

Comp 55 Application Development

JumpStart

By

Drew Overgaard, Isabelle Tran, Josiah Yoshimura,

Kyle Hartman

<https://basecamp.com/2707759/projects/11638120>

May 1st, 2016

Individual Contributions Breakdown

Drew

- GUI – 15%
- Notes – 80%
- Quit – 100%
- File I/O – 30%
- Timer – 80%

Isabelle

- Graphics – 100%
- Workout – 80%
 - Bulking/cutting
 - Cardio
 - MuscleGroup
 - Difficulty
 - workoutDisplay
 - Enumeration

Josiah

- Statistics – 100%
- Notes – 5%
- CustomWorkout – 100%
- File I/O – 70%

Kyle

- GUI – 85%

- Notes – 15%
- Workout – 20%
 - Bulking/cutting
 - Cardio
 - MuscleGroup
 - Difficulty
 - workoutDisplay
- Timer – 20%

Table of Contents

Summary of Changes.....	5
Customer Statement of Requirements.....	8
Glossary of Terms.....	9
Functional Requirements Specification.....	10
Interactions Diagram.....	14
Class Diagram and Interface Specification.....	15
System Architecture and System Design.....	17
Algorithms and Data Structures.....	18
User Interface and Implementation.....	19
History of Work & Current Status of Implementation.....	20
Conclusions and Future Work.....	21
References with Annotations.....	28

Summary of Changes

- Overall objective maintained
- Instead of working on embedding workout videos we decided to create a custom workout option
- We changed GUI to create a more user friendly interface to reduce user effort
- Created pause, start, and reset for timer allowing users to feel that their in control of their workout and progress
- UML
 - We added more classes
 - Timer
 - FileIO
 - ColumnScale
 - Counts
 - Histogram
 - StatsFunction
 - WorkoutSpec
 - WorkOutReps
 - We designed panes for each of our classes
 - CustomWorkoutPane
 - BulkingMuscleOptionPane
 - CardioPane
 - CuttingMuscleOptionPane

- DifficultyPane
 - MainWorkoutPane
 - MenuPane
 - NotesPane
 - SomePane
 - TypeSelectPane
- We enumerated strings
 - DifficultyEnum
 - Type
 - MuscleGroup
- Interaction/Sequence Diagram
 - Due to the changes in our UML and structure of our application, our sequence diagram differs a lot from what was submitted in our System Design Report
 - The user begins by clicking SomePane instead of our elusive Menu we originally had. We heavily utilized Panes which allowed us to switch to different functionalities of the application. From SomePane the user is taken to MenuPane which provides them the options of Work Out, Statistics, Notes, and Quit.
 - If Work Out is selected the user is taken to TypeSelectPane which is broken down in a separate Interaction Diagram
 - If user selects Bulking or Cutting they will be taken to BulkingMuscleOption or CuttingMuscleOption, respectively. This is where the type is set.

- Users will then be given the option in DifficultyPane after they have selected which muscle group which will setMuscleGroup. DifficultyPane will setDifficulty and switch to MainWorkoutPane once selection for difficulty has been made.
 - incrementCount will be called once workout is completed in MainWorkoutPane and that count will be sent to Count
- If user selects Cardio they will be taken to Cardio Pane and set their workout type to cardio, once they have completed their workout, incrementCount will called and that information will be sent to count
- If user selects Custom they will be taken to CustomWorkoutPane which will call take_input() which will allow users the ability to customize workout. Once the workout is selected and completed, incrementCount() will be called and information will be sent to Count.
- If Statistics is selected the user is taken to StatisticsPane which utilizes File I/O and Histogram class for its functionality
- If Notes is selected the user is taken NotesPane which uses File I/O to store text and font size.

Customer Statement of Requirements

Health is wealth. Whether you are in pursuit of a chiseled body or just want to maintain a healthy lifestyle, Jumpstart can help you stay accountable. Fitness is not a destination but rather a lifelong commitment to taking care of your physical wellness. Often times, people are easily discouraged by slow results and progress. Jumpstart helps track progress to provide users the ability to see the hard work they put in when the mirror may not reflect it. According to, Rachel Dixon for The Guardian, photographic motivation causes the inverse effect and makes the end goal seemingly intangible. With statistical analysis at the user's fingertips they will be able to see their weekly performance and monthly performance allowing them to adjust their workouts accordingly. Currently, users have to use separate applications to track different aspects of their progress and our goal is to consolidate features into one user-friendly interface. With Jumpstart, users will have a note-taking interface which will provide them the opportunity to plan ahead and prioritize working out into their schedules which Dixon states is a great way to stick to your fitness regime. The application will provide a variation of workouts tailored to specific muscle groups keeping users engaged, lack of interest in workouts has been shown to greatly diminish progress and leads back to sedentary ways.

