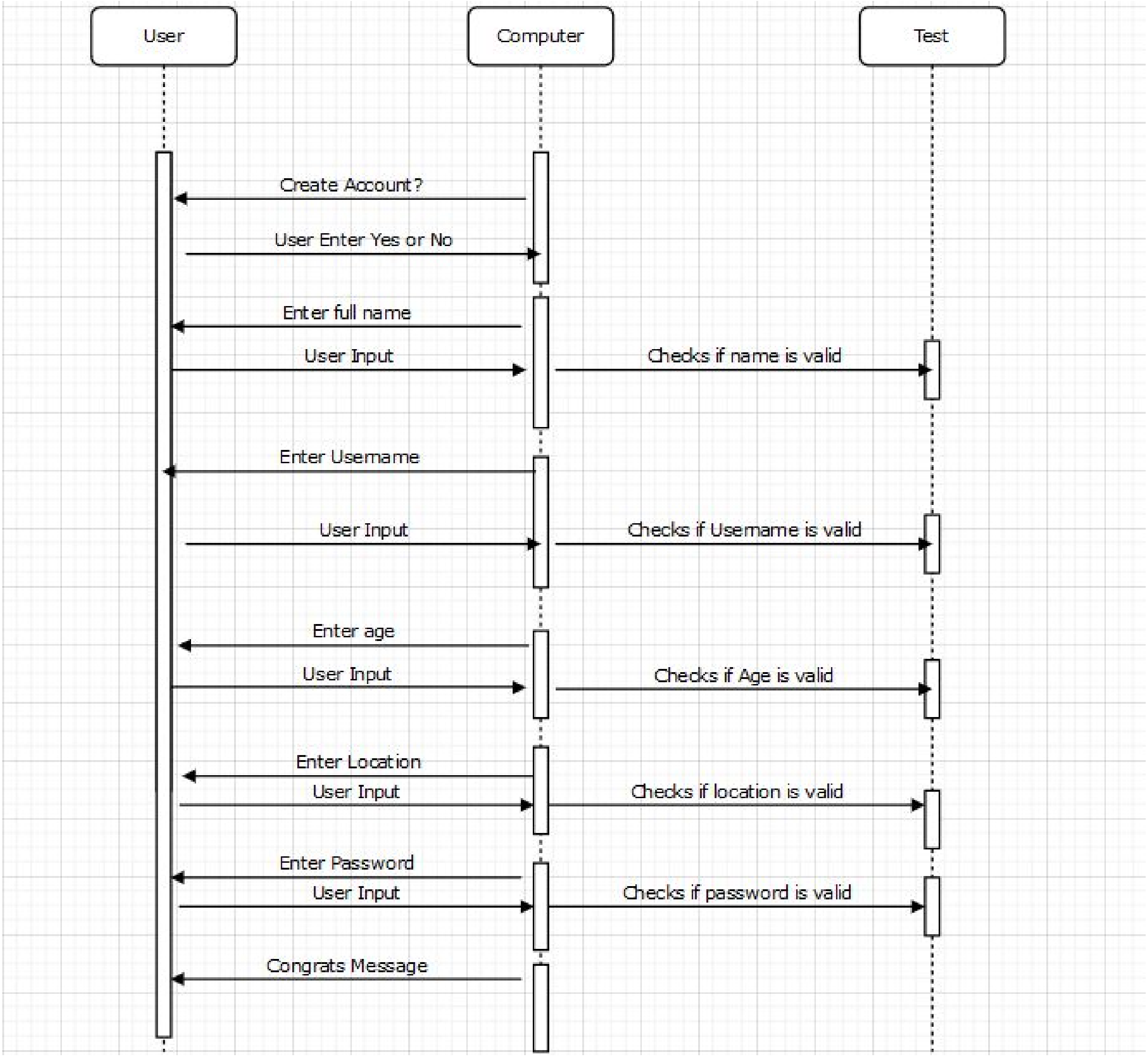
Google Account Creator Diagrams

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

