Crucible

Project System Design Document

1. **Contents**
   1. Purpose of this document
   2. References
2. **System Architecture**
   1. Architectural Design
   2. Decomposition Description
3. **Persistent Data Design**
   1. Database Descriptions
   2. File Descriptions
4. **Requirements Matrix**

## ***Introduction***

### **Purpose of this Document**

The purpose of this document is to plot the design structure of the crucible fitness in order to enable a more efficient implementation of our app and also to give an abstract view of the implementation to the client. Within this document, there will be diagrams of every process within the app.

### **References**

Title: How to Make a Fitness App and Make Money 2021

Author: Addevice

<https://addevice.io/blog/how-to-create-a-workout-app-detailed-guide/>

Title: The Swift Programming Language

Author: Apple

<https://docs.swift.org/swift-book/GuidedTour/GuidedTour.html>

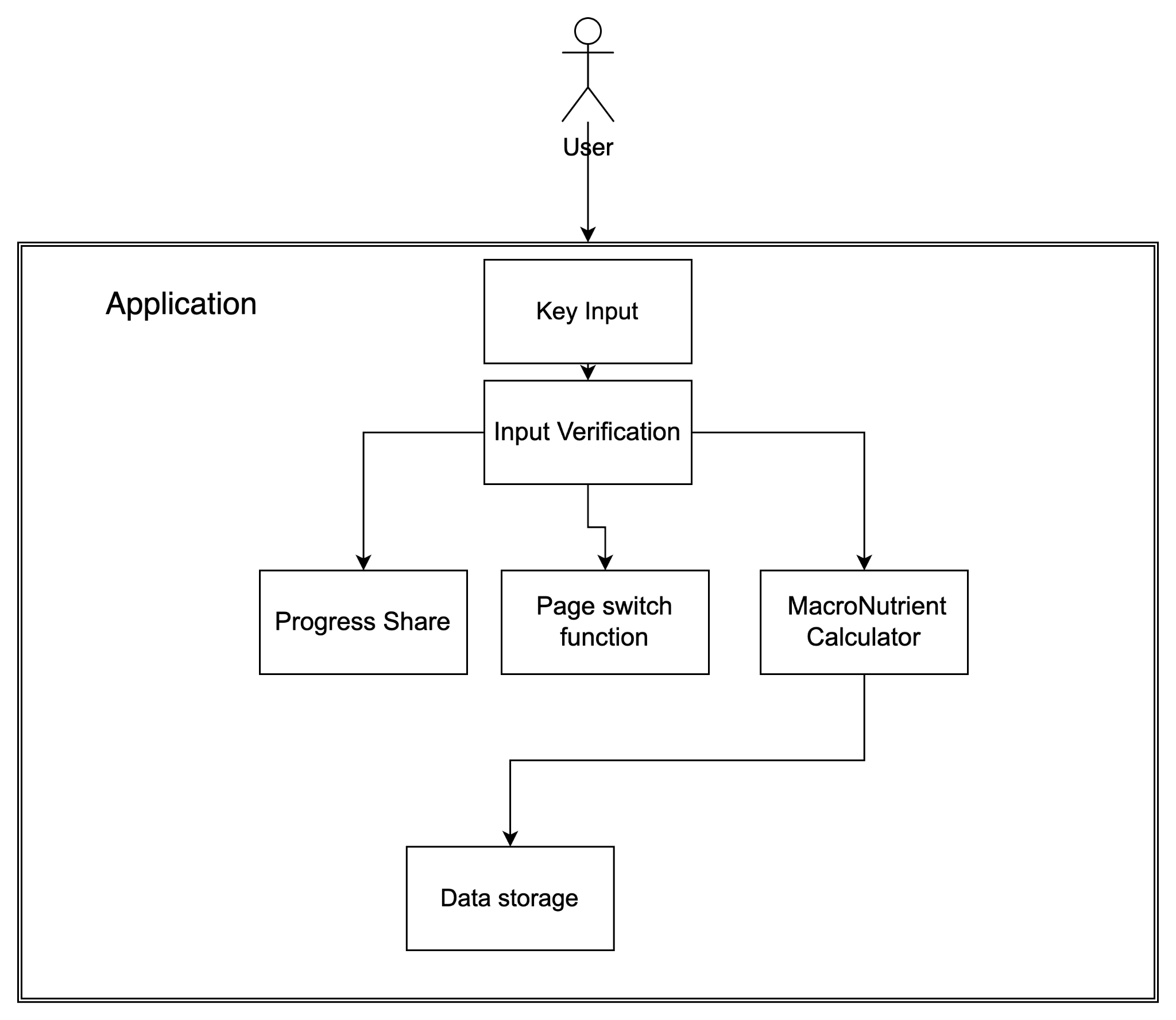
