**Project Analysis:**

**“Cyber Fitness” Application**

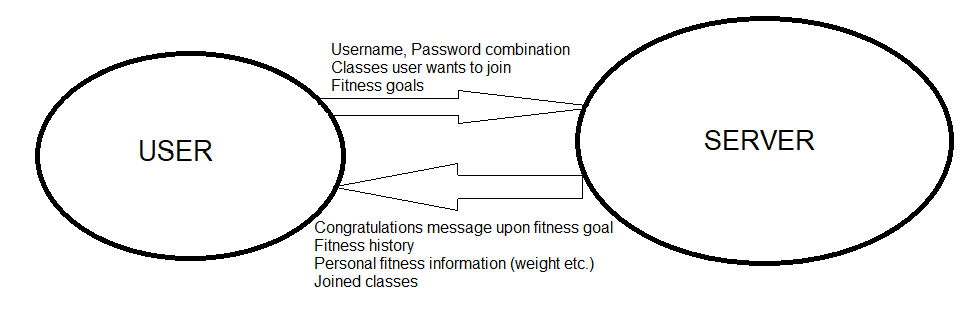
**CMSC 495 6381**

Created: January 30, 2020

**Group 4 Members:** Brisson, Lesly **|** Ciullo, Stephen| Cochran, Jacob **|** Fahlgren, Travis

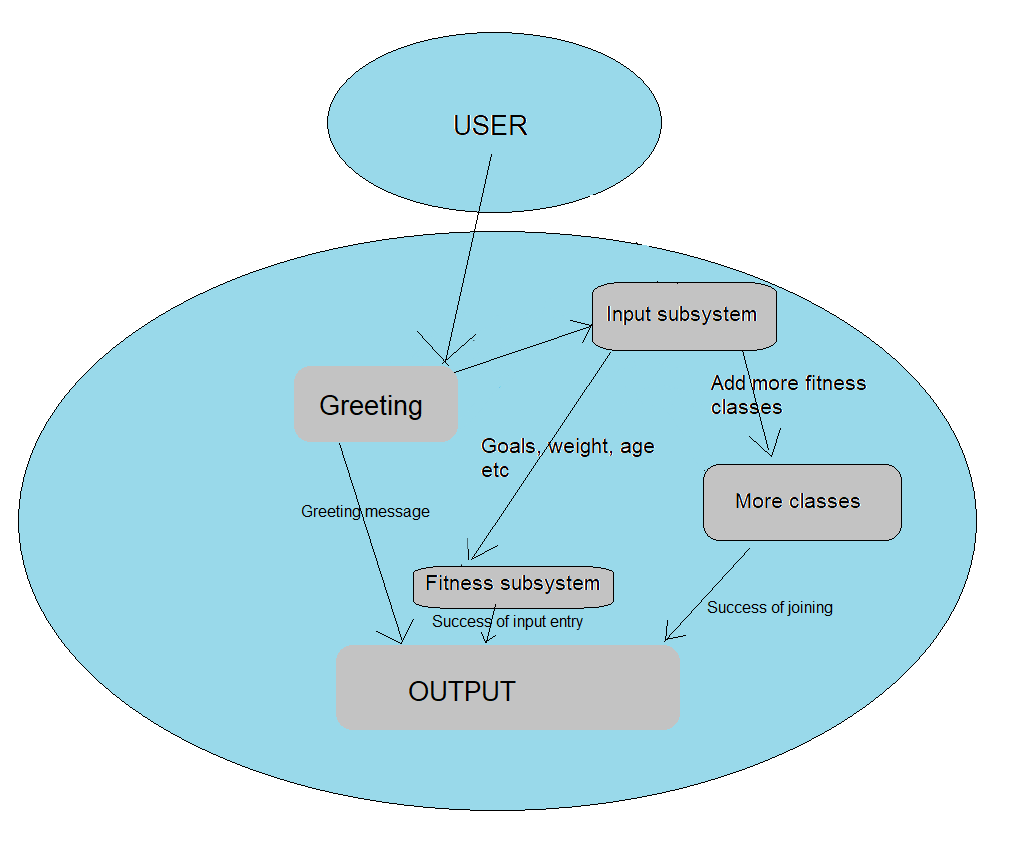
**Identifying components of the design:**

1. Input data coming from user
   1. User Identifying Information (username, password)
      1. Password has at least one capital letter, one special character, and at least eight characters
   2. Classes that the user would like to join/leave
   3. Fitness goals (current weight/target weight), (classes completed/classes desired)
   4. Fitness progress
2. Outside System
   1. User provides information
3. Output data coming from server
   1. All will be displayed to the user and backed up in a log file.
   2. Display of user classes joined
   3. View fitness history specific to the class joined
   4. Display weight, classes, and progress
   5. Congratulations message when fitness goal is met/achieved
   6. Display joined classes upon login
4. Data Processing
   1. Sanitizing data from user to ensure it meets our required standards.



*Figure 1: Context Diagram*

*Figure 1 represents the context diagram for the Cyber Fitness application. The diagram includes the input data, output data, and where they come from. Arrows represent where each part comes from.*



*Figure 2: Subsystem demonstration*

**Explanation of Subsystems**

1. Input Subsystem
   1. The user will be giving input in several different forms. These forms include classes the user wants to join/leave, the user’s goals including their weight and age, and more. It will be called initially by the “Greeting” Subsystem and will display the message of the day.
2. Greeting Subsystem
   1. The greeting subsystem will welcome the user to the page after they authenticate. It will be a personalized message based on the user
3. More Classes Subsystem
   1. The more classes subsystem will be used for adding and removing fitness classes. The more classes will also handle the actual operation of the class.
4. Fitness Subsystem
   1. The fitness subsystem will handle all the behind the scenes calculations. It will also handle when the user wants to start a class, and will prompt the user for post-class information (including weight lost etc.)
5. Output Subsystem
   1. The output subsystem handles the information passed to the user. The subsystem will display an appropriate webpage depending on the input provided, as well as background logging of all activity

|  |  |
| --- | --- |
| **Requirement Number** | **Subsystem** |
| 10001 | Input, Output |
| 10002 | Input, Output |
| 10003 | More Classes, Input, Output |
| 10004 | Greeting, Output |
| 10005 | More Classes, Fitness, Output |
| 10006 | More Classes, Fitness, Output |
| 10007 | Fitness, Output |
| 10008 | Output |
| 10009 | Output |

**Possible Enhancements**

1. Allow the user to have optional fitness goals, i.e. their goal is “to be in better shape”.
2. Encrypt all data from the user
3. Store all user data in an encrypted file
4. Add option to have group classes (more than one user)
5. Timeout feature to prevent brute force logins

**Possible risks and mitigation**

1. User provides invalid data (weight, age, etc)
   1. Have the data sanitization determine appropriate ranges and error out if the user provides invalid information
2. User attempts to circumvent webpage and crawl through file structure
   1. Have checks that prevent the user from modifying their path. For instance, if the user attempts to do “http://fitnessapplication.com/../../../usr/bin/etc” we will have checks that prevent it from navigating there.