**General Deployment Guide**

**for**

**Spacecraft-Control-Center-Training and Testing Environment (STaTE)**

**Version 1**

**Prepared by Jesse Slager, Juliana Altamira, Kyle Garber, Jeff Cevallos, AnnaMaria Summer, Carly Bosma**

**2022-23 Senior Design Group**

**Created 04/25/2023**

**Table of Contents**

[**1. Intended Purpose 3**](#_c2rvumu8wj1n)

[1.1 Preface 3](#_qnxk81xyquaw)

[1.2 Assigned Login Credentials 3](#_x7thtliu0az2)

[1.2.1 Dr. Laskey 3](#_rv96oao3n51w)

[1.2.1 Admin - SuperUser Database Access 3](#_8rb05iii25mt)

[**2. Understanding Project Framework 3**](#_qx1y6pdg9ku5)

[2.1 Landing Page 3](#_j6bccdvgbte7)

[**3. Deployment Requirements 4**](#_r9hcdcodax7f)

[3.1 Entering Credentials 4](#_8k984qaom9pa)

[3.2 Submit Login Credentials 4](#_74uwdpw236zc)

[**4. Deployment Options 4**](#_uppu6l4rjfp)

[4.1 Layout 4](#_7nlartyess9g)

[4.2 Add New Class 4](#_i4buzgc4q5rn)

[4.3 Edit Class 4](#_tiz2cgt1ccsk)

[**5. Project Security 4**](#_1szpl1lh4m6m)

[5.1 Layout 5](#_mmdkzmestcrk)

[5.2 Sim](#_4lhrcrrg22x) [**5**](#_1vggcce4f7cj)

# 

# 

# **1. Intended Purpose**

## **1.1 Preface**

The Spacecraft Control Center Testing and Training Environment (STaTE) project application can be run locally (per machine) or as a host (to multiple users). This document outlines the steps to run the program according to design requirements and other functionalities of hosting a server application.

# **2.** [**Understanding Project Framework**](#_qx1y6pdg9ku5)

## **2.1 Django Framework**

STaTE was developed as a compromise to a mobile application with the idea that any user can access the application on any mobile device wherever the user may be. The solution to this problem was by using a Website development framework called Django allowing users to access the web application as easily as a mobile application.

**2.2 Deployment Options**

Due to the versatility of the project objectives, this application can be run both locally and as a server. Although hosting a server locally (localhost) is usually for development purposes, it can be used to host the application to machines on the same network. On the other hand, hosting the application on a server and allowing clients to connect to the server from different locations was one of the requirements of the project that was also integrated successfully. The only downside of this feature is that it requires adding inbound rules to the firewall of the computer/server that is hosting the application and having to set up port forwarding on the network router that the server is being hosted on, or using a university-assigned Remote Access Virtual Private Network (RA-VPN) or using a cloud hostings service like Microsoft Azure or Amazon Web Services (AWS).

**3. Server Hosting**

**3.1 Client-Server Relationship**

For the sake of simplicity, this guide only goes more in-depth for hosting the server as opposed to localhost. The idea of a Client-Server relationship is allowing the application to be hosted and run on one computer with multiple users (clients) connecting to the server, taking the computational load off of the client and onto the server. Below are some of the possible options for hosting the STaTE application using the Client-Server relationship.

**3.2 Private Server Hosting Instructions**

Although this is probably the most resource-demanding option, the STaTE application does have the ability to be run on any computer that is online for the entire duration of the simulation, including a personal computer at home. This is the method that was used for development testing and mock deployment. Most computers that were built within the last 5-7 years that can stay on for long periods of time can be used to host the STaTE application. The only other requirements for this deployment option are the ability to access your router configuration panel and the computer acting as a server hosting the STaTE application has a steady connection to the internet. The setup instructions are given below.

1. Once the computer that is going to be used to host the server is

# 