Architecture / Platform Decision (Revised 12/09/19):

Based on each of our member's experience, we decided to use Python for our web application. We felt this would provide us a familiar playing field while learning new skills like hitting API's, connecting to a database, and using third party services like OAuth. Also, Python easily allows us to manage our dependencies using pip and virtual environments.

For the framework, we decided to use Django because there are many additional resources available and it is easy to prototype with. Also, creating a project using Django allows us to compartmentalize the project into different components. We can use Django's URL system to separate and combine components. Additionally, Django has extensive support for Bootstrap that we will use to enhance our frontend without worrying too much about HTML and CSS.

To fulfill the database requirement, we chose to use PostgreSQL. Django has built in support for relational databases with its own ORM and model mapping, which makes it convenient to manage the database. Initially we wanted to use a non-relational database like MongoDB, but these require third-party utilities, many of which are deprecated or do not exploit the benefits of a NoSQL database (they end up behaving similarly to the Django ORM and treat the database like a relational one). So, instead of using documents to have variable sized sessions, we use foreign keys to map objects (mainly users and sessions) to one another.

An alternative platform we considered was to use Flask instead of Django. We ultimately thought Flask was not a good fit because we would need to use a lot of external libraries to achieve the same functionality provided natively by Django. Two such functionalities are built in database support and the admin panel that provides a clean interface in the browser to manage the application. After prototyping with both frameworks, we solidified this choice and stuck with Django. We also considered using JavaScript for the web application but chose not to as some of us had little to no experience with it.