Project Design

Project Topic: Health Monitoring System

Audrey Harcum

Blake Green

Brandon Sroufe

Todd Weber

Revision 1.5

CMSC 495 6380

15 September 2020

Group 4

# Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| Revision Number | Date | Description | Name |
| 1.0 | 9/8/2020 | First draft created by Blake | Blake |
| 1.1 | 9/10/2020 | Added pseudocode to class prototypes | Todd |
| 1.2 | 9/10/2020 | UPDATE: Added seven potential scenarios and individual event-trace diagrams for each. | Audrey |
| 1.3 | 9/11/2020 | UPDATE: Entire team contributed during Zoom collaboration meeting | Team |
| 1.4 | 9/12/2020 | UPDATE: Class Diagram and document formatting | Brandon |
| 1.5 | 9/15/2020 | UPDATE: Final Review & Submit to Classroom | Blake |

Table of Contents

[Revision History 2](#_Toc50808120)

[Design 5](#_Toc50808121)

[Class Diagram 5](#_Toc50808122)

[Event-Trace Diagram 6](#_Toc50808123)

[a. Scenario 1 - Create Profile -> Successful 6](#_Toc50808124)

[b. Scenario 2 - Create Profile -> Unsuccessful 7](#_Toc50808125)

[c. Scenario 3 – Log On -> Successful 8](#_Toc50808126)

[d. Scenario 4 – Log On -> Unsuccessful 9](#_Toc50808127)

[e. Scenario 5 – Successful Log On -> Delete User 10](#_Toc50808128)

[f. Scenario 6 – Successful Log On -> Log Off 11](#_Toc50808129)

[g. Scenario 7 – Successful Log On -> Update GOAL / Biometrics 12](#_Toc50808130)

[Class Design 13](#_Toc50808131)

[h. Input/Main subsystem: 13](#_Toc50808132)

[i. Login subsystem: 13](#_Toc50808133)

[j. Calculation subsystem: 14](#_Toc50808134)

[l. Display/Input subsystem: 15](#_Toc50808135)

[m. Database Interface 15](#_Toc50808136)

[Unresolved Risks/Mitigation 17](#_Toc50808137)

**List of Figures**

[Figure 1 – Class Diagram 5](#_Toc50808138)

[Figure 2 – Create Profile -> Successful 6](#_Toc50808139)

[Figure 3 – Create Profile -> Unsuccessful 7](#_Toc50808140)

[Figure 4- Log On -> Successful 8](#_Toc50808141)

[Figure 5 – Log On -> Unsuccessful 9](#_Toc50808142)

[Figure 6 – Successful Log On -> Delete User 10](#_Toc50808143)

[Figure 7 – Successful Log On -> Log Off 11](#_Toc50808144)

[Figure 8 – Successful Log On -> Update Goal / Biometrics 12](#_Toc50808145)

# Design

This is the Project Design document for Group 4’s Health Monitoring System project.