Glossary of Terms

Chest: large pectoralis major, smaller pectoralis minor

Arms: triceps, biceps

Shoulders: deltoids – anterior, middle, posterior head

Back: upper and outer lats, lower lats, middle back, lower back

Core: trunk flexors, back extensors, obliques

Cardio: exercise to raise your heart rate for at least 20 minutes

Legs: Hamstrings, quads, and calves

Bulk: gaining muscle and fat at about equal weight to gain muscle mass as quickly as possible

Cut: dropping body fat, while attempting to preserve as much muscle mass as possible

Function Requirements Specification

A. Stake Holders

- a. Users
 - i. Avid Gym Goers
 - ii. Body Builders
 - iii. First-time gym users
 - iv. People without a gym membership
 - v. People interested in beginning their fitness journey
- b. Sponsors
 - i. Gym owners
 - ii. Doctors interested in physical wellness
 - iii. Fitness gurus

B. Actors and Goals

- a. Initiating
 - i. Goals – making health and fitness accessible for all
- b. Participating
 - i. Goals – fitness and health

Casual Description:

Workout:

User opens the program and selects workout. It generates a workout for them, and increments the count associated with that particular workout. User can quit at any time.

Statistics:

User selects statistics and then the program displays some simple histograms with the average amount of times they worked out per week included. User can quit at any time.

Notes:

User selects notes, and a Text box will appear allowing the user to enter in anything they want, also being able to increase/decrease the font. User can quit at any time.

Fully-Dressed Description**Workout Main Path:**

1. User opens program
2. Programs displays a menu (Workout, statistics, notes, quit)
3. User selects workout
4. Program prompts user to select a workout type (bulk.cut)
5. After selection, program prompts user to select a muscle group
6. After selection, program prompts user to select a difficulty (easy/medium/hard)
7. Once the timer is up, a pop up will let the user know they are done

ALT Path: User finishes workout early or decides to quick and can quit back to the main menu

8. Returns to main menu
9. User exits the program

Statistics Main Path:

1. User opens program

2. User selects statistics from menu
3. Program displays the following statistics
 - Program displays a histogram of each work out type and the frequency for each (gets frequency from counts variable for each workout type)
4. Returns to main menu
5. User exits the program

Notes Main Path

1. User selects notes from menu
2. User clicks text box and enters in any information they wish
3. User can change the size of the font using buttons in the window
4. User quits to main menu or any other menu and the changes are saved

