up in order to authenticate their identities. This makes sure that all of our users are real users that will use our services as intended.

### **Authorization**

The web application will have multiple pages which require a high level account to gain access to said pages. The high level accounts will be created by the development team giving these users access to the secure pages.

### **Privacy**

### **DDoS Mitigation**

To prevent a distributed denial of service attack our web application will be equipped with a DDoS mitigation system that will limit the amount of connection if an attack is detected. In addition our web server which runs the backend application will be secured with a DDoS mitigation system.

### **Logging**

Any activity performed by the user will be logged into the application’s database. The log will consist of a timestamp and the details of the activity including who performed the event and what the event was.

Categories of Logs:

* Trace
  + Sensitive info just for debugging
* Debug
  + Where
    - All layers
* Information
  + Where
    - Controllers
  + What
    - Flow of application.
    - Requests made by user.
    - User interactions with system
* Warning
  + Where
    - Controllers
  + What
    - Security alerts
* Error
  + Where
    - All layers that have error handling
  + What
    - Errors handled that don’t crash the system
* Critical
  + Where
  + What
    - Crashes

### **Error Handling**

* Will have error handling in the Business Layer, Service Layer, and Data Access Layer.
* Will have input validation split between the UI and the controllers.

### **Communication**

### **Database Communication**

The application layer communicates directly with the database using a tcp connection to query the datastore and have data returned to it.

### **Application Communication**

The presentation layer will be able to send http requests to the application layer to have the applicaiton compute the requested action.

### **Client Communication**

The clients will connect to the web application using a forced HTTPS SSL2 connection to ensure maximum security.

## **Architecture**

### **Description**

The “Insert Application Name here” will use a three-tiered architecture. The architecture will consist of a UI/frontend, backend, and a data store. The user will be able to communicate with the application layer by interacting with the presentation layer. The application will store data on a local database, access the data on that database and also send data to presentation layer

### **Presentation**

The presentation layer of the application is responsible for allowing the user to view all information that is requested and allows the user to interact with the application.

### **Application**

The application layer is responsible for processing user input from the application layer and computing the desired request. The application layer will access the database layer to access user requested data and return the information to the user. The application layer will also process any information entered into the presentation layer and insert the information into the database layer if necessary.

### **Database**

The database layer is responsible for storing all information required for our web app. This will include the logs for the application, user account and login information and all information critical to the system.