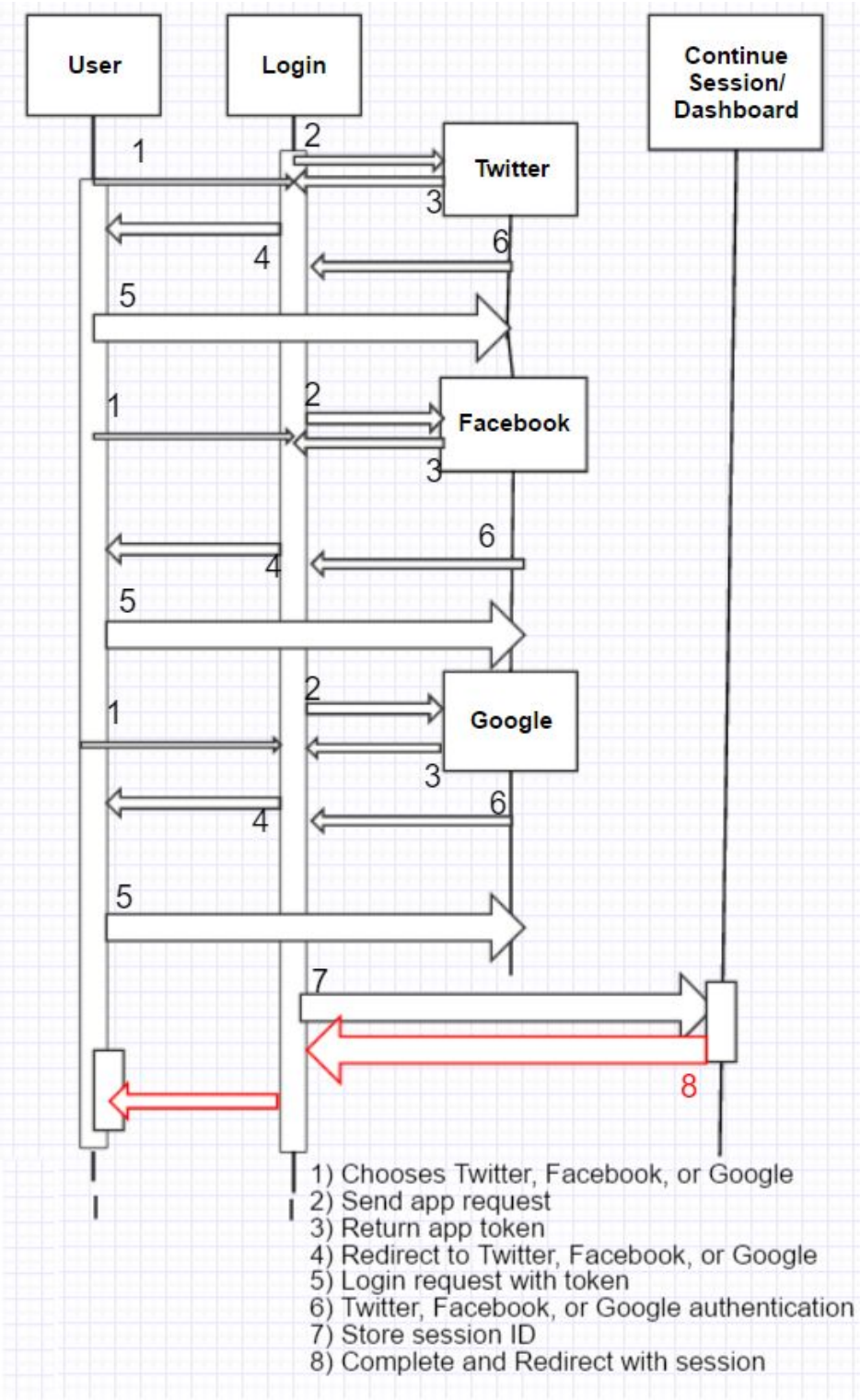
1.)


User
Login
Continue Session/ Dashboard
Twitter
Facebook
Google
1
2
3
4
5
6
7
8
1) Chooses Twitter, Facebook, or Google
2) Send app request
3) Return app token
4) Redirect to Twitter, Facebook, or Google
5) Login request with token
6) Twitter, Facebook, or Google authentication
7) Store session ID
8) Complete and Redirect with session

2). **Create Account:**

1. Set Username (Does not exist, Valid)
2. Set Password (Meet criteria, Successful)
3. Enter email address and other information(valid email)
4. Finish register
5. Open verification email (Click link in the email to activate account)
6. Setup profile/preference
7. Add friends (Successful)

Create Account (Wrong Step):

1. Set Username (Already exist, Invalid)
    a. System notice Username Username already exist
    b. Allow User retype Username
2. Set Password (Does not meet the criteria)
    a. System notice User that password format is wrong
    b. Allow User set another password
3. Enter email address (Invalid email)
    a. System notice User that email address is invalid
    b. Allow User re-enter email address

5. Open verification email
    a. Link sent unsuccessfully
    b. System notice User to resend verification email

7. Add friends
    a. User does not exist
    b. System notice that user does not exist and enable re-enter user id

Log In:

1. User provides Username (Valid)
2. User provides Password (Successful)

OR:

1+2. User use facebook/twitter account to log in (Successful)

3. The system validates that the password matches the one on file
4. Choose whether want to stay logged in
5. Finish log in

Log In (Wrong Step):

2. User provides Password (Unsuccessful)
    a. System notice log in unsuccessful
    b. Enable user re-enter username/password

OR:

1+2. User use facebook/twitter account to log in (Unsuccessful）
    a. System notice log in unsuccessful
    b. Enable user retype account information

5. Log in (Unsuccessful)
    a. System notice user log-in unsuccessful

Enter Search Term(Correct Log In):

1. User provides valid search term
2. Map shows venues based off the users search terms

OR:

Search Term is Invalid:

1. No output, user is asked to input search term again

4. Our team has decided to use a node+express backend framework, with MongoDB for the database, and possibly angular or react for the frontend framework. We choose node and express because of its ease of use, our team having some experience with node, and also the fact that many companies are switching to a node/express based architecture, it seemed like the viable choice to use. Python is our alternate choice, however it is not as easy to setup and use in contrast to javascript, which is a more web-oriented language. MongoDB was used because we don't need a full relational database for our website, much of our data will be obtained in real-time from an API, as such our database will function more of a cache(for the time being, the use of the database may change in the future) for user searches. The ease of use and speed of a non-relational database, along with our team having some experience with MongoDB made it a viable choice. For the frontend, we will use either angular or react js to render the view to the user, this is the learning opportunity for our group as we are not familiar with either framework and want to learn something new. For the time being we will use both in our prototypes to see which one would fill our needs, and decisively choose one later on.