## ***System Architecture***

Under the system architecture section it will display diagrams that showcase the logical architecture of the system that the Crucible team will create. In addition, it will also describe the functions in a descriptive manner within this section. It shall also describe the technology that comprises both hardware and software. In accordance with this, it shall also have a corresponding technology architecture design graphic. This shall all be located within the architectural design section of system architecture. The information that is located within section 2.2 shall discuss the decomposition information for the Crucible project. Therefore, it will describe the functions of the application, contain diagrams, and discuss the methods in a descriptive manner. This application will be using swift and is being developed for ios systems.

### **Architectural Design**

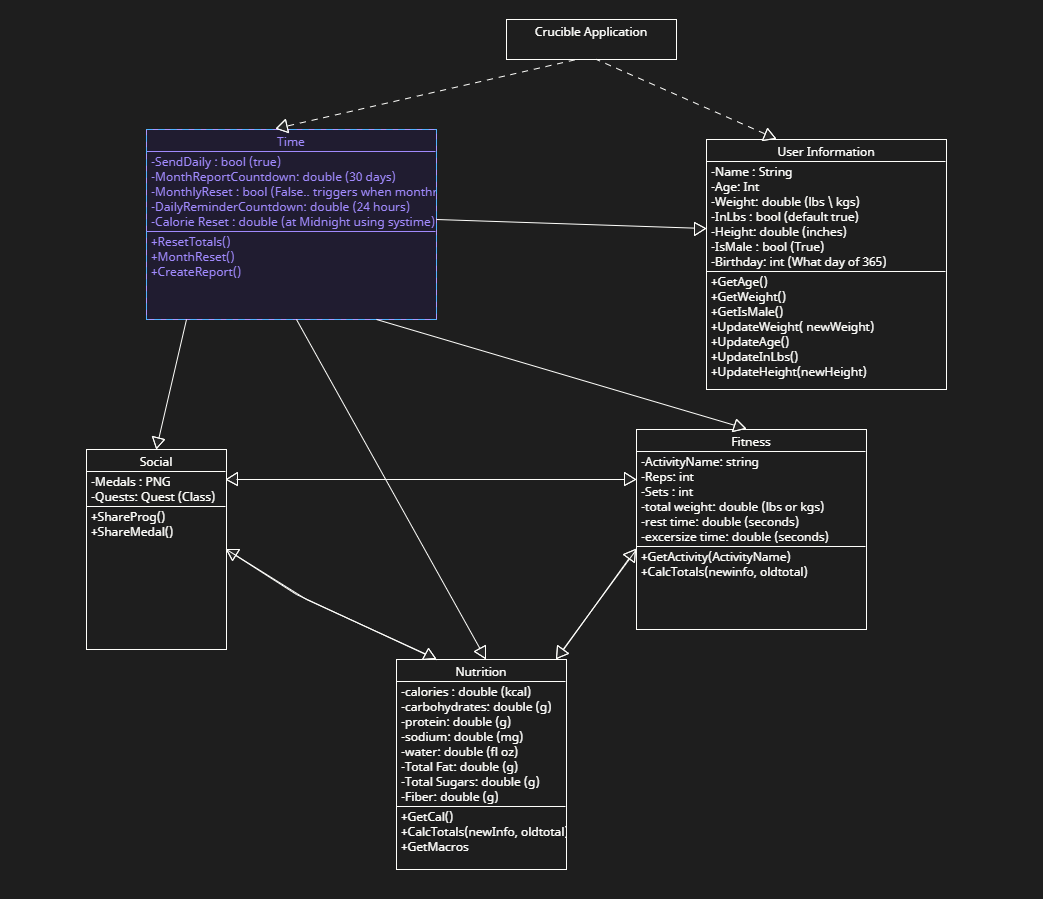
#### An Overview of the Crucible Application

