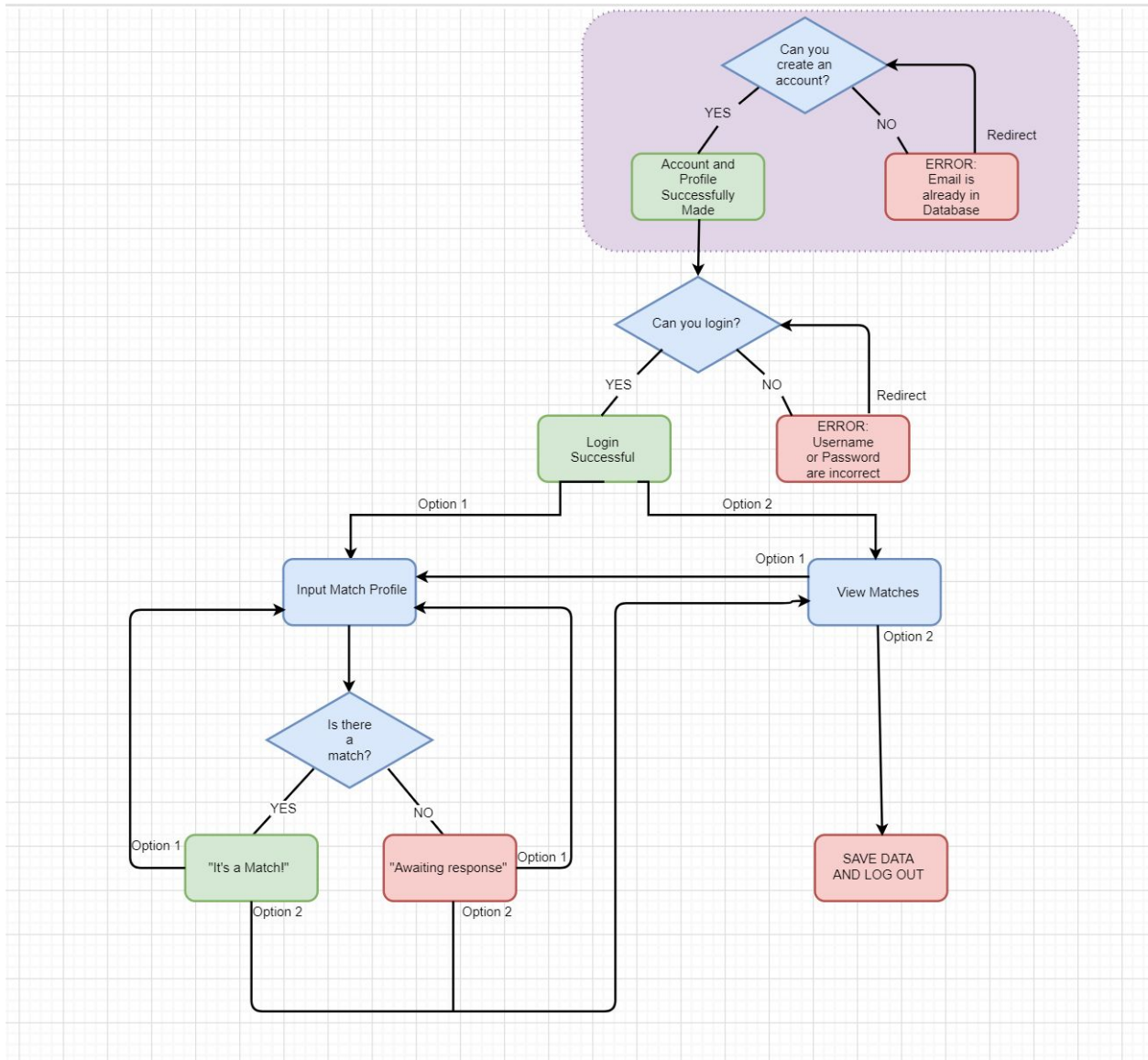


User Acceptance Test Plans: Create a document that describes how at least three features within your finished product will be tested. The test plan should include specific test cases (user acceptance test cases) that describe the data and the user activity that will be executed in order to verify proper functionality of the feature.

Automated Test Cases: Provide a link to the tool you use to automate testing, or explain how to run the automated test cases or schedule time with the TAs to demonstrate your automated tests. Provide a copy of the output showing the results of the automated test cases running.

- features tested:
 - creating an account
 - creating a profile
 - matches confirmed
 - matches denied
 - sending emails
 - repeating process searching for another match
 - log out
 - look at matches
 - quit before submission
- test case 1:
 - create an account
 - test if username, password, and email data are placed in the Logins table
 - create self profile for specific match
 - test if all self information is placed in the Self Information table
 - test if proceeds to find a match page
 - find a match
 - test if all searching information is placed in the Searching For table
 - match confirmed
 - test if the match is found
 - email sent by program
 - test if matched users are sent emails
 - ask user if they want to go through the process again
 - test linking to repeat process
 - user selects yes and enters information again
 - tests ability to enter multiple sets of data into the database from one user
 - log out
 - tests logging out and closing the program from the point where the program asks the user if they want to go through the process again
- test case 2:
 - log in
 - test if username and password is found in the Logins table and proceed to profile page
 - create self

- test linking to self information input page from profile page
 - test if all self information is placed in the Self Information table
 - test if program proceeds to find a match page
 - find a match
 - test if all searching information is placed in the Searching For table
 - no match confirmed
 - test response when no match is found
 - ask user if they want to go through the process again
 - test linking to repeat process
 - user selects no
 - test if program routes back to profile page if user selects no
 - log out
 - test logging out from profile page
 - quit
- test case 3:
 - log in
 - test if username and password are found in the database and proceed to profile page
 - look at matches
 - test proceeding to look at matches page from profile page
 - log out
 - test logging out and closing program from look at matches page
- test case 4:
 - log in
 - test if username and password are found in the database and proceed to profile page
 - create self
 - test linking to self information input page from profile page
 - test if all self information is placed in the Self Information table
 - test if program proceeds to find a match page
 - find a match
 - test if any data entered is placed in the Searching For table
 - user quits before submission of find a match
 - test deletion of any logged information in both Searching for and Self Information tables
 - log out
 - test logging out and closing program from searching for page



HOW TO RUN TEST FILES:

For our automated tests we used a test suite called Selenium, which basically allows you to record your mouse and keyboard actions within a website. In other words it emulates a users actions that were prerecorded. The file type produced a .side file. To begin using Selenium download the browser extension Selenium IDE found on Selenium homepage. Next, set up our test database MakeDB.sql which can be found in the makedatabase folder in our github directory. Then start of MySQL server and your local Web Server that can run PHP. Navigate to the homepage.php page and open the Selenium extension from the browser application bar. Select the open project button in the upper

right hand corner of Selenium application and open AutomatedTests.side. Make sure that each tests target matches the directory your local server is in if not change it. Proceed to run each test, and actions should be emulated on the webpage the name of each test is representative of the goal we want and the running of the test will show us the result of the given condition.