# Class Diagram

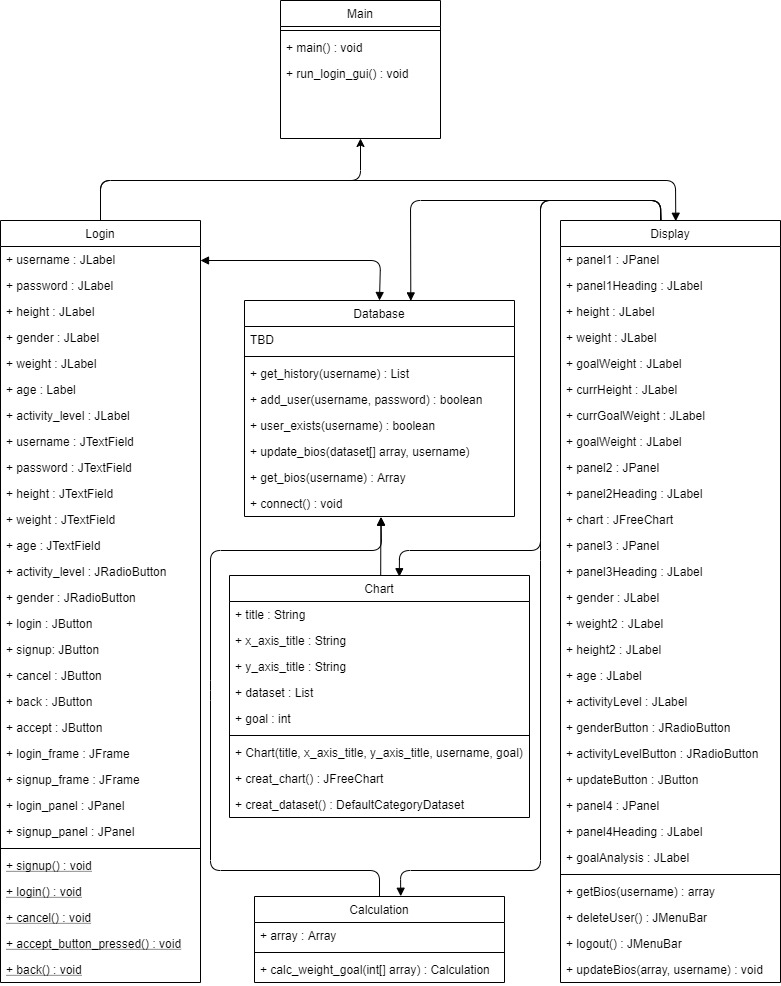


Figure 1 – Class Diagram

# Event-Trace Diagram

## **Scenario 1 - Create Profile -> Successful**

Description: The user runs the program from system desktop. They are prompted to enter username and password, and have the option to “Create Profile”, “Login”, or “Cancel”. The user chooses to create profile.

Precondition: The program is awake, local database is waiting to be queried/written to.

Post-condition: A new profile is created; a new user entry is written to the database. User is able to see and interact with main UI.

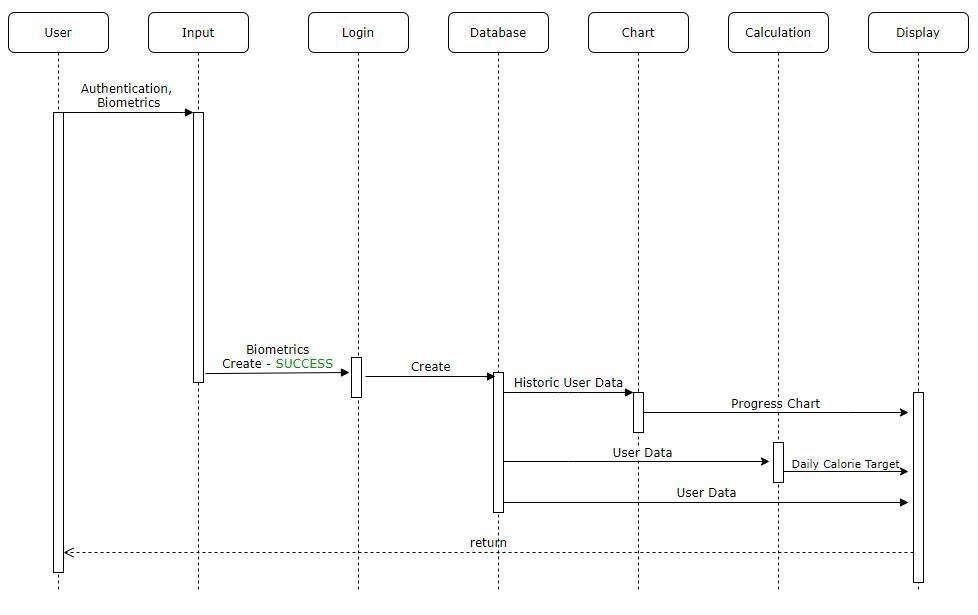


Figure 2 – Create Profile -> Successful

## **Scenario 2 - Create Profile -> Unsuccessful**

Description: The user runs the program from system desktop. They are prompted to enter username and password, and have the option to “Create Profile”, “Login”, or “Cancel”. The user chooses to create profile.