Within this diagram shows the basic logic that follows our application. First, the user attempts to add information to the application. Then that input is verified and added to the applications’ database if and only if the information that the user has provided is valid. In addition, we also have to have the ability to switch between our three pages for our different pieces of the application namely Fitness, Nutrition, and Social pages. In addition, we shall add the ability to share progress with others and also the ability to share medals which will fall under the progress share category. Moreover, We have to store all the information that the user inputs within a database so they can keep track of their progress, in addition, it allows us to create reports and updates for the user.



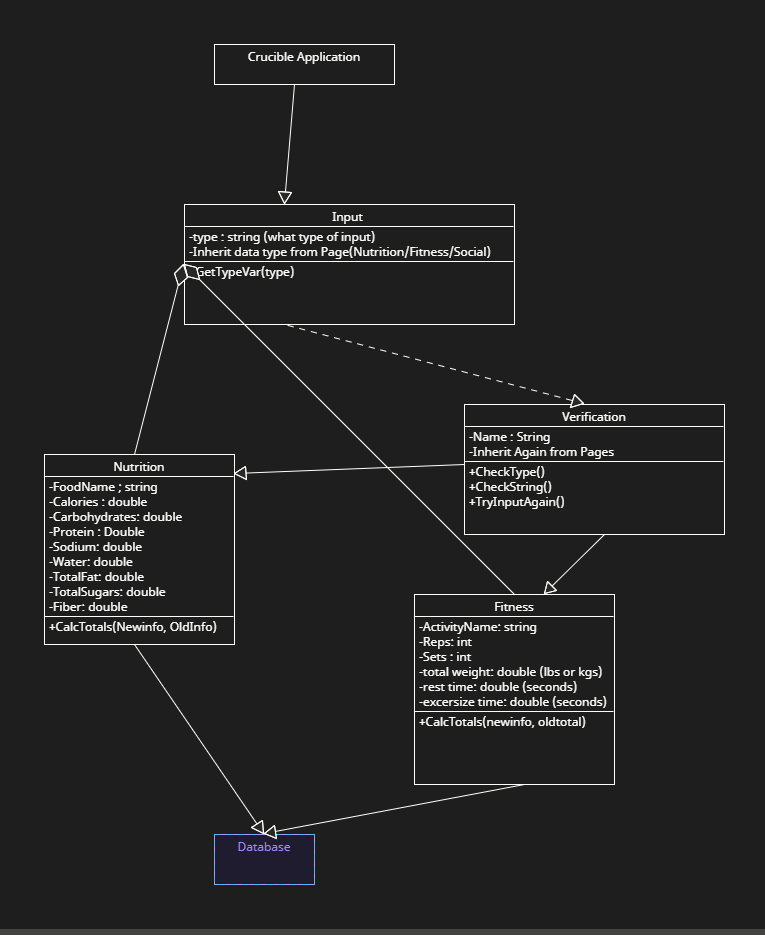
#### The inter-functional interaction

Within the crucible application, we are using many different functions. Across multiple different displays, therefore, the crucible team found it necessary to showcase the different functions’ interactions. The User information will only interact with the Time Class which will use that information to make accurate reports. In addition, the Social, Nutritional, and Fitness segments will all communicate with each bi-laterally. This is because many of the functions will use the information that is input into the other classes and components of those classes. For instance, the Social Class has the ability to share progress with others, therefore naturally it must be able to have some progress to share whether that information is located within the fitness or nutritional sections of the application.



