**A few notes on this document:**

• I excluded most of the portions that I didn't discuss in lab and replaced these with placeholders: Gray rectangular boxes.

• Primarily in the interest of document length, I excluded some of the requirements I *did* discuss and replaced these with placeholders too (some include the statement in these boxes, others don't).

• Aside from the class diagram on the last pages, there was **no color** in the documents my team submitted. All of this color was added explicitly for instructional purposes to help me navigate through the lengthy reports.

– Further, the color that *was* used in the class diagram was intended to provide a slight   
 enhancement to readability. However, eliminating the colors from the class diagram   
 would not change its meaning in any way.

• Comments in blue boxes were added before lab today.

• Comments in orange boxes were added after lab.

• There are links at various places in the document which I added before lab for swifter navigation to the major concepts and examples I was discussing in class.

**Fitness Witness**

**Software Requirements Specification**

**(MODIFIED FOR INSTRUCTIONAL USE)**

**COP 4331, Summer 2013**

**(Modified in Spring 2014)**

Team Name: The Delirious Designers

Team Members:

Modification history:



**Contents of this Document**

1 Introduction

* 1.1 Software to be Produced
* 1.2 Reference Documents
* 1.3 Applicable Standards
* 1.4 Definitions, Acronyms, and Abbreviations

2 Product Overview

* 2.1 Assumptions
* 2.2 Stakeholders
* 2.3 Event Table
* 2.4 Use Case Diagram
* 2.5 Use Case Descriptions

3 Specific Requirements

* 3.1 Functional Requirements
* 3.2 Non-Functional Requirements

4 Supporting Material

* 4.1 Stored User Input
* 4.2 System-Generated Feedback
* 4.3 Muscle Groups
* 4.4 Exercise Plan Formulas



**Section 1: Introduction**



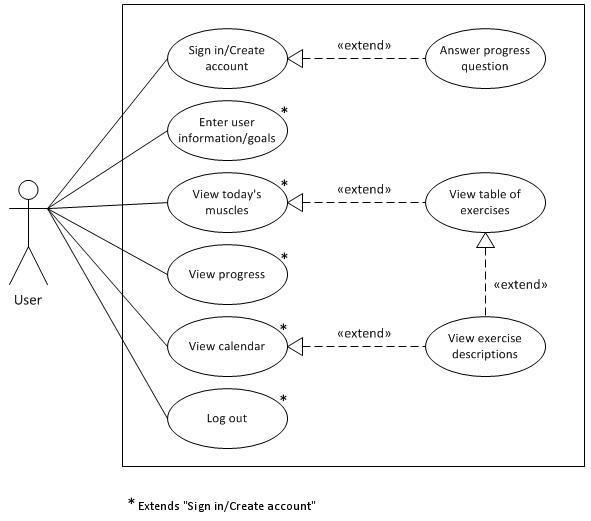


**Section 2: Product Overview**



2.4 Use Case Diagram

*The Use Case Diagram for our system is shown below.*



2.5 Use Case Descriptions





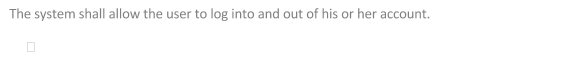
**Section 3: Specific Requirements**

3.1 Functional Requirements

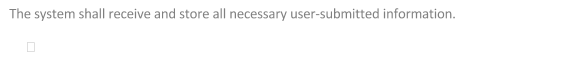
*3.1.1 Creating a new account*

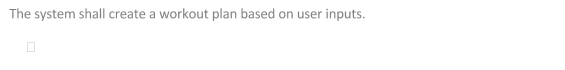
|  |
| --- |
| No: 1 |
| Statement: The system shall enable a user to create a new account. |
| Source: Data Requirement |
| Dependency: No. 3 and No. 7 |
| Conflicts: None |
| Supporting Materials:   * 4.1 Stored User Input |
| Evaluation Method: The tester will follow the process specified for creating a new account and confirm that a message is received to advise that the task was successfully completed. |
| Revision History:   * D. Wittock, 6/17/13, added requirement to document. * M. Kaprocki, 3/15/14, updated phrasing of evaluation method. |

*3.1.2 Logging in and out*



*3.1.3 Storing user-submitted information*

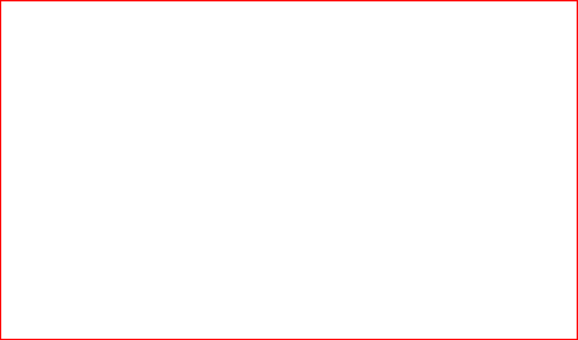


*3.1.4 Creating a personalized workout plan*

*3.1.5 Displaying data*



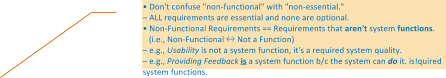
*3.1.6 Tracking data*

*3.1.7 Validity checks on data*

|  |
| --- |
| No: 7 |
| Statement: The system shall check the validity of all user inputs and reject those which do not meet the corresponding criteria. |
| Source: Data Requirement |
| Dependency: No. 3 |
| Conflicts: None |
| Supporting Materials: None |
| Evaluation Method: The testers will attempt to insert illegal values into all input fields that are not simple buttons or checkboxes and will only confirm the sufficiency of validity checks when none of them are accepted by the system. |
| Revision History:   * M. Kaprocki, 6/17/13, added requirement to document. |