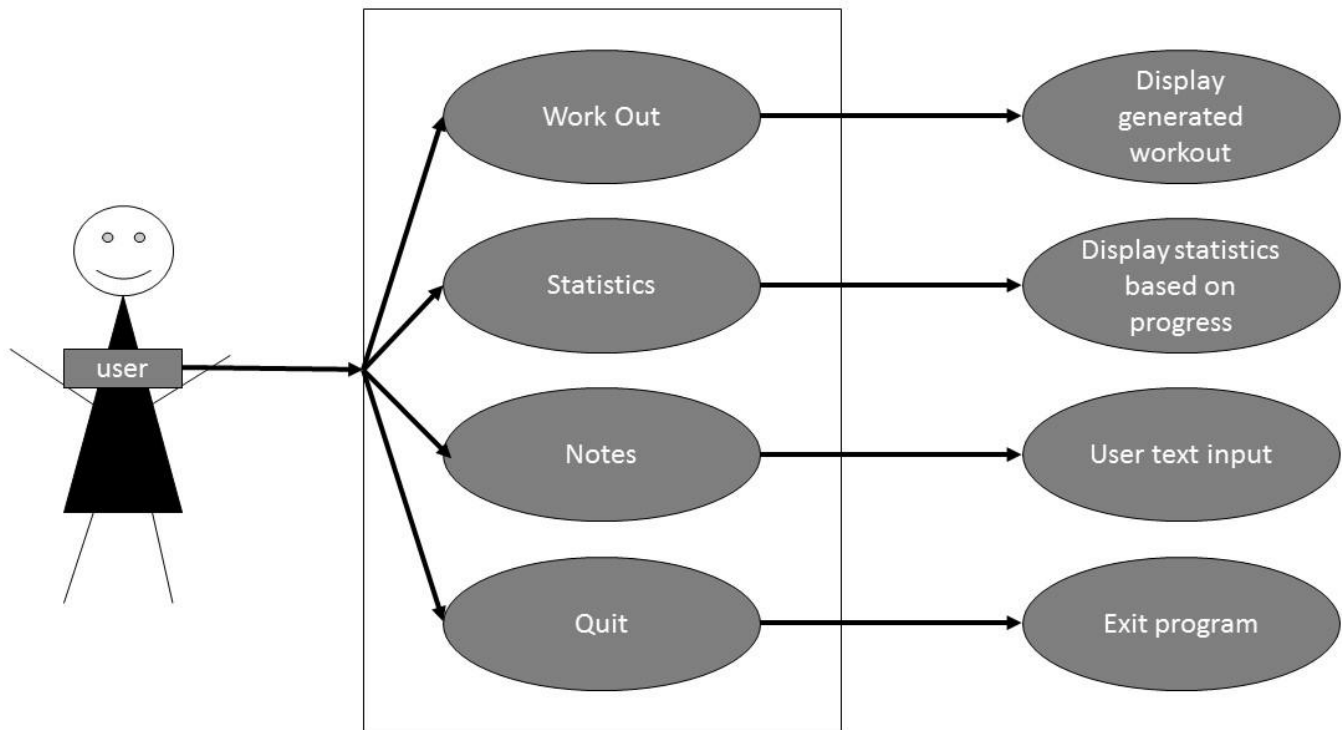
General Alt Path (Quits from menu):