Precondition: The program is awake, local database is waiting to be queried/written to.

Post-condition: A new profile is not created. Error is displayed to the user.

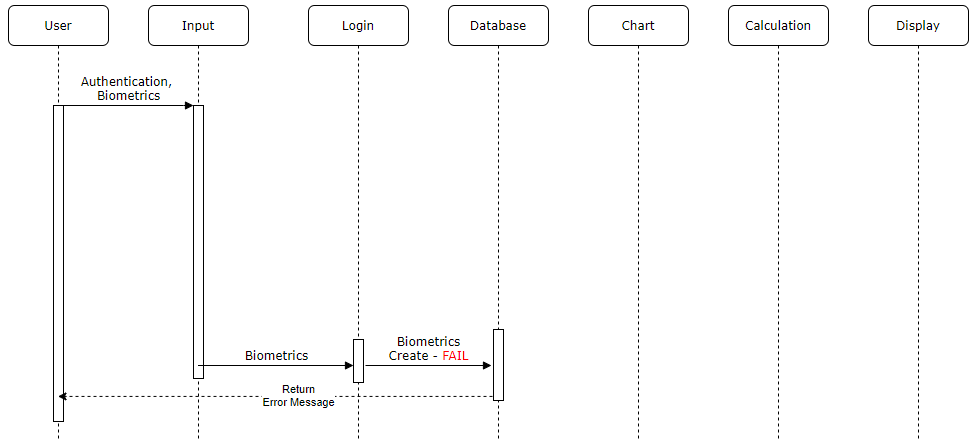


Figure 3 – Create Profile -> Unsuccessful

## **Scenario 3 – Log On -> Successful**

Description: The user runs the program from system desktop. They are prompted to enter username and password, and have the option to “Create Profile”, “Login”, or “Cancel”. The user enters valid credentials and successfully logs on.

Precondition: The program is awake, local database is waiting to be queried/written to.

Post-condition: The user’s credentials are a valid entry in the local database, user is confirmed (authenticated) and user is able to see and interact with main UI.

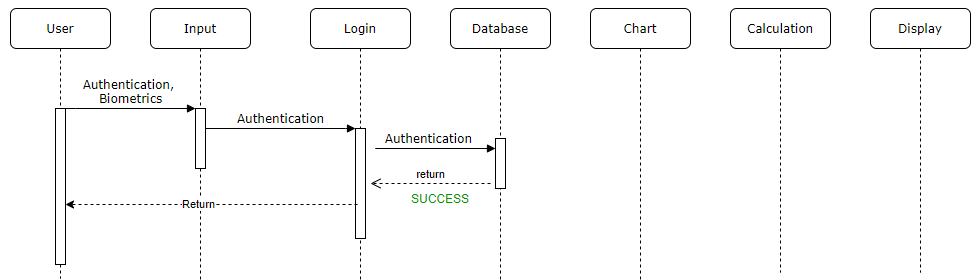


Figure 4- Log On -> Successful

## **Scenario 4 – Log On -> Unsuccessful**

Description: The user runs the program from system desktop. They are prompted to enter username and password, and have the option to “Create Profile”, “Login”, or “Cancel”. The user enters invalid credentials and is unauthorized access.

Precondition: The program is awake, local database is waiting to be queried/written to.

Post-condition: The user’s credentials are not a valid entry in the local database, user is denied entry, and an error message pop-up alerting the user. The user is unable to navigate past the main login page without valid credentials.