#### Input and Verification

For the Crucible Application we will need to accept user input and also verify that the input is correct for the data types and also make sure that the information will not break anything within the application. Therefore, when the user wishes to input information they must select for which page they are inserting information. By inheriting from the relevant class, we are able to make sure that the information that they are adding to the variable not only corresponds to the class but are also valid inputs.



### **Decomposition Description**

Our system for the project shall be an object oriented programming approach. It is the type of programming that the group as a whole has the most experience in and in addition to that it perfectly fits the requirements that we are to fulfill.

For example we will be using classes and protected variables for most of our data storage that will interact with the other classes of the application. Displaying this will be the nutrition, fitness and social classes. These different classes will need to communicate with each other however, they should not be allowed to alter their information therefore, we will be using protected variables to accomplish this. In addition, our classes have to be able to access the database that we intend to use. As of this current moment we intend to use a hashtable and a self-made hash function to be able to store our information within our hashtable.

Another point that lends itself to using classes as well as object oriented programming is the need that each sub-type of our application that uses will hold its own respective class as to reduce collisions with data types and also allow for inheritance when we have the user adding new workouts and meals.

## ***Persistent Data Design***

### Database Descriptions

We break down the database into 3 classes connected to one user account.

* Nutrition class: containing all macros that a user has consumed. Including fats, protein, and carbohydrates. All of which will be of the double data type. The nutrition class will be able to calculate the total calories consumed.
* Fitness class: containing all workouts that a user has done. This will contain string data type for the kind of activity that the user has done. Will have an integer data type for the number of reps and sets of that activity that the user has done, as well as the amount of weight the specific exercise was done with. The total weight will be a double type that the user will input that states the amount of weight actually lifted, e.g. 135 pounds input if the user did bench press of 135 pounds.
* Photography class: containing progress pictures and metadata from those pictures. Which will be HEIC
* Achievements class: containing the badges the user has earned. Which will be a boolean type to see if the user has already achieved certain achievements.

### **File Descriptions**

We will organize our project documents into five main groups:

* UI-Assets: All of the images used by the application are stored here. This includes the Crucible logo, icons for the fitness, nutrition, and social pages, icons for the badges, and images for specific exercises.
* Network: This directory holds files the application uses to communicate with the servers and databases. It also contains the user’s unique id for information lookup.
* Fitness: The fitness directory holds files relevant to the fitness portion of the app. The fitness class and the data for specific exercises will be stored here. The fitness class has member variables of type float to hold the user's weight and height, and integer variables to hold the user's age. There is an array of type workout to keep track of the workouts the user has done. The workout class stores the name of the exercise as a string, the date and time the workout was done as an int (with the default value of 0), the number of reps done in each set as an int array, and the number of calories doing one rep of this exercise burns as a float. There is a file with default workouts in this directory which can be added to by the user if they wish to add a workout not listed in the app. There is also a file containing descriptions of the default workouts.
* Nutrition: All files relating to the nutrition aspect of the app are stored in this directory. The nutrition class has an array of the meal class type that stores the meals the user has entered, the number of calories the user has consumed in the day as an int, and the target calorie count as an int. This class also has functions that can calculate the target calorie count per day based on the user’s current body measurements and their health goals. The meals class stores the number of calories in the meal as float, the date and time of the meal as an int (default 0), and the grams of protein, fat, carbs, and sodium as ints.
* Social: This directory holds the files necessary for the social and achievement part of the application. Information such as social accounts and medals are stored here. A file containing all possible achievements and their descriptions is used to display the achievements to the user.

## ***Requirements Matrix***

| Requirements Matrix | |
| --- | --- |
| Function Name | Short description |
| 1.Key Input | |
| Input | Cleans out any possible input that may contain SQL injection. |
| 2.Progress Share | |
| ShareProg | Will allow users to share the progress of their fitness journey. |
| ShareMedal | Will allow users to share their achievements. |
|  |  |
| 3. Macronutrient Calculator | |
| calcTotal | Calculates the total calories that a user has consumed. |
| GetMacros | Getter for all macros, fats, protein, and carbohydrates. |
| GetCal | Getter for calories consumed. |
|  |  |
| 4.Input Verification | |
| Input | Cleans out any possible input that may contain SQL injection. |
| UserInfoHash | Hashes sensitive user information. |
| 5.Page Switch | |
|  |  |
| 6.Data Store | |
| monthReset | Resets monthly data. |
| resetTotal | Resets total daily calorie total . |
| CreateReport | Generate a report of the previous month, such as total calories of month, total weights lifted, etc. |