1. User selects quit at main menu
2. Program exits

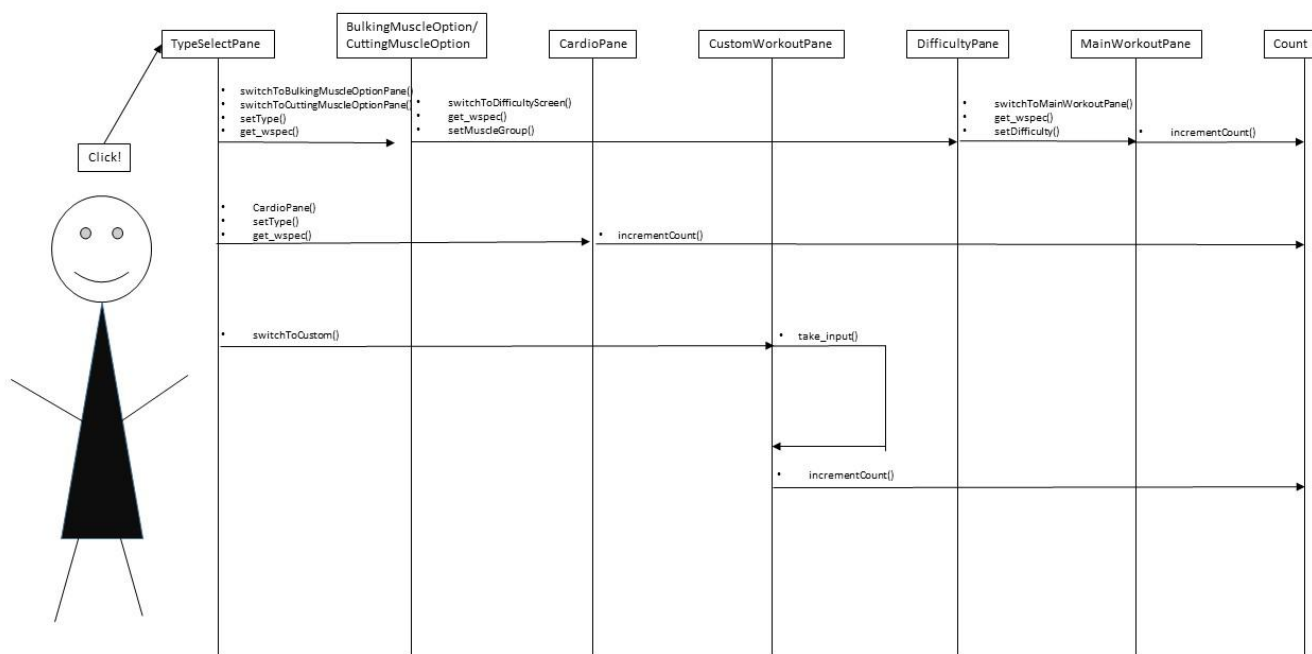
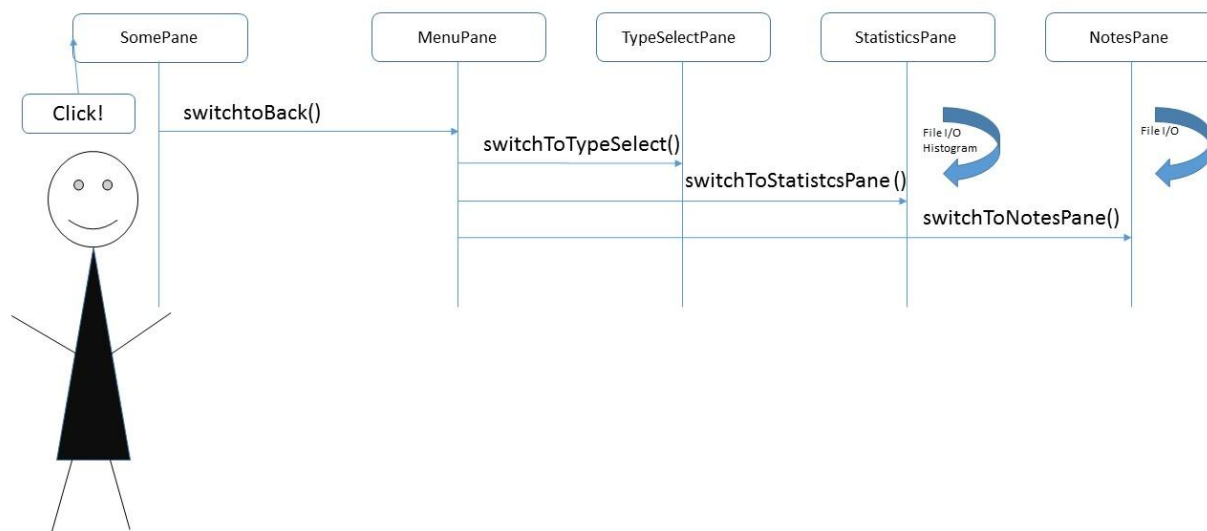
General Alt Path (quits from anywhere else):

1. User selects quit during a workout or in statistics (Should be able to do this after any step in any use case)
2. Program exits to

