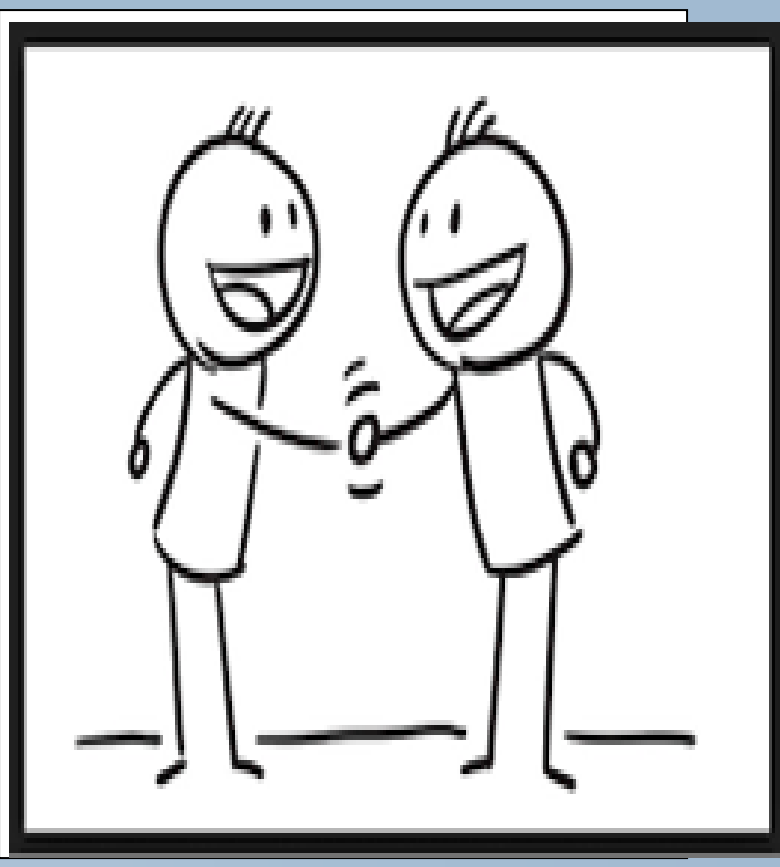




Friend Finder

Jake Anderson, Sedek Ciprien, Eric Easley, Brendon Rhodes
