

Group 7:
Jessica Alberto
Megan Horan
Yishan Lin
Zehua (Harry) Zhao

CS411: Project Deliverable 3
Analysis and Architecture Decision

1. The following diagrams present our user log in/third party authentication based off our user story from the previous deliverable, utilizing major objects and components.

Diagram 1: User Log In and Third-Party Authentication Diagram

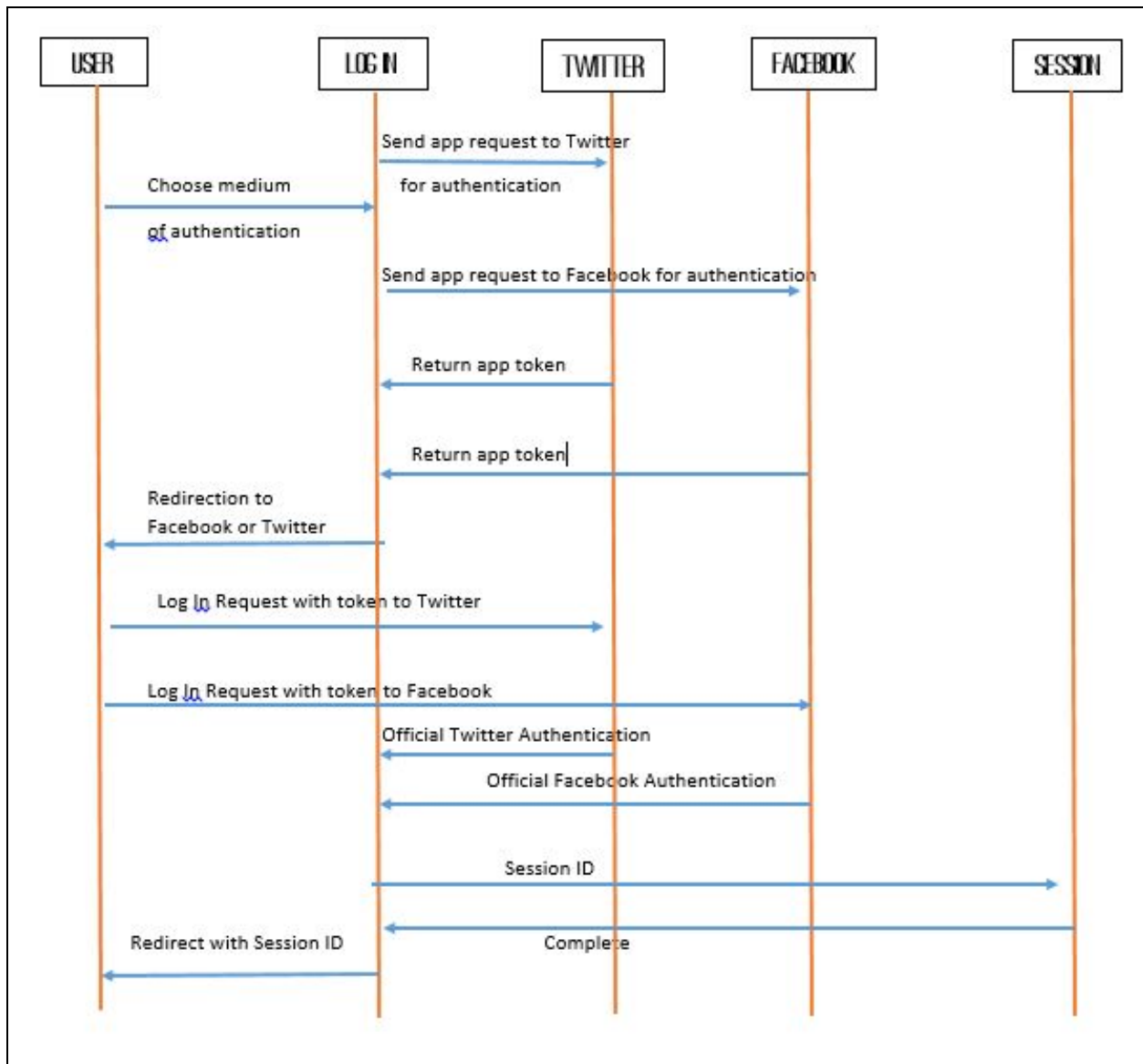
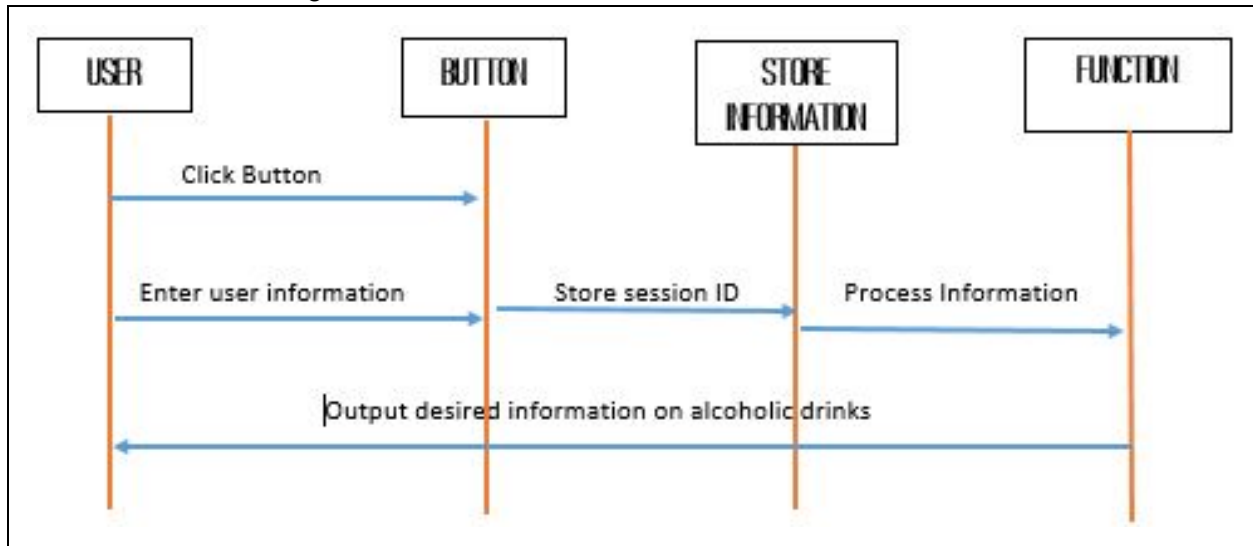