*3.1.8 Providing feedback*

|  |
| --- |
| No: 8 |
| Statement: The system shall provide feedback to the user about important internal events. |
| Source: User and Human Factors Requirement |
| Dependency: No. 7 |
| Conflicts: None |
| Supporting Materials:   * 4.2 System-Generated Feedback |
| Evaluation Method: For each feedback item generated by our system (see "Supporting Materials"), recreate the scenario(s) that prompted it to be delivered to the user and verify that the post-conditions of the system match that of the table. If all feedback items can be recreated and their corresponding post-conditions confirmed, then this requirement is satisfied. Otherwise, the requirement is not met. |
| Revision History:   * M. Kaprocki, 6/14/13, added requirement to document. |



3.2 Non-Functional Requirements

*3.2.1 Reliability*

|  |
| --- |
| No: 9 |
| Statement: The system must experience no more than two distinct unresolved failures within 30 days. |
| Source: Quality Assurance Requirement |
| Dependency: None |
| Conflicts: None |
| Supporting Materials: None |
| Evaluation Method: All occurrences of failures should be tracked by maintaining a log book with the following information for each entry:   * Date of failure * Name of user/developer * Preconditions (original state, etc.) * User's last action(s) * Expected system behavior * Actual system behavior * Additional notes (if necessary) * Date resolved (if/when applicable)   The status of this requirement can be determined by reviewing the log book and examining the dates of successive failures. More specifically, the requirement is currently satisfied if no more than two unique unresolved failures occurred within a 30-day period; if this is not the case, then the requirement is not satisfied. |
| Revision History:   * M. Kaprocki, 6/14/13, added requirement to document. |

*3.2.2 Security*

*3.2.3 Ease of use*

|  |
| --- |
| No: 11 |
| Statement: The system must be easy to use. |
| Source: Accessibility Requirement |
| Dependency: None |
| Conflicts: None |
| Supporting Materials: None. |
| Evaluation Method: Surveying users who are unfamiliar with the system about how easily they could navigate and find information. |
| Revision History:   * L. Pereira, 6/17/13, added requirement to document. |

*3.2.4 Performance*

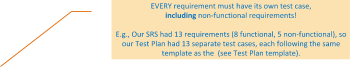
*3.2.5 Supplemental Documentation*

[Go To Example Requirement](#h.1fob9te)

**Section 4: Supporting Material**







**FROM TEST PLAN:**

**Section 4: Description of Individual Test Cases**

⋱  
 *4.1.7 Validity checks on data*

|  |
| --- |
| **Test Objective:** Confirm that the system performs validity checks on all user inputs.  NOTE: Performing this test also completes part of test case 4.1.8 (i.e., *"Providing feedback"*), as all of the possible feedback events during account creation will be expected in response to at least one of the input test cases. |
| **Test Description:**   1. Click the link to create a new account from the login screen. 2. Perform each of the following sequences of subtests, clicking "OK" upon encountering any error message popups (the messages to be expected are noted for each subtest):    1. Click "Submit" (Expected message: "Please enter a valid weight.")    2. Enter the following inputs into the Weight field, one at a time, and click "Submit" after each entry. (Expected message: "Please enter a valid weight.")       1. "five"       2. "1001    3. In the Weight field, input "200"    4. Enter the following inputs into the Username field, one at a time, clicking "Submit" after each entry. (Expected message: "Please enter a valid username.")       1. "username!"       2. "username\_username"       3. "user name"    5. In the Username field, input "user"    6. Enter the following inputs into the Password field, one at a time, clicking "Submit" after each entry. (Expected message: "Please enter a valid password.")       1. "12345"       2. "this\_password\_is\_entirely\_too\_long"    7. In the Password field, enter "testing"    8. Enter the following inputs into the Confirm Password field, one at a time, and click "Submit" after each entry. (Expected message: "Passwords did not match.")       1. "12345"       2. "TESTING"    9. In the Confirm Password field, enter "testing"    10. Enter the following inputs into the Age field, one at a time, clicking "Submit" after each entry. (Expected message: "Please enter a valid age.")        1. "151"        2. "18ish"    11. In the Age field, enter "18" For the Gender option, select Male.    12. For the Level of Exercise Experience option, select 1.    13. For the Area of Focus option, select Stamina/Weight Loss.    14. For the Days Available option, select the following day(s), clicking "Submit" after each entry. (Expected message: "Please select 2 to 5 days. ")        1. Tuesday        2. Tuesday, Wednesday, Thursday, Friday, Saturday, & Sunday    15. Enter the inputs below:        1. Weight: "200"        2. Username: "user"        3. Password: "testing"        4. Confirm Password: "testing"        5. Age: "18"        6. Gender: Male        7. Level of Exercise Experience: 1        8. Area of Focus: Stamina/Weight Loss        9. Days Available: Tuesday & Thursday    16. Click "Submit" |
| **Test Conditions:** See Test Environment |
| **Expected Results:** With the exception of the final submission, which will occur when all fields contain legal values, the tester should be presented with the proper error message (per the specifications above) each time he clicks "Submit," after which he should be returned to the form to insert a new value. |