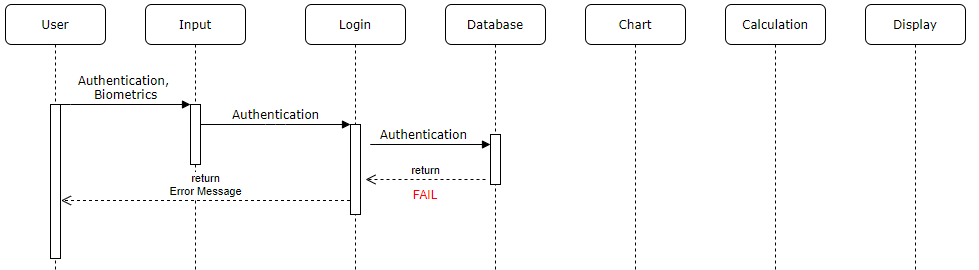


Figure 5 – Log On -> Unsuccessful

## **Scenario 5 – Successful Log On -> Delete User**

Description: The user has successfully authenticated and is able to interact with main UI. The user selects a menu option to either log off or delete their profile.

Precondition: The program is awake, local database is waiting to be queried/written to. The user is authenticated and able to interact with the main UI.

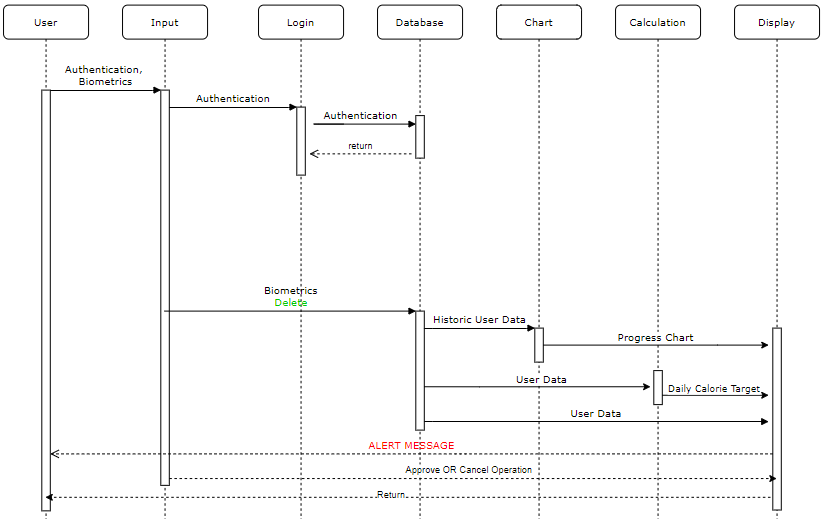
Post-condition: User then navigates to a LOGOFF/Delete profile dropdown menu. The user can select Delete Profile, an ALERT message pops-up asking if they are sure; OKAY to delete CANCEL to cancel operation. If “OKAY” profile is deleted, and a successful alert message notifies user and the user is prompted back to the main login portal; ELSE, user cancels out of current operation and the main UI is present.

Figure 6 – Successful Log On -> Delete User

## **Scenario 6 – Successful Log On -> Log Off**

Description: The user has successfully authenticated and is able to interact with main UI. The user selects a menu option to either log off or delete their profile.

Precondition: The program is awake, local database is waiting to be queried/written to. The user is authenticated and able to interact with the main UI.

