Requirements- Group 10

(Use case diagram not needed, discussed in meeting 2 for Sprint 1)

Background:

Our goal is to make Augur as safe and secure as possible in order to fully gain the trust and confidence of our users. In order to maintain the highest standard and class possible, it is vital that web services only provide their services through a secure connection, and the strongest happens to be Hypertext Transfer Protocol Secure or HTTPS.

Standard HTTP does not protect data from being intercepted or altered, which can result in eavesdropping, tracking, and the modification of received data. They create a privacy flaw and expose sensitive information about the users of the web services. Data sent over HTTP is vulnerable to interception, manipulation, and impersonation. HTTPS verifies the identity of a website and encrypts all vital information that is being sent. This prevents the data from being intercepted and modified.

Requirement(s):

Make Augur compatible with HTTPS in order to provide the best, and most secure and reliable service for our users.

Ensure there are no conflicts with the current framework of Augur upon adding HTTPS compatibility as to avoid causing problems that would need to be solved or maintained in the future. Ensure that existing tests still pass after implementation.

Make sure any internal links that worked with http now are properly changed to https, as this could also break some functionality

Development process:

Methodology:
Do a team mob during the initial implementation to keep everyone on the same page. Then, split up into sub-teams to tackle sub-components of the feature such as certificate generation/management, configuring flask to use the certificates, configuring all endpoints to use HTTPS.

Implementation steps:
Acquire and setup SSL certificate
Configure and update Augur (flask) to enable HTTPS
Configure and update endpoints to use HTTPS