Grant Luna



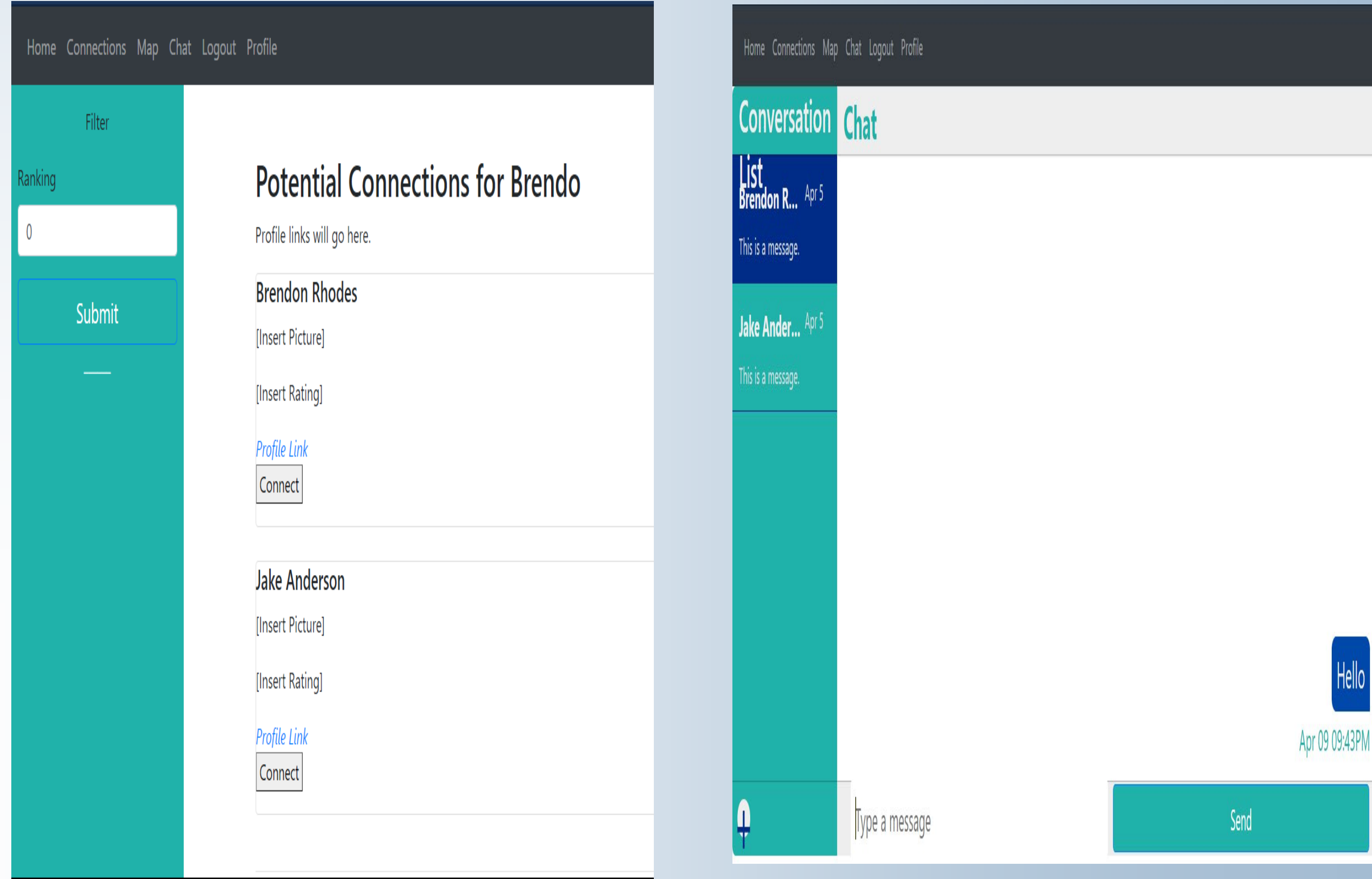
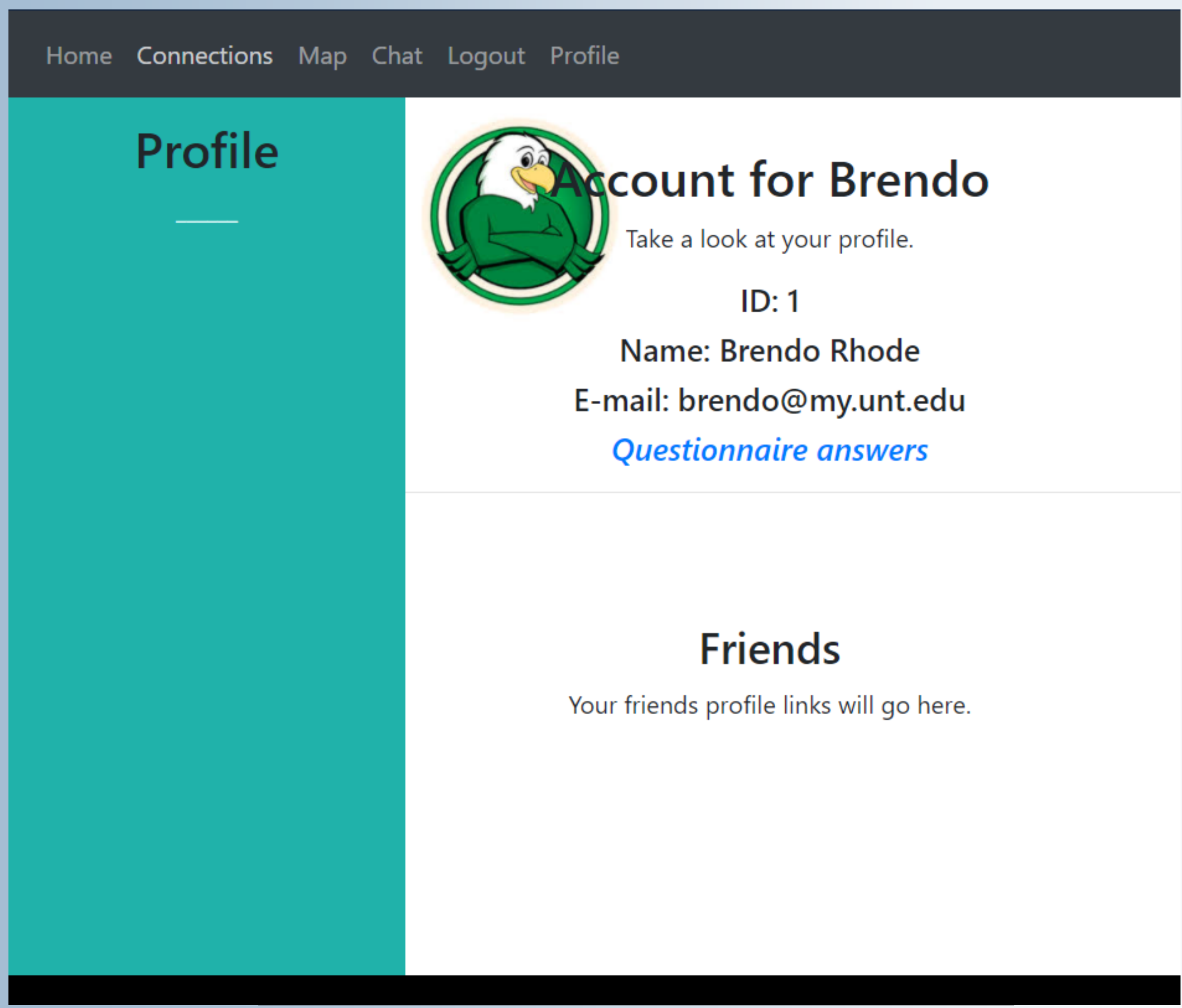
OVERVIEW

