INFO 3235: Software Quality Assurance (S50)

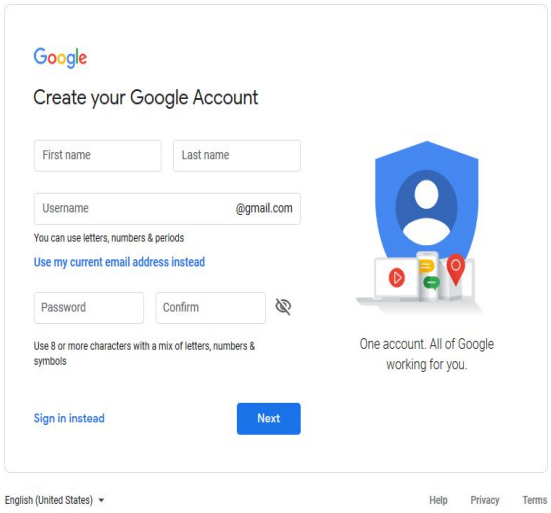
Team Project

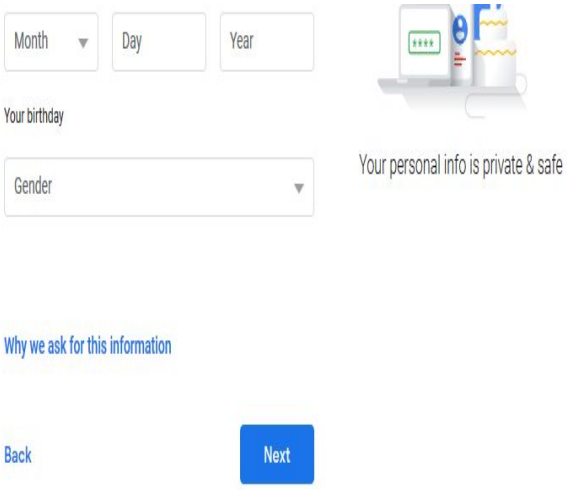
Matthew Finlayson: 100356255
Reke Ogbokor: 100276246
Ryan Carr: 100336211

Table of Contents

Research

For our project, we will be creating a Google Account Creator. Our prototype would authenticate a new user's inputs such as their full name, username, password, age, and location and then enable them to log into their accounts. Each user's input will be individually tested. For example; the user's full name can't be "Matt56_", as the name can't have numbers in it. The user's username can't be just "m", it has to have more than 1 character. The user must provide an age greater than 18. The user must input a real location; like the name, it can't have any numbers. And the user must create a password, with the correct credentials, the password can't be "1234". The password must have 8 characters, one digit, one lowercase, one uppercase, one special character and does not contain space or tabs. When the user is putting in their input, if they place an invalid input they will need to restart the whole process. After the user enters valid information, the program will show all their entered data. The program will autofill the username to have @gmail.com at the end of it. The user will be told to press enter to exit the account creator.





