**Team Members:**

Angela Steinmetz, Anahita Ahmadi, Aayushi Agrawal, Siddharth Bansal, Sanjana Ilango, Anisha Reddy, Logan Norman, Ramneek Singh

White Box Testing

* Test Case: Test the login functionality by specifying a username that exists and a password that doesn’t match. The user should be taken back to the login page with an invalid login message.
* Test Case: Test the login functionality by specifying a username that doesn’t exist. The user should be taken back to the login page with an invalid login message.
* Test Case: Test the login functionality by specifying a username that exists and a password that exists and a password that matches. The user should be successfully logged in and taken to their profile page.
* Test Case: Test the create account functionality by specifying a username that already exists in the database. This should produce an error saying that the email is already registered
* Test Case: Test the create account functionality by specifying an email that doesn’t yet have an account, and have the correct password and check password boxes. This should log in the users and bring them to the profile page.
* Test Case: Test the create account functionality by specifying an email that doesn’t yet have an account, but non-matching passwords in the password and check password fields. This should create an error, saying the passwords don’t match
* Test Case: For the form, if not all fields have an input, and if the user tries to submit the form, produce an error message and highlight the questions that have yet to be filled.
* Test Case: For the form, if all the fields have an input, and the user submits the form, have it be successful with a success message displayed
* Test Case: A new user signs up, should be prompted to submit the desired qualities form
* Test Case: For the profile page, if not all the fields have an input,and the user tries to save their data, then throw an error message.
* Test Case: For the matching, if both users accept, then the match stays and they don’t get a new matching as long as both remain as acceptances
* Test Case: For the matching, if only one or neither user accepts the matching, then the next day they will be given a new, different matching.

Black Box Testing

* Test Case: Testing invalid inputs to the algorithm. It’s not necessary to look at the code to know that the ranking is invalid
* Test Case: Run the program with duplicated users
* Test Case: Run the program with empty input/missing parameters
* Test Case: Run program with inputs in a different order and it should produce the same output
* Test Case: Having multiple users and making sure that the correct matchings are made (no false matchings)
* Test Case: For the profile picture, if the user tries to upload a picture of the incorrect format, throw an error.
* Test Case: Testing to make sure the designated matching based on the algorithm is the matching displayed on the match page
* Test Case: Testing to ensure all information displayed on the match page is from the same user
* Test Case: Make sure the header where users are able to go to different pages accurately brings the user to the page they want to go to.

Unit Testing

* Test Case: Insert SQL code in the username variable of the UserAuthentication.authenticate() method. Verify that the authentication method is accurate. Verify that the SQL code specified is not executed against the database.
* Test Case: Test to ensure that the algorithm operates as desired. Run the algorithm with various inputs to confirm that it produces optimal results in each case, and modify the algorithm if necessary.
* Test Case: Check the case if there’s only one user. This should have the user be matched with no one, and the match page shouldn’t display anyone
* Test Case: Check the case where there are only two users, who could potentially be attracted to each other. The algorithm should pair them up initially. If they reject the match, have the algorithm run again and ensure that they don’t get paired up again.

Regression Testing

* Test Case: After first only allowing users to rank people they know via the form and the matched output is determined to be accurate, then even after the other questions in the form about personal preferences are used in the matching process, testing to make sure that people are still accurately matched with those that they know
* Test Case: After making the REST API live, ensure that the front-end queries return the correct information for each of the users
* Test Case: After deploying all software to a cloud computing provider, testing functionality above