[Low-Level Requirement Trace](#h.tyjcwt)  
[return to SRS](#h.1fob9te)



**FROM LOW-LEVEL DESIGN:**

**Section 3: Trace of Requirements to Design**

⋱  
 *3.1.7 Validity checks on data*

* **Statement:** The system shall check the validity of all user inputs and reject those  
   which do not meet the corresponding criteria.
* **Implementation:** AccountDetails, Security

*3.1.8 Providing feedback*

* **Statement:** The system shall give users feedback about important internal events.
* **Implementation:** – Highlighting current tab: TabbedForm  
   – Warn on duplicate username: AccountDetails, Security  
   – Warn on password mismatch: AccountDetails  
   – Warn on invalid username: Login, Security  
   – Warn on invalid password: Login, Security  
  

[SRS](#h.1fob9te)[Class Diagram](#h.3dy6vkm)[Test Case](#h.2et92p0)

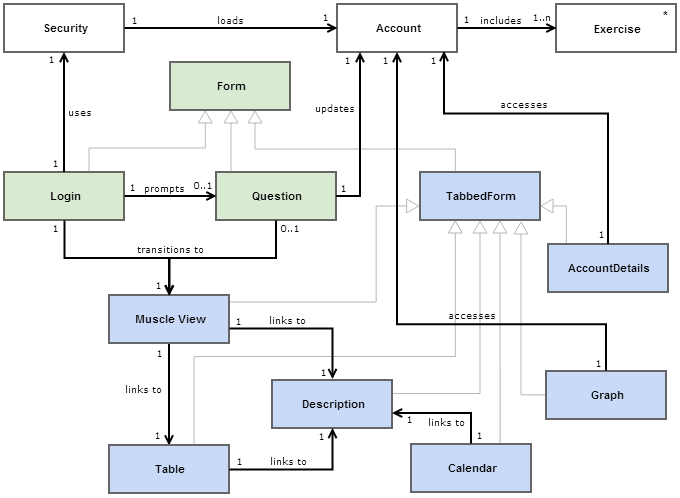
  
**FROM LOW-LEVEL DESIGN:**

[(next)](#h.1t3h5sf)

**Section 2: Detailed Design Information**

The class diagram for our system is presented below. Due to the complexity of the relationships between entities and the space limitation imposed by this document's page width, we have broken the class diagram into two components. That is, the information contained in these diagrams can be combined to form our system's class diagram in its entirety.

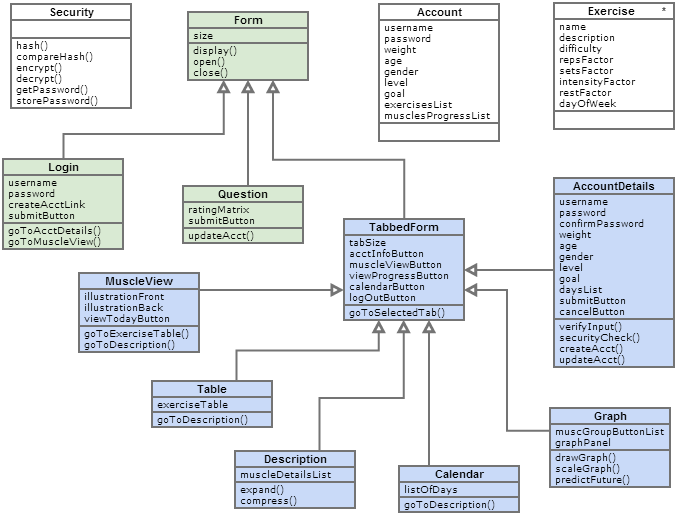
In order to clearly depict the functional relationships between classes, the first component does the following:  
 (1) Omits class attributes and operations  
 (2) Deemphasizes the hierarchical relationships 

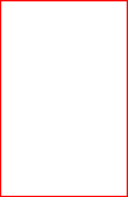




[(next)](#h.1t3h5sf)

The second component of the class diagram includes the attributes and operations for each class and very clearly illustrates the hierarchical relationships found within the system:





[(back)](#h.3dy6vkm)

[Requirements Trace](#h.tyjcwt)

[Test Case](#h.2et92p0)[SRS](#h.1fob9te)