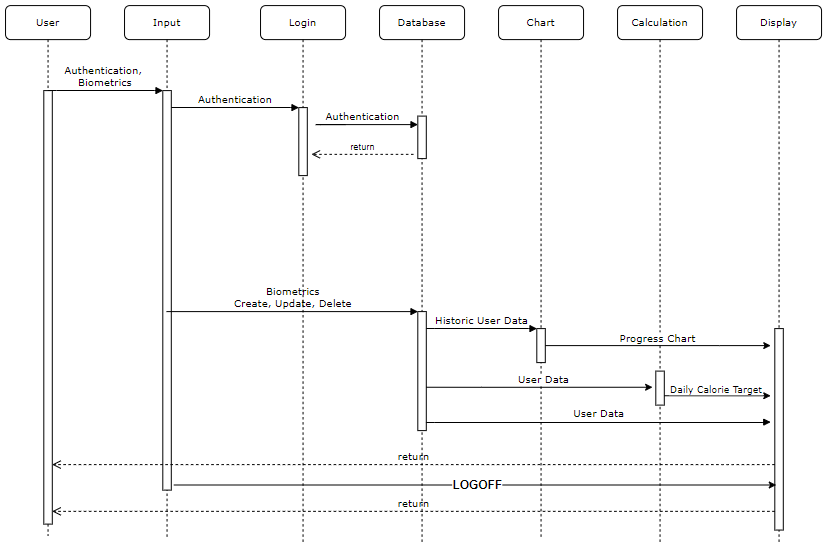
Post-condition: The user then navigates to a LOGOFF/Delete profile dropdown menu. The user selects log off and the user is prompted back to the main login portal.

Figure 7 – Successful Log On -> Log Off

## **Scenario 7 – Successful Log On -> Update GOAL / Biometrics**

Description: The user has successfully authenticated and is able to interact with main UI. The user enters in biometrics and is able to update goals and writes to local database to be saved. User must exercise “Update” button.

Precondition: The program is awake, local database is waiting to be queried/written to. The user is authenticated and able to interact with the main UI.

Post-condition: Table entry is written/appended to the current authenticated user entry.

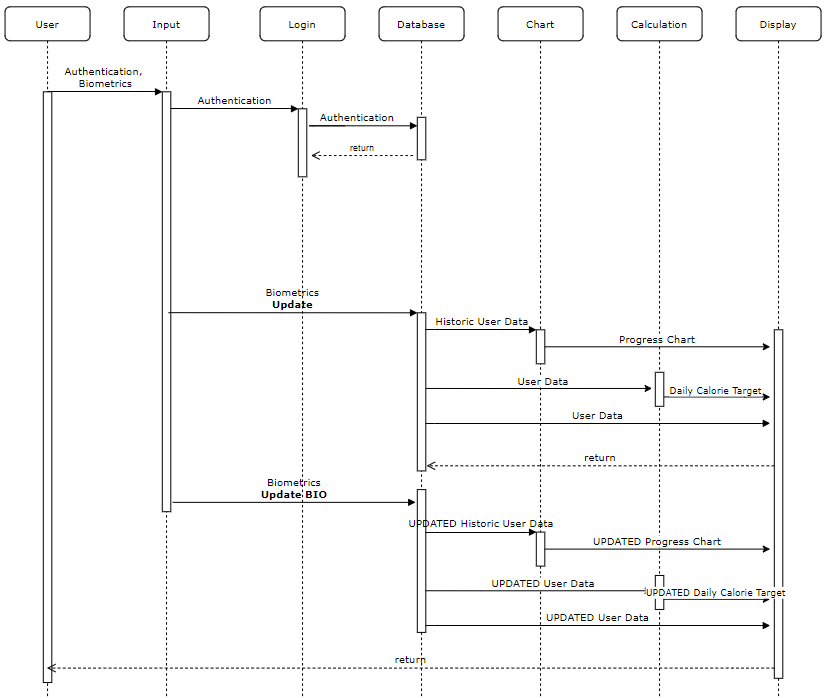


Figure 8 – Successful Log On -> Update Goal / Biometrics

# Class Design

## **Input/Main subsystem:**

Class Main {

Run LoginGui();

}

## **Login subsystem:**

Class Login\_Signup {

JLabel Username, Password, Height, Gender, Weight, Height, Age, Activity Level;

JTextField Username, Password, Height, Weight, Age;

JRadioButton ActivityLevel, Gender

JButton Login, SignUp, Cancel, Back, Accept

JFrame LoginFrame, SignupFrame;