Diagram 2: User Button Interaction and Information Flow



2. User Story and Happy Path

User Story # 1:

- Perry Donham, a young and ambitious college student, wants to figure out what he's doing this Saturday night out by planning ahead. As a logged-in user, he wants to find the right alcoholic drinks to consume in order to achieve his desired level of drunkenness.
 - Perry logs in using his Facebook or Twitter account. He enters in his gender, along with weight (measured in pounds). He then selects the level of drunkenness he wants to achieve, ranging from tipsy all the way to blackout drunk.
 - Perry is also asked what his favorite drink is. After clicking the submit button, Perry then sees the number of shots/number of beers/wine glasses that will be needed to drink to achieve his selected level.

Happy Path of the User Story:

- Step 1: User logs in application with Facebook or Twitter account.
- Step 2: Get authorization from Facebook or Twitter.
- Step 3: Ask user to set user name and profile picture - this step can be skipped if user willing to use Facebook/Twitter profile.
- Step 4: User enters personal body information such as gender or weight.
- Step 5: Access user location or designated area of interest.

Step 6: Enter user's favorite alcohol type of he or she has one.

Step 7: Select the drunkenness level that user wants to reach.

Step 8: Web application shows how much alcohol (cups of beer, shots, wine) user should consume in order to reach "goal".

Step 9: User picks preference on location of target bars: Location or Bars that serve the type of alcohol user desires.

Step 10: If user chooses distance as prefer option, a window of map will be shown on the page that shows user the nearest bars according to the location/address user input;

Step 11: If user chooses specific type of alcohol to be prefer option, there will be a map with pins that shows user the bars around him/her that serves specific type of alcohol picked.

Step 12: User can click on the results shown on the map to view if the bar is busy currently and along with the review of the bar from previous customers of the bar that selected;

Step 13: After user visited the bar and consumed the amount of alcohol suggested by the web application according to the drunkenness level user selected, user will be asked to write reviews on the last experience.

Step 14: User can modify the drunkenness level accordingly if they do not select the right one - repeat process.

Step 15: With the user previous experience taken in, the web application will be able to make better, more accurate suggestions next time that user make a selection.

1. If Twitter or Facebook authentication fails, user will be alerted and redirected back to splash screen, and prompted to try again using a different account.
2. If user does not select the correct level the desired drunkenness that they intended, there will be an option to return back to the selection panel.
3. If the user's location preferences do not have any local bars nearby, he or she will be notified information about this destination, and then prompted to enter a new or nearby location.
4. If the user did not enjoy the bars visited throughout their night, they can log their night's experience and will modify the application accordingly.
5. If the user believes that the readings were not accurate, such as alcohol consumption or bar optimization, he or she will be given the option to Start Over.

3. Decision on Project Architecture / Framework

- We intend on using Node for our back-end and then Angular.JS in our front-end. We believe this particular architecture is the right choice because a few team members have been experimenting and playing with Node/JS and have found the framework easy to implement and intuitive. The documentation for both of these frameworks is also very intensive as these are two very trendy frameworks. Angular allows us to make calculations in real time on the front-end while Node implements the RESTFUL/Get. We're going to use Express, Nodemon, and likely MongoDB to work in conjunction with our Node.