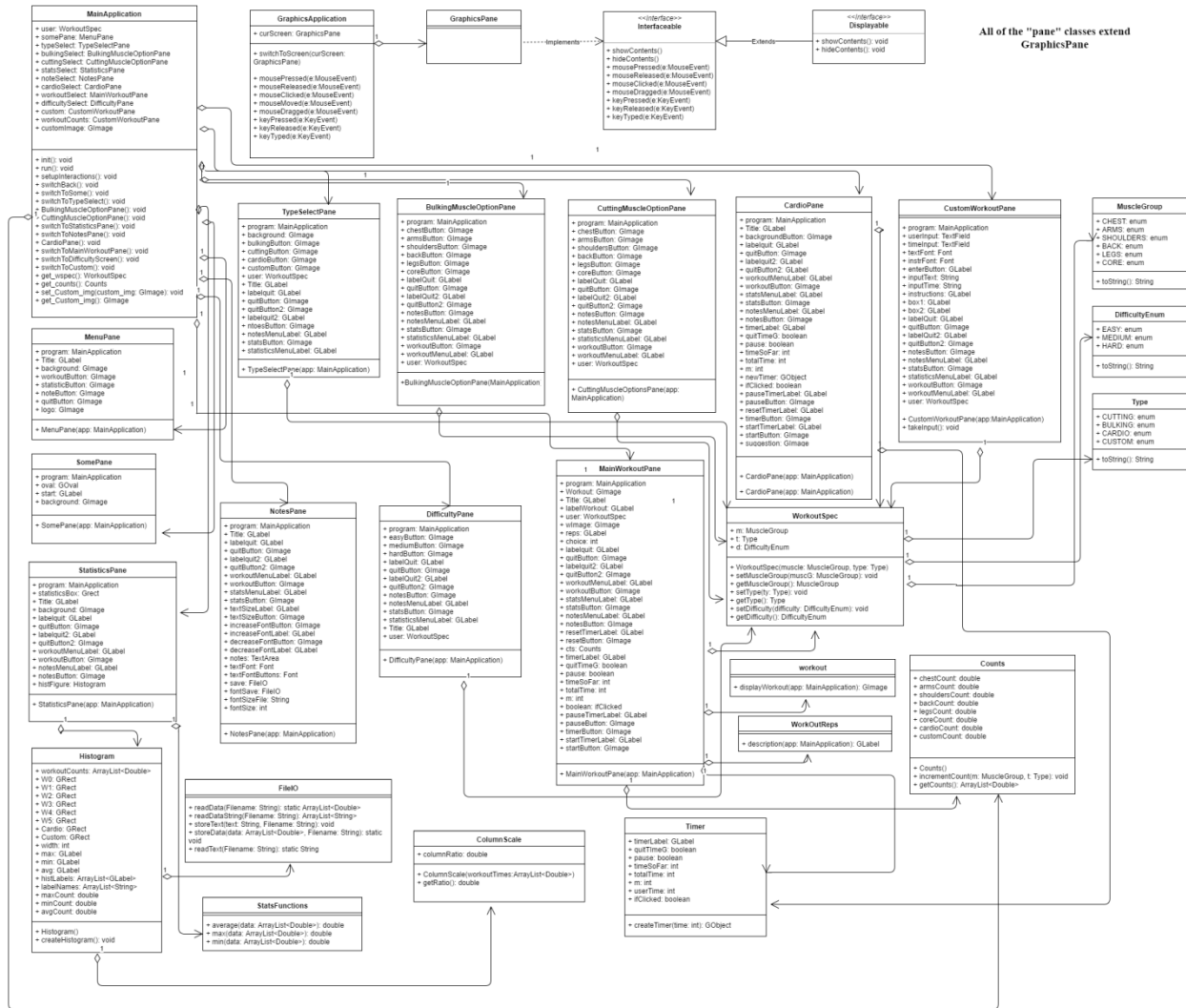
Use Case Diagram



Interaction Diagram



Class Diagram and Interface Specification



Difficulty.java

The purpose of the difficulty class was to provide feedback to the user providing them with useful tips based on their progress. The intent was to encourage users after a certain number of workouts of a specific type and muscle group to progress to the next difficulty level. Furthermore, the application would provide risk warnings to users in the event that they attempted a harder difficulty level workout prior to attempting the lower difficulty level workouts. These would all be highly suggested input but it would not force users to go below or

above the difficulty they chose. This aspect of the application would have provided a more personal touch giving users a virtual personal trainer.

System Architecture and System Design

a. Mapping Subsystems to Hardware

Our system only needs an Operating system to run, the user will not have to access any kind of web browser or web server to use or receive any of the information that will be in our program.