In the midst of social distancing, it is difficult for one to conveniently network with other people. Therefore, our aim of this product is to allow users to connect with like-minded individuals regarding personal hobbies. Unlike other social networking apps, this product will find people with similar interests as the user instead of having the user manually seek them out, along with the added benefit of filtering random individuals.

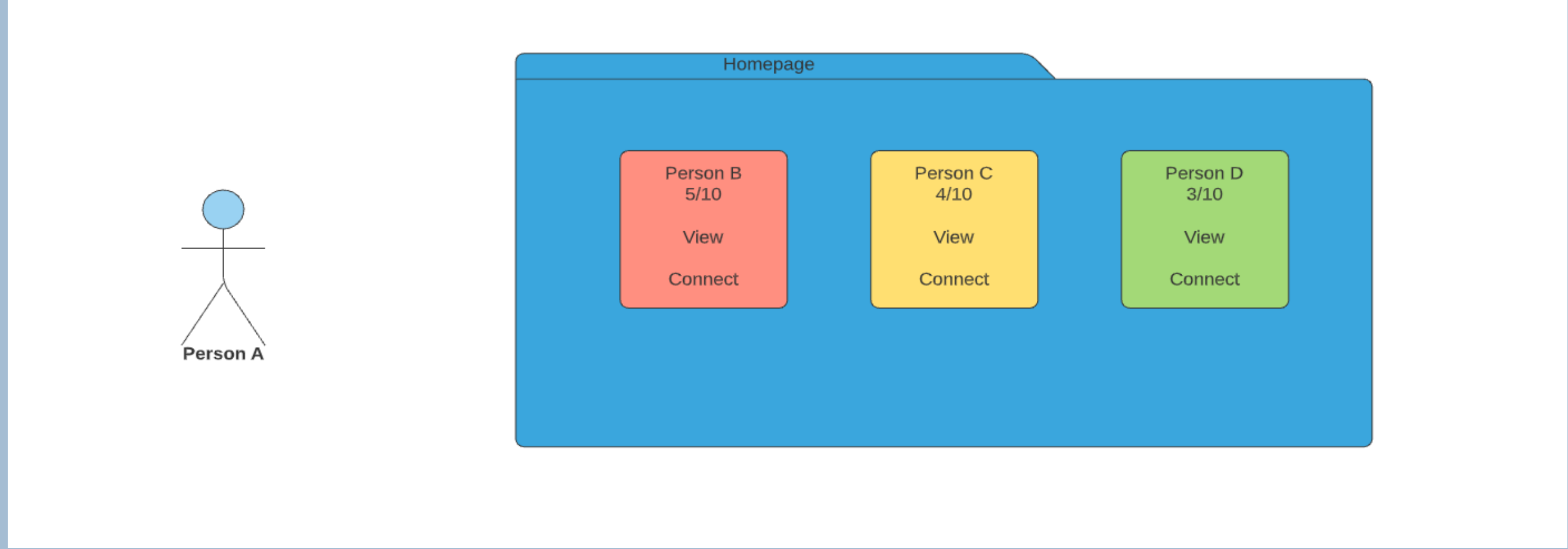
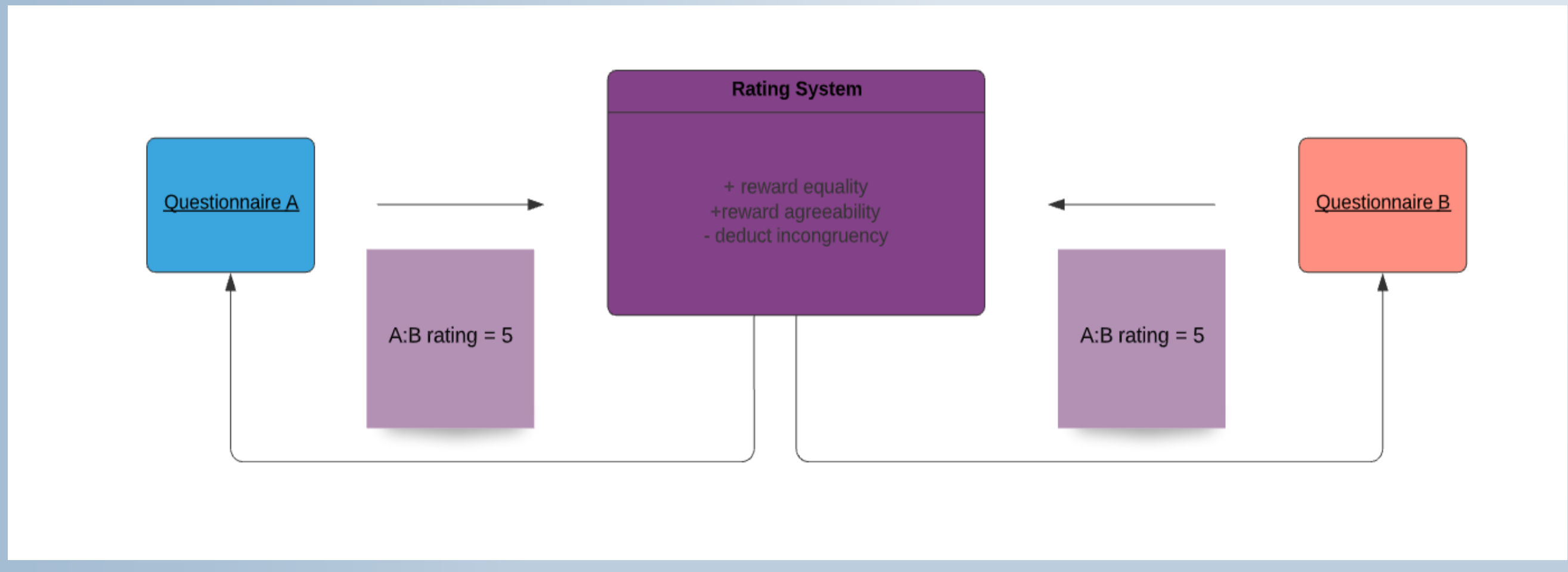
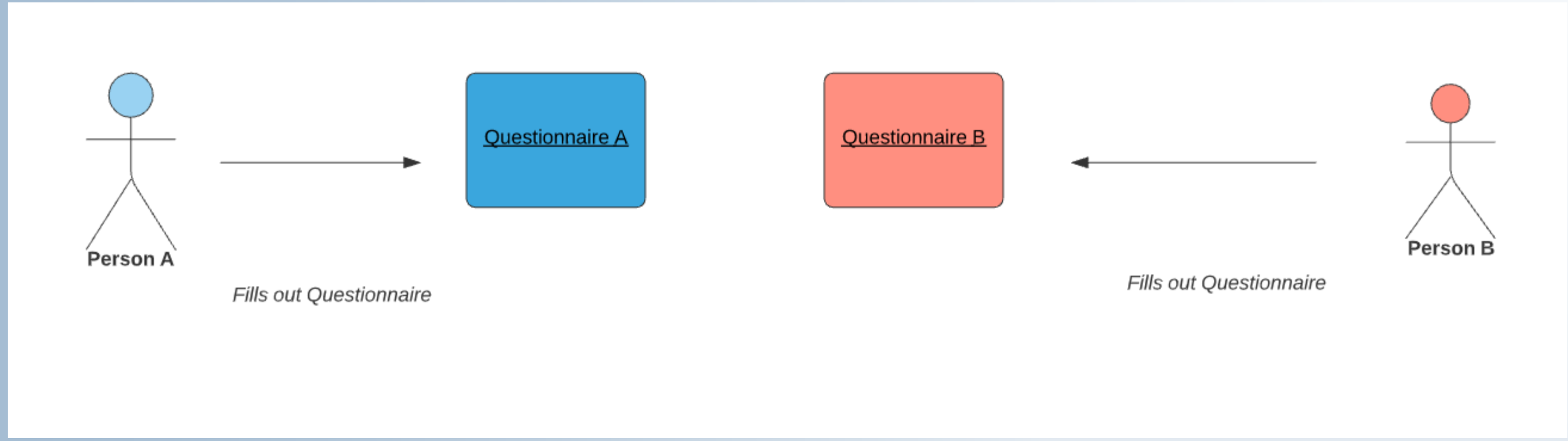
People will have complete visibility over the compatibility other users in the system. Using one's answer to the questionnaire, our system gauge the compatibility of each pair of people and give them the discretion of connecting with whomever they please.