### 

## Appendix A – Agreement Between Customer and Contractor

## The team and our customers are agreeing to have a fitness app created that will allow the user to track their exercise and see how many calories they ate during the day. The app will be secure, and users can securely share their workouts with other people who use the software. There will be no transaction between the user and the team without notifying them and asking them to pay for our app. Also, the customer has to approve the use of their workouts being tracked when on the fitness application. The customer will receive a working app when they download it and we can address any issues they have after they use the features that we have.

## Our group knows that there could be future additions to this document and parts of the app may not be developed. In the event that we have to change this form, we will notify the customer and make sure they approve the new requirements we have and would work with them to have the best application we can build. The way we will make sure they participate in all aspects of the development process and also be notified of any and all changes that the team may wish to implement. In addition, the Crucible project team will be subjected to creating update reports should the requirements change in a vast manner that shall be produced in a timely manner. Should the customer require any modifications of the system that they wish to implement they can submit a written document that shall entail the details and specifications of the changes that they will desire. However, these changes may or may not be approved by the Crucible team under the circumstances of cost, and time. However, should the changes be necessary they shall be implemented with a new timeframe of development and release to the customer. In addition, should the change be infeasible to implement within the release timeframe, an update can be developed for the Crucible application and this update will be released after the initial agreed-upon release date, however, the update shall take precedence with suggested changes for the application. Lastly, both parties must agree upon large changes within the application. This is to mitigate the amount of time that is required to develop the application. In addition, the development team has oversight as well as responsibility for their respective domains and the development of the Crucible application.

## Signing below acknowledges the facts that are denoted above and I do accept on behalf of myself or the company I represent has read and approves of everything outlined above.

## Customer Signature: X\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

## 

## Customer Comments:

## Appendix B – Team Review Sign-off

All Members of the Crucible team have read and reviewed this document and have agreed on the content and format.

I [Miles Wilson](mailto:milesw1@umbc.edu) do affirm that I have read and understood the contents of this document and as well do actualize the fact that this document will contain information that declares what is expected of myself and my teammates.

X\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Date:\_\_\_\_\_\_\_\_\_\_\_\_\_

I [Michael Duong](mailto:mduong2@umbc.edu)do affirm that I have read and understood the contents of this document and as well do actualize the fact that this document will contain information that declares what is expected of myself and my teammates.

X\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Date:\_\_\_\_\_\_\_\_\_\_\_\_\_

I [Aryan Singh](mailto:arysin1@umbc.edu)do affirm that I have read and understood the contents of this document and as well do actualize the fact that this document will contain information that declares what is expected of myself and my teammates.

X\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Date:\_\_\_\_\_\_\_\_\_\_\_\_\_

I [Tommy Stinson](mailto:tstinso1@umbc.edu) do affirm that I have read and understood the contents of this document and as well do actualize the fact that this document will contain information that declares what is expected of myself and my teammates.

X\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Date:\_\_\_\_\_\_\_\_\_\_\_\_\_

I, [Jason Garcia Solorzano](mailto:jgarcia8@umbc.edu) do affirm that I have read and understood the contents of this document and as well do actualize the fact that this document will contain information that declares what is expected of myself and my teammates.

X\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Date:\_\_\_\_\_\_\_\_\_\_\_\_\_

## Appendix C – Document Contributions

This section holds the individual contributions of each team member and an estimation of what percentage each member has contributed to the writing.

Tommy -

Miles - Appendices, 1.1 100%,1.2 100% Section 2 33%, Section 3 20%, section 4 33%

Michael - Section 2 33%, 2.1 100%, Section 4 33%

Jason - Section 2 33% Section 3 20%, Section 4 33%

Aryan - Section 3 50%