b. Network Protocol

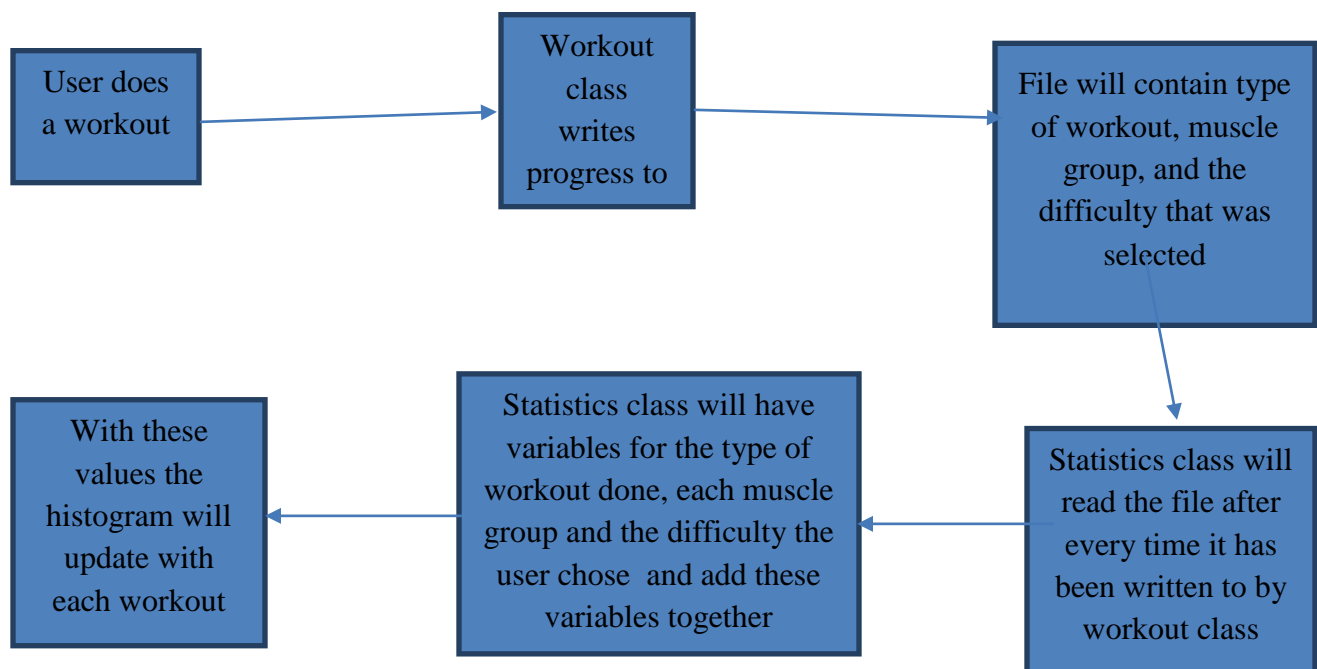
Our system will be run on a single machine and will not need any communication protocol.

Hardware Requirements

- c. Our System will need a screen display and some disk storage to store the program and all of the user information. We will have a color display and for resolution we have decided to manually set the size of the window. We would like to also have a minimum of 1 Gigabyte of hard disk space to store the files the program along with its pictures and user information.

Algorithms and Data Structures

- a. Yes, our system will need an algorithm to calculate user progress and display a statistical analysis of their progress. We will store the user progress including, number of workouts completed, which muscle groups the workouts completed were in, and difficulty of the workout that the user chose. Based on this data we will create a bar graph to show users which areas they can improve upon and which areas they are doing well in



- b. We used arrays a number of times in our project, we used an array to store in the information that the user entered when they clicked certain buttons, this data was then saved in our FileIO class to be used later for our statistics, we also used an array to store the GLabels that appear below the histograms in our statistics screen.

User Interface Design and Implementation

In the final product we removed the edit and save buttons for the notes, but decided to add in increase/decrease font buttons, because we felt the font size was too small. We have also implemented a drop down menu for each screen so that the user has easy access to switch from one window of the application to another. We have also added in play/pause/reset buttons for the user to use to make the workouts more user friendly.

History of Work & Current Status of Implementation

On our initial Gantt chart we were scheduled to have a base created for the windows and the menu buttons by April 2nd. We didn't start coding until April 4th and the basic window wasn't completed until the 7th of April. Throughout the project time Kyle and Isabelle updated and made the user interface more appealing. We were also scheduled to complete our main data structures by the 15th. Josiah used an array list for statistics and it was completed by the 19th. We were scheduled to have the notes completed by the 23rd and Drew completed it early, also including additional features for the notes such as increasing/decreasing the font. Creating statistic was scheduled to be done by the 23rd, which it was. We spent the 26th and the 28th adding additional features that were not originally in our plan, such as the user being allowed to create a custom workout, we were also able to add in play/pause and reset buttons for the timer which again was not originally planned.

Conclusion and Future Work

Drew Overgaard:

1. Describe the technical challenges you encountered in the development of your software product.

We encountered multiple technical challenges at different stages of developing our software product. The earliest challenge was coming up with a way to store user choices depending on the buttons they pressed in order to display the correct workout based on their choice. Isabelle and Kyle worked on this aspect of the application. They solved the solution by creating a user class, which had variables to store the user's choices. Another technical challