

Brody Pearman

CSCI 240

Professor Goettsche

September 28, 2018

## Requirements

*Preface:* This is the first version of this document and it will outline the system and user requirements for our mobile application's handling of user input. Since this is the first version, everything is new, and the rationale for its creation is to outline what is needed to effectively and efficiently handle user input.

*Introduction:* The system is a necessity for the function of our software. Because our application provides people with workouts and exercises tailored to their goals and abilities, we need precision in effectively storing the user input such that it can be readily referenced for our other systems in our program. The storage of and usage of user input is essential to the system that creates the workouts for users in the sense that the workouts are built from what users provide to us. Additionally, after our users complete weeks and months of workouts, they are going to want to see their goals and the progress they have made. Our storage of this information will thus interact with that system in a manner that allows that system access.

*Glossary:* HashMap- a data structure in Java that allows for efficient access time by storing pairs of data

String- object class in Java that is built out of character

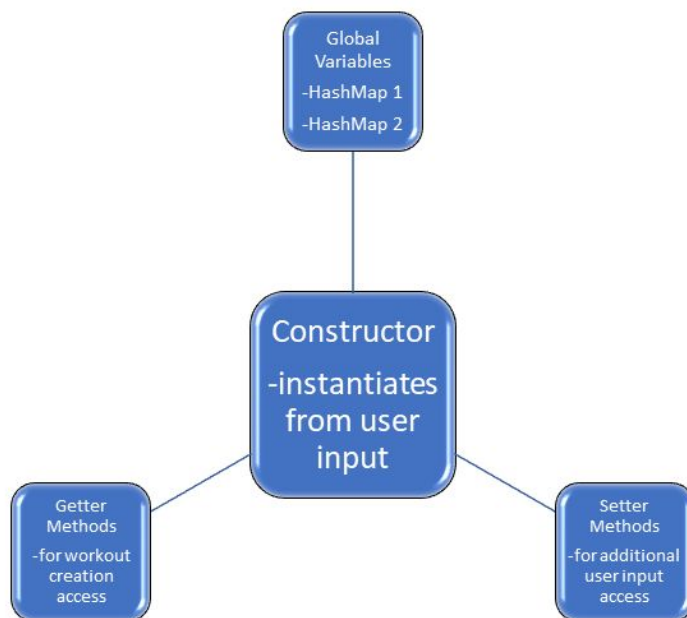
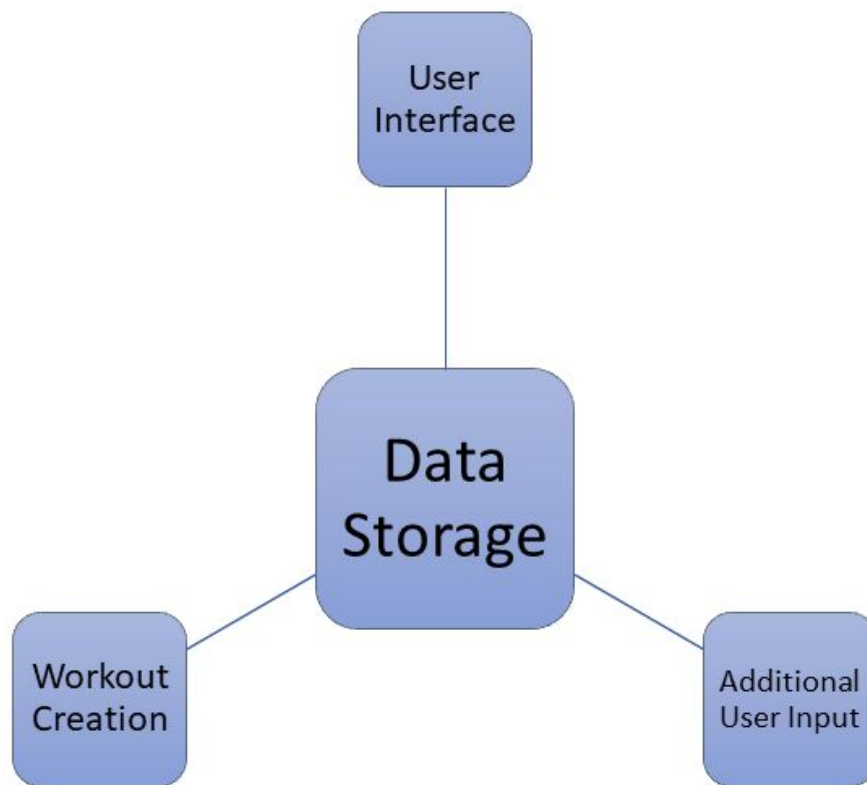
*User Requirements Definition:* The user will be able to enter their data about their weight, lifting experience, reduced to a scale, and their exercise goals from a drop down menu.

This is a functional requirement. The user will be able to consistently reference their data input, meaning that they will be able to see their original weight, their original goals, and their original strength numbers. This is a functional requirement. The user should be able to have access to a graph of the growth of their core lifting numbers based on the user input. This is a functional requirement. This portion should be available to users by November. This is a non-functional requirement.

*System Architecture:* The user interface shall provide our databases with initial user input when our application is downloaded. We shall store this information in our databases and provide access to the group member who is creating the algorithm for the creation of the workouts. Additional user input will occur weekly and should be comparable to the initial user input.

*System Requirements specification:* The system needs to be able to interact with the other three portions of the application. It needs to be able to take the inputs from the user interface and store and organize them efficiently into a HashMap that maps their goals to their weight in the form of a String and into another HashMap that maps their experience to their lifting numbers (in the form of a String, which is a functional requirement. These strings will be cast into a form that is able to be easily parsed by the system that creates the workouts, another functional requirement. It needs to be able to provide this information to the system that handles new periodic user input so that our users can see their progress. This is a functional requirement. Lastly, because it depends on the user interface, that portion of the project is constraint on the completion of this system and thus, it cannot be completed until that portion is functioning, which is a non-functional requirement.

*System Models:*



*System Evolution:* The fundamental assumptions that this system is based on is that the user interface will provide a parsable string of user input that can be assessed and stored into its proper data structures. The lifting experience, max numbers and weight should all be integers, while the goals will be specific strings with attached integers. The anticipated changes that could be expected are the changing from HashMaps to a more memory efficient structure if memory becomes an issue. However, the reason we are choosing HashMaps initially is due to their effective access time complexity.

*Appendices:* We are choosing to code in Java for the development of Android based application. Our system for data storage will likely be a data storage class that has two HashMaps as its global variables, a constructor that instantiates them with user input, getter methods for the class that creates the workouts, and setter methods for the class that receives the new user input.

*Index:* Whole System Structure- page 3, display 1

Data Storage Structure- page 3, display 2