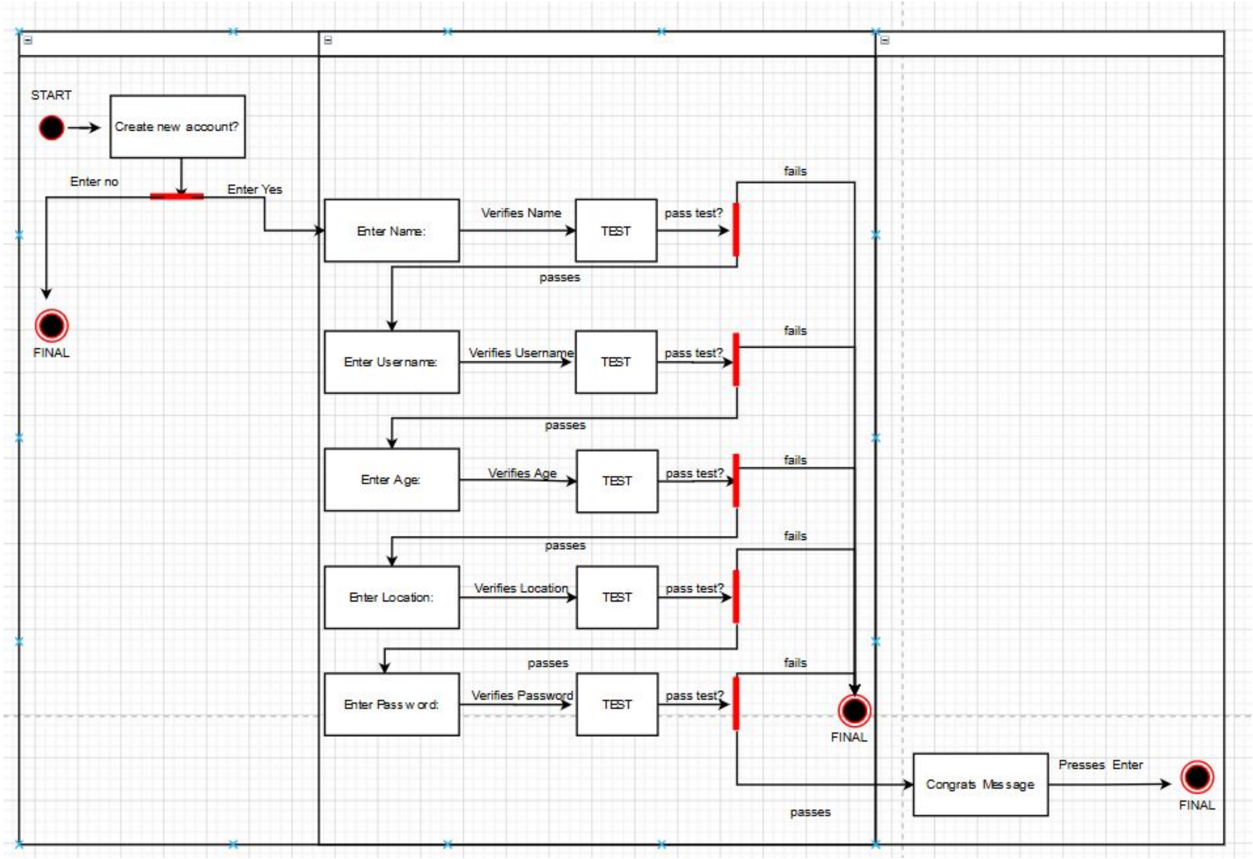
Sequence Diagram

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.



