Blue=Katelyn Green = Kevin

* 1. Requirements:
     1. Products to include:
        1. Log in to run the program including a hint and a certain many tries before the program shuts down.
           1. If the user is a first-time user, allow them to create a profile.

Profile requirements is: username, password (password requirements in item h), first and last name

Once the user registers, require them to log in

* + - 1. Method to take in user input to store Account(username,password) based on categories.
         1. Accounts must be able to be pulled based on categories.
      2. Save information to a text file.
      3. Be able to navigate through the categories the user used
         1. Examples of categories: home, work, entertainment, bills, etc.
         2. Should be able to print out the categories in a list.
      4. Password generator option if the user requests. (This is the same password generator from our homework assignment)
         1. Half the points, include this in a method
         2. All the points, include this as a class
      5. Be able to delete or modify passwords.
      6. Output needs to be simple to understand:
         1. “The account: {accountName}  
             The username: {username}  
             The password: {password}”
      7. Need to make sure that user input and the generated passwords meet the requirements of the following:
         1. 1 capital character
         2. 1 number
         3. 1 special character (There are only certain special characters allowed)
         4. At least 8 characters long
         5. Half of the points, create a method to check for this
         6. All the points, create a class to check for this
      8. 10% Bonus: save the file as a password protected or encrypted file
      9. 5% Bonus: create a method that will check how long to brute force break it
      10. 15% Bonus: save the info on a database
    1. Computer Science concepts to include
       1. User Input
       2. Arrays, Lists, and ArrayLists
       3. Utilizing built in functions to create the passwords and not a list of characters
       4. Loops
       5. ALL methods are commented
       6. Multiple Classes
          1. (Suggestions: PasswordRecord, Library, OpenCloseFile, etc)
  1. The user should **not** be able to create a runtime error.
  2. The usability of the program should be smooth and easy to understand.
  3. Remember that each section of code needs to be commented, and any outside sources need to be documented. In the documentation, you must include commentary on what you are using.
  4. MUST HAVE a design overview approved by Bander before beginning this project. In that design overview Bander needs to see the features, details on how to implement the features, and who is doing which features.

red=done

* 1. Requirements:
     1. Products to include:
        1. Log in to run the program including a hint and a certain many tries before the program shuts down.
           1. If the user is a first-time user, allow them to create a profile.

Profile requirements is: username, password (password requirements in item h), first and last name

Once the user registers, require them to log in

* + - 1. Method to take in user input to store Account(username,password) based on categories.
         1. Accounts must be able to be pulled based on categories.
      2. Save information to a text file.
      3. Be able to navigate through the categories the user used
         1. Examples of categories: home, work, entertainment, bills, etc.
         2. Should be able to print out the categories in a list.
      4. Password generator option if the user requests. (This is the same password generator from our homework assignment)
         1. Half the points, include this in a method
         2. All the points, include this as a class
      5. Be able to delete or modify passwords.
      6. Output needs to be simple to understand:
         1. “The account: {accountName}  
             The username: {username}  
             The password: {password}”
      7. Need to make sure that user input and the generated passwords meet the requirements of the following:
         1. 1 capital character
         2. 1 number
         3. 1 special character (There are only certain special characters allowed)
         4. At least 8 characters long
         5. Half of the points, create a method to check for this
         6. All the points, create a class to check for this
      8. 10% Bonus: save the file as a password protected or encrypted file
      9. 5% Bonus: create a method that will check how long to brute force break it
      10. 15% Bonus: save the info on a database
    1. Computer Science concepts to include
       1. User Input
       2. Arrays, Lists, and ArrayLists
       3. Utilizing built in functions to create the passwords and not a list of characters
       4. Loops
       5. ALL methods are commented
       6. Multiple Classes
          1. (Suggestions: PasswordRecord, Library, OpenCloseFile, etc)
  1. The user should **not** be able to create a runtime error.
  2. The usability of the program should be smooth and easy to understand.
  3. Remember that each section of code needs to be commented, and any outside sources need to be documented. In the documentation, you must include commentary on what you are using.
  4. MUST HAVE a design overview approved by Bander before beginning this project. In that design overview Bander needs to see the features, details on how to implement the features, and who is doing which features.