Explanations

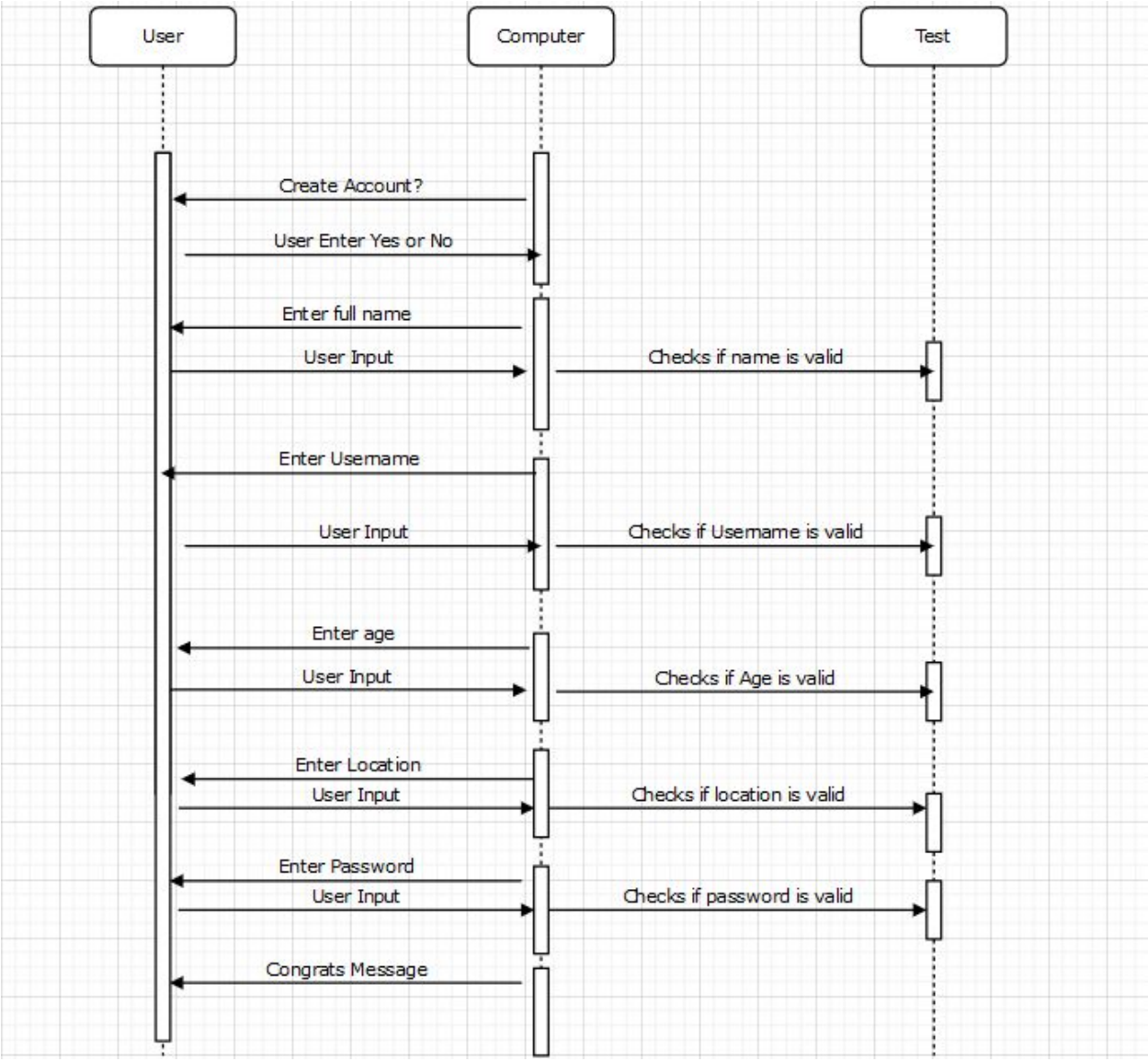
In this part, I will explain where everything is located in our project. To locate our UML diagrams, scroll down this report you're currently on and all the diagrams will be there. To locate where we did FSM testing, it will be in the folder named "FSMTesting". In that folder, it will have the java code for our testing and also two .png files showing two separate test runs. To locate where we did Black Box Testing, it will be in a folder named "BlackBox Testing". In this folder, it will have the java code for our Black Box Testing as well as a Black Box Testing Diagram(Equivalence class partitioning). To locate where we did White Box Testing, it will be in a folder named "WhiteBox Testing". In this folder, it will have the java code for our White Box Testing as well as two White Box Testing Diagrams (Branch Coverage Table and Control Flow).