FEATURES

- Chat**
Start direct communication with connected individuals
- Ranking System**
The product will us tally up a person's characteristics (from the questionnaire) and produce a score. The higher number signifies compatibility with the user.
- General Location Map (Geolocation)**
User's general location will be displayed on map to visualize the distance between connections and places for people touch base.
- User Dashboard**
Display the user's current connections and preferences



DESIGN



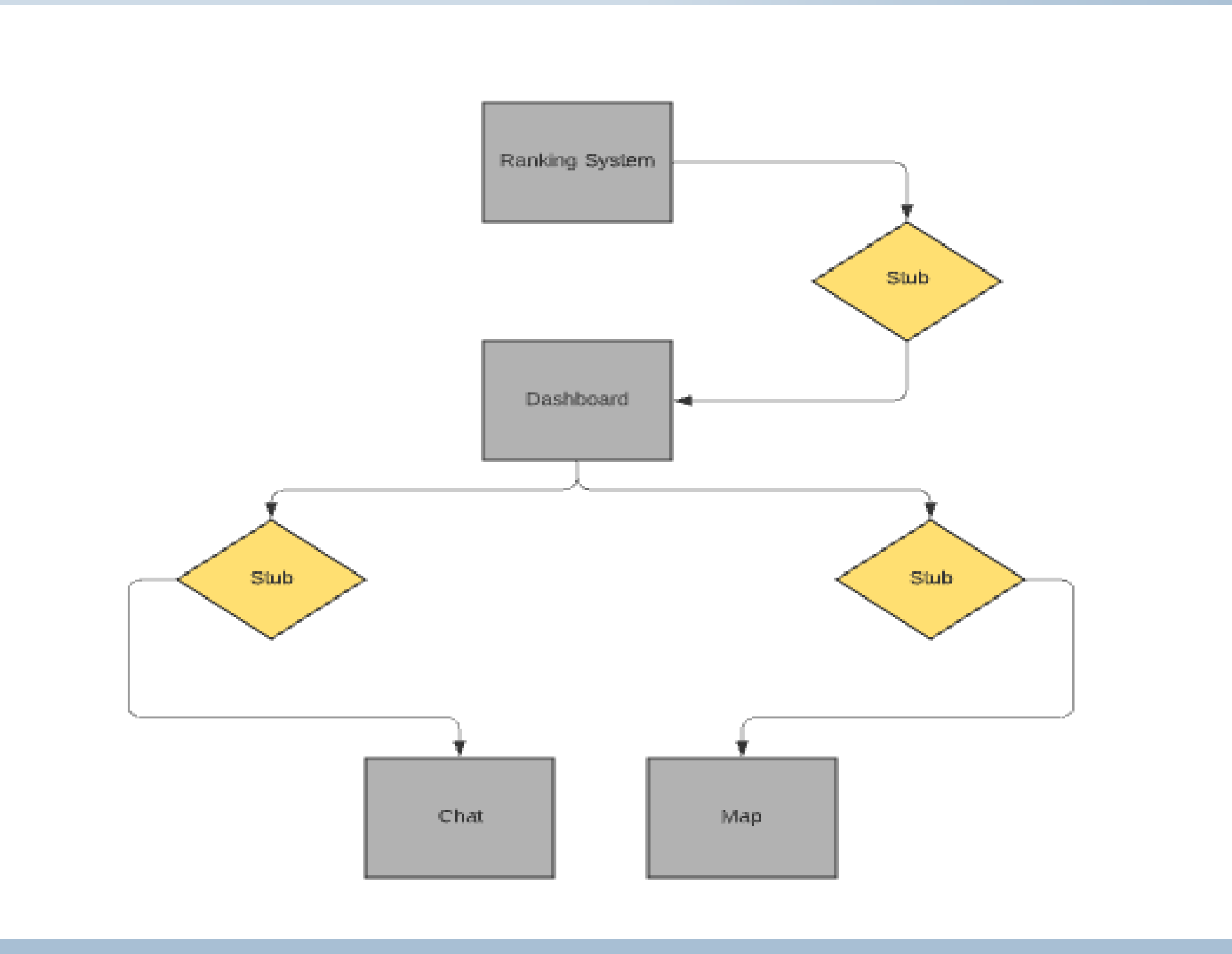
TESTING

```

[9.5, 4.25]
[2, 3]
[{'id': 2, 'rating': 9.5}, {'id': 3, 'rating': 4.25}]
POST OPTIONS
[{'id': 2, 'rating': 9.5}, {'id': 3, 'rating': 4.25}]
DETAILS2
[<flask_sqlalchemy.BaseQuery object at 0x7f5ff41ae0b8>, <flask_sqlalchemy.BaseQuery object at 0x7f5ff4171710>]
[]
Friends List
[<flask_sqlalchemy.BaseQuery object at 0x7f5ff417f710>, <flask_sqlalchemy.BaseQuery object at 0x7f5ff4197588>]
127.0.0.1 - - [09/Apr/2021 21:41:53] "POST /connections/ HTTP/1.1" 200 -
127.0.0.1 - - [09/Apr/2021 21:41:53] "GET /socket.io/?EIO=3&transport=polling&t=NVvg3l3 HTTP/1.1" 200 -
127.0.0.1 - - [09/Apr/2021 21:42:16] "POST /socket.io/?EIO=3&transport=polling&t=NVvg9F_8sid=14fe2b1d43194a939be9524c84d
bF9ab HTTP/1.1" 200 -
127.0.0.1 - - [09/Apr/2021 21:42:16] "GET /socket.io/?EIO=3&transport=polling&t=NVvg2w4&sid=14fe2b1d43194a939be9524c84d
bF9ab HTTP/1.1" 200 -
127.0.0.1 - - [09/Apr/2021 21:42:18] "POST /socket.io/?EIO=3&transport=polling&t=NVvg9sQ&sid=8ac11091e3b74e3a87cfaad3fbb
  
```

Due to the sequential nature of the product, our modules were tested in a top down incremental fashion. This allowed us to work on multiple features at along with isolating, locating, and resolving any issues. Interdependent features, such as the ranking system, took precedence and required more time and effort. Our testing workflow is three simple steps:

- Develop module
- Create stub (Classes Objects) as a means to test functionality
- Ran multiple scenarios against stub



We also utilized a little bit of white box testing techniques by verifying the output of every major coding statement, conditional loops, and functions