JPanel LoginPanel, SignupPanel

Void SignUp(){

//build frame

SignupFrame.visible = true;

LoginFrame.visible = false;

}

Void Login(){

* 1. Get username from text box
  2. Get password from text box
  3. Call database.authenticate(username, password)
  4. If (true)
     1. Display main GUI
  5. Else
     1. Error message

}

Void Cancel(){

Exit program

}

Void AcceptButtonPressed(){

1. Verify required text fields are populated with appropriate information.
   1. If not filled alert user
2. Build data array
3. Call database.UserExists(Username)
   1. If true alert user
   2. Else
      1. Call database.addUser(username, password)
      2. Call database.UpdateBiometrics(array, username)

}

Void Back(){

SignupFrame.visible = false;

LoginFrame.visible = true;

}

}

## **Calculation subsystem:**

Class Calculation {

Int CalculateWeightGoal(int array)

1. Take dataset from array
2. Feed into calculation
3. Return final Calculation

}

1. **Chart subsystem:**

Class Chart {

String title, xAxisTitle,yAxisTitle;

Int goal

Array dataset

Constructor Chart(title, xAxisTitle,yAxisTitle, Username, goal);{

Set Variables

Dataset = database.getBios(username);

}

JFreeChart CreateChart(){

Return JFreeChart(title, xAxisTitle,yAxisTitle, createDataset())

}

DefaultCategoryDataset createDataset()

1. Dataset = new dataset
2. Loop through list
   1. Add list to data set
   2. Add goal to data set
3. Return dataset

}

## **Display/Input subsystem:**

Class Display {

Buttons and panels and charts galore

Dataset array;

1. Call database.GetBios(username)
2. Build GUI

GUI consists of four panels and a menubar

* Current stats
* Progress chart
* Analysis
* Update section

1. List current stats in panel 1
   1. JLabel panel1Heading, height, weight, goalWeight, currheight, currWeight, currgoalWeight
2. Add chart to panel 2
   1. Create chart
   2. Add to panel
3. Add labels and text boxes to panel 3
   1. JLabel Panel3Heading, Gender, Weight2, Height2, Age, ActivityLevel
   2. JRadioButton GenderButton, ActivityLevelButton
   3. JButton Update
      1. Onclick
         1. Compile data array
         2. call database.UpdateBios(array, username)
4. Goal analysis on panel 4
   1. Call CalculateWeightGoal(data array)
5. MenuBar functionality:
   1. delete\_user()
      1. call database.deleteUser()
   2. logout()
      1. log out of the database
      2. return to login screen

}

## **Database Interface**

Class Database {

Boolean auth = false;

List dataset = null;

Boolean file = false; //file successfully opened or not

Boolean Authenticate(username, password){

1. Connect()
2. query for username and password
3. compare password
4. return
   1. true if the same
   2. false is not the same

}

Boolean AddUser(Username, Password)

1. Connect()
2. Insert username and password into database
3. Return
   1. True = successful
   2. False = failed

Boolean UserExists(username)

1. Connect()
2. Query for user
3. Return
   1. True = exists
   2. False = does not exist

Boolean UpdateBios(dataset array, username)

1. Connect()
2. Write array to applicable user row in .csv
3. Return
   1. True = successful
   2. False = failed

Boolean DeleteUser(username){

1. Connect()
2. Run delete query
3. Return
   1. True = successful
   2. False = failed

}

Array GetBios(username)

1. Connect()
2. Run select query
3. Compile array
4. Return array

Void Connect(){

1. If not connected
   1. Try/ catch connection
   2. Set connection variable true/ false}

# Unresolved Risks/Mitigation

Based on the scope of this project, we do not feel that these risks require mitigation at this time.

Previously identified risks and mitigations:

**Risk:** Loss of data **Mitigation:** Local/cloud database backups

**Risk:** Account Hacked **Mitigation:** Allow password reset

**Risk:** Exposing data **Mitigation:** Keep user login separate from user data