Google Account Creator Diagrams

Scroll down for the diagrams and a short description of each one

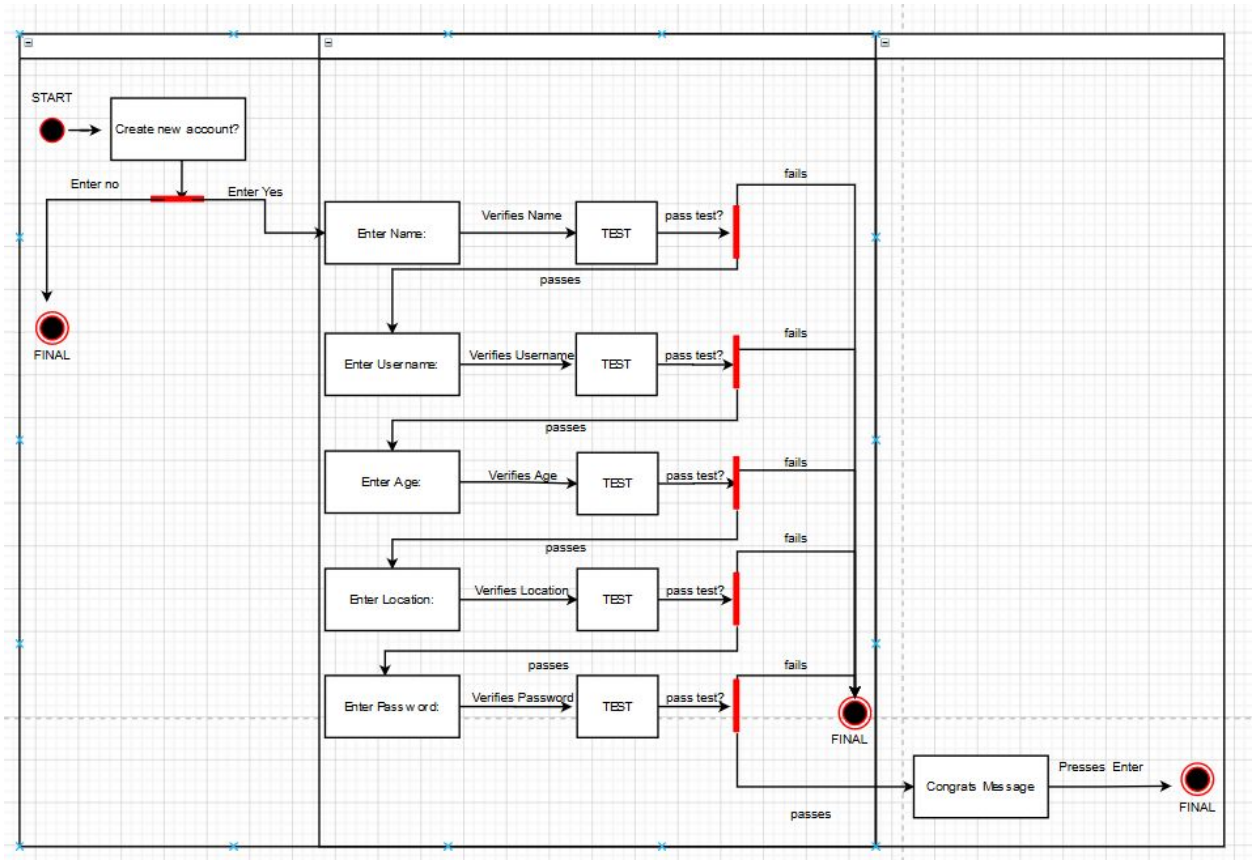
Sequence Diagram

This diagram shows the sequence of the entire program. The diagram has 3 actors. User, computer and test. The user inputs data throughout the program. The computer accepts the input. And the test validates the data. Each time the user enters an input it passes through from the computers and then to the test. This sequence follows all the way up until the Congrats message at the end.



