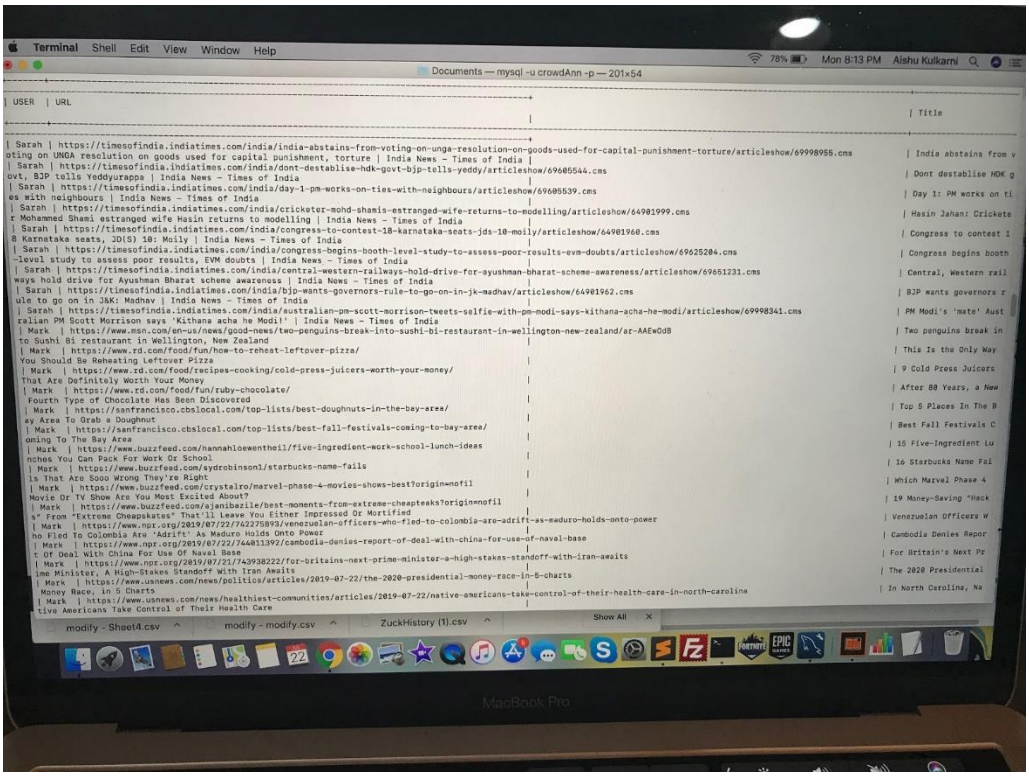
System and Unit Test Report
Crowdsourced Anonymity
Crowdsourced Anonymity
07/20/19

System and Unit Test Report

System Test scenarios:

A. User Story 1 from Sprint 1: As a user, I want to be able to have my data stored safely and efficiently in a database so that my privacy is ensured.

Scenario:



After Running the DB code to insert our csv values and when we check what information is stored in the DB by running a Select Statement we can see that information is being stored properly into the db.

B. User Story 2 from Sprint 1: As a user, I want to be able to ban certain websites from being used so that I can choose what websites and what data I feel comfortable sharing.

Scenario:

1. Run Danger.py which takes the csv file.
2. It ultimately parses every url that gets sent through, and then checks whether a specific url is banned or not banned.
3. If a specific url is not banned, it goes to the next one, however if the url the parser comes across is banned then it becomes removed.

4. The end result will show the user a result on the terminal with all the unbanned files with their url's as is and the banned files will be erased.

C. User Story 1 from Sprint 2: As a user, I want a tcp server so I know that files are able to get sent from the client to the server in a safe manner.

Scenario:

1. Start both the Client and Server programs, and send files to both of them.
2. Check to make sure that each part of the server is able to receive and accept files from each other.
3. User should see that their files are being successfully transported without losing any critical files.

D: User Story 2 from Sprint 2: As a user, I want to make sure that the database collecting the histories of random users gets randomized correctly so i don't get targeted ads.

Scenario:

```
Aishus-MacBook-Pro:Documents aichu$ python3 groupings.py
[[('Loki',), ('Carol',), ('Sarah',), ('Anna',)], [('Mark',)]]
Aishus-MacBook-Pro:Documents aichu$ python3 groupings.py
[[('Sarah',), ('Loki',), ('Mark',), ('Carol',)], [('Anna',)]]
Aishus-MacBook-Pro:Documents aichu$ python3 groupings.py
[[('Loki',), ('Sarah',), ('Carol',), ('Anna',)], [('Mark',)]]
Aishus-MacBook-Pro:Documents aichu$ python3 groupings.py
[[('Mark',), ('Carol',), ('Sarah',), ('Anna',)], [('Loki',)]]
```

1. Open the database, and open the csv file consisting of user histories.
2. Store the data into a list, remove the elements that are greater than four elements, and then randomize this list.
3. User should see that they are receiving information from different users that is totally irrelevant.

E: User Story 1 from Sprint 3: As a user, I want to make sure that the files in the csv program is automatically able to open files on my browser to ensure that those are indeed the links I am visiting.

Scenario:

1. Open up the CSV file and let the program automatically run every url and open up a new link.
2. The user can now verify that these are indeed the sites they have been visiting.