|  |
| --- |
| myMathTutorLogin  *GetAccount* |
| -<User> sever |
| -getAccount();  -login();  -signUp();  -initilize(); |

|  |
| --- |
| myMathTutorProblem  *GetAccount* |
| -<Problem> currentList |
| -getAccount  -Initizilze  -Submit  -Next  -Preveious  -displayHint |

|  |
| --- |
| myMathTutorWelcome  *GetAccount* |
| -<User> currentUser; |
| -Initialize();  -getAccount();  -continueLastAction(); |

|  |
| --- |
| myMathTutorAdmin  *GetAccount* |
| -<User> server |
| -removeUser  -addUser  -editUser  -updateServer |

User wants to sign up for an account

Loads Randomly generated problems with Problem GUI

Loads Lesson Object with Problem GUI

User clicks logout

User clicks Take Lessons

Serializes account and pulls up login GUI

Checks to see what action the user did last session

Access and loads necessary assets for quizzes

Access and loads necessary assets for lessons

User clicks continue last action

Welcomes user and asks what they would like to do via buttons

User clicks Take Quiz

Loads assets

(Stats, buttons, graphics, etc)

If user isn’t found

Program throws exception if their account isn’t found, or their password or username is incorrect

Displays alert

Saves retrieved user for loading into the next GUI

Program pulls up the welcome GUI

Program retrieves user with matching records

Program searches user arraylist using user’s submission as parameters

User is asked to login using their username and password

User fills in their information using Sign Up GUI

Validates and saves user information as a new User to the arrayList of Users

User is asked to complete a quiz

After completion, programm displays grade, and displays stats of that lesson

User is asked to complete a practice quiz

User is taken through a lesson (depending on selection)

User will be asked to complete a set of 10 problems

After completion of quiz. Program will display score, update users stats.

User is returned to the Welcom GUI