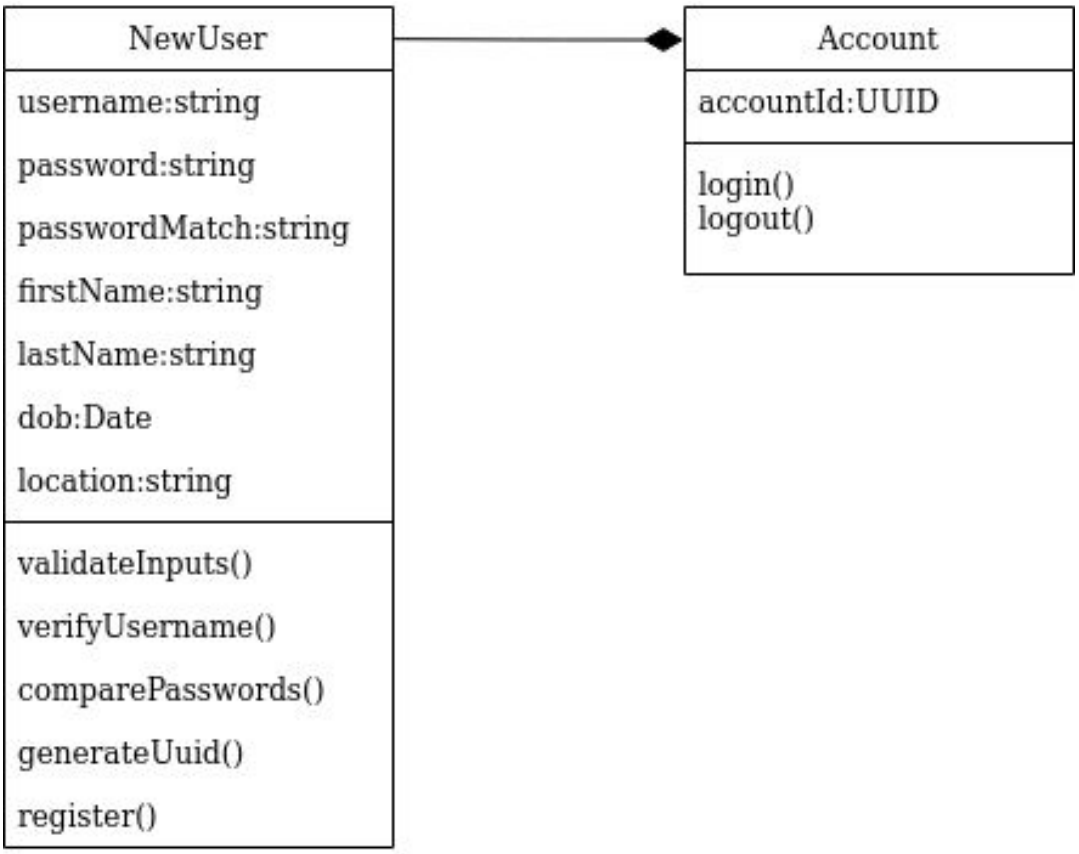
State Diagram

This diagram is created to represent the condition of the system. This state diagram was divided into three sections as there are 3 parts the program goes through. In each section, it is possible for the program to enter the final state. If the user initially chooses to not create an account, the program will enter a final state. If the user enters any invalid data, they will enter the final state. Finally, if the user inputs all valid data, they will enter the final state. After the user decides to create an account, they will have to enter valid data to proceed to the next state. If at any state the data entered is invalid, they will enter the final state.



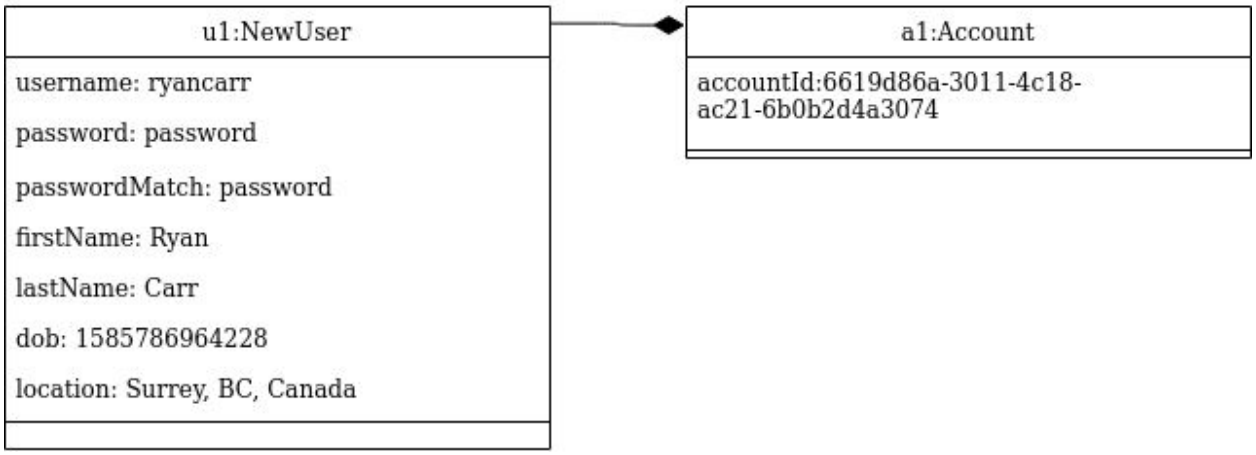
Class Diagram

Our approach to designing this diagram was to create attributes for all of the input data that a user must enter to create their account. There will also be some standard methods for that class that compare if the password and repeat password fields match if the username is in use or not, if there is an account already associated with the phone number, and finally to post the user to the database.



Object Diagram

The object diagram was simply done by filling in the class diagram's attributes with test values according to the data types outlined in the class diagram. As for the names of the objects, since this is a simple diagram, we just used the first letter of the class name and added a number to symbolize that multiple objects could be made and this uniquely identifies them.



Use Case Diagram

In this diagram, we will be modeling the functionality of our system. This diagram has two actors. The user and Test Software. The entire program is functioning within the system computer. The user must complete each action like “Enter Name”, but it must be validated after, that’s why there is an includes relationship after each use case that is connected to the user. The Test Software also must perform the action of validating the input after each action the user takes.