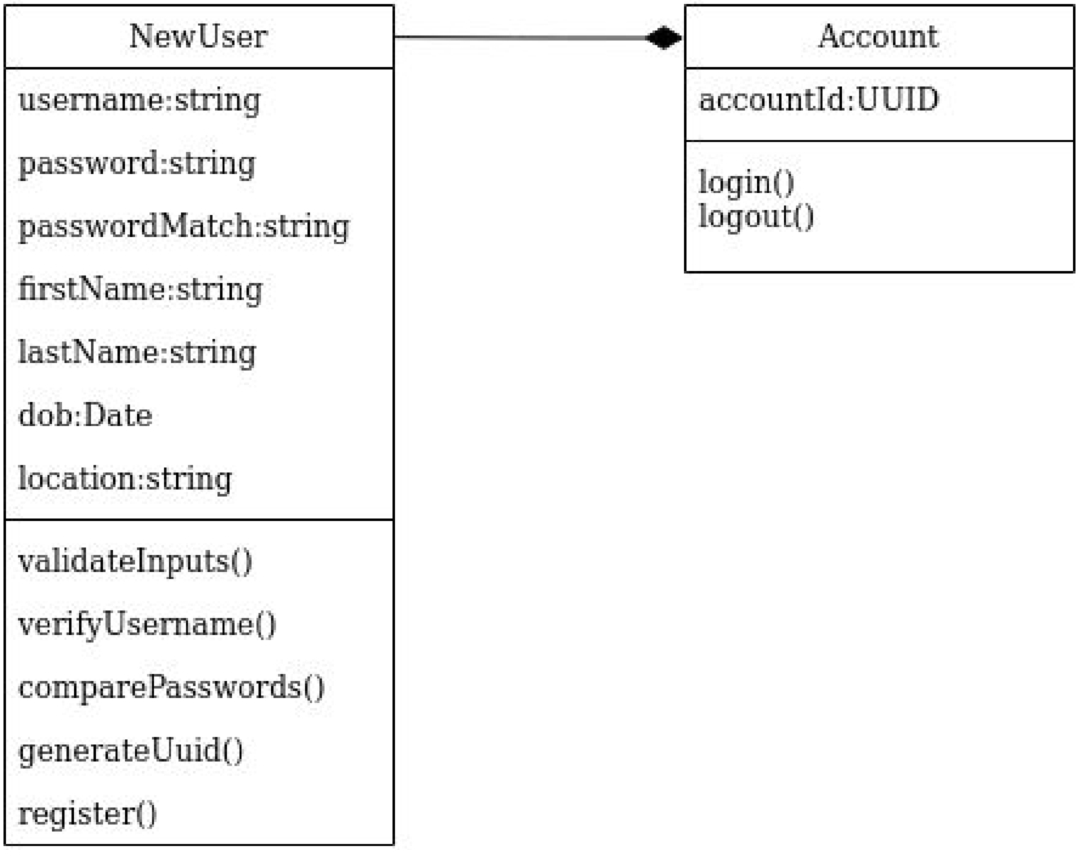
State Diagram

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.



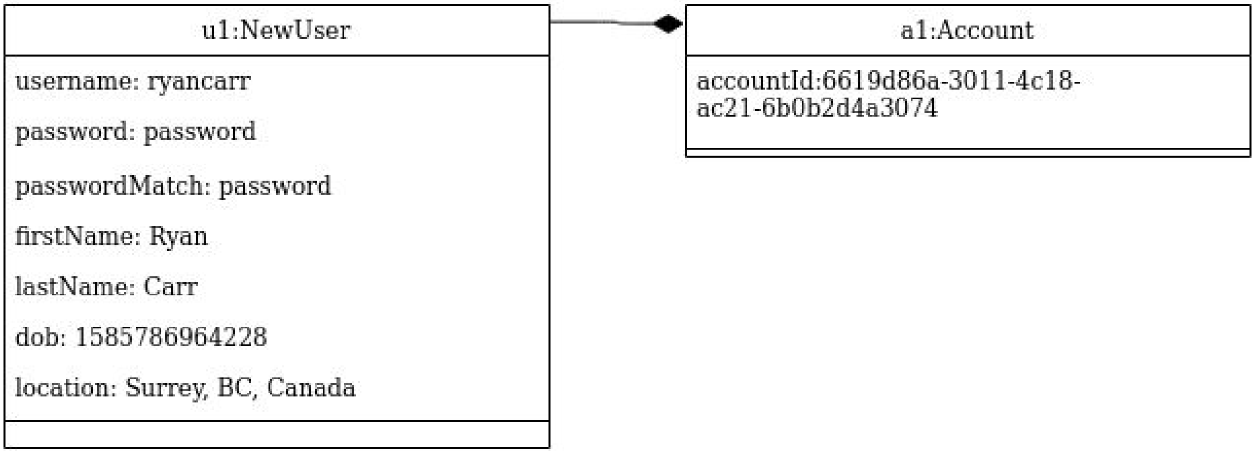
Class Diagram

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.

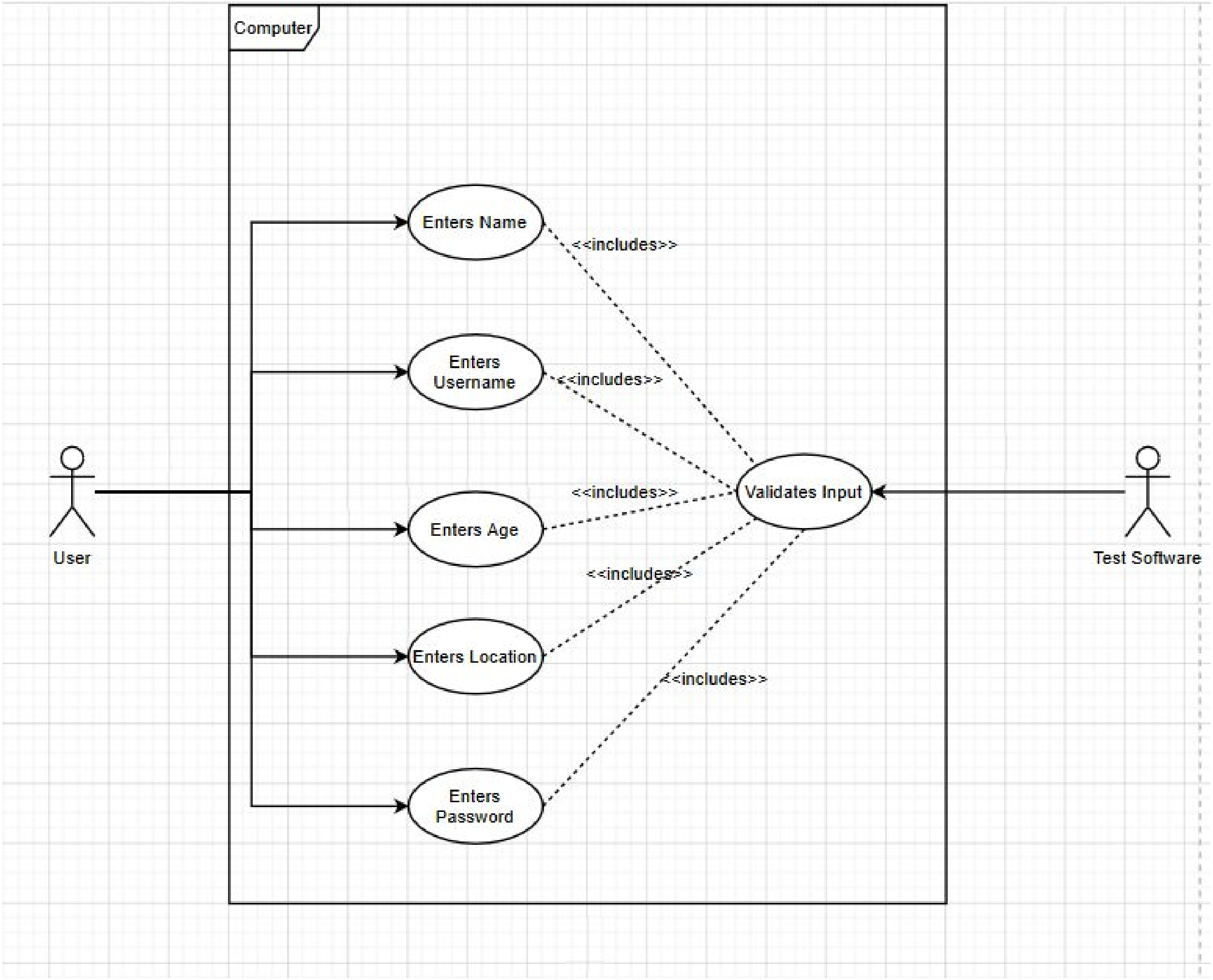


username: ryancarr
password: password
passwordMatch: password
firstName: Ryan
lastName: Carr
dob: 1585786964228
location: Surrey, BC, Canada

accountId:6619d86a-3011-4c18-ac21-6b0b2d4a3074

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.


Computer
Enters Name
<<includes>>
Enters Username
<<includes>>
<<includes>>
Validates Input
User
Enters Age
Test Software
<<includes>>
Enters Location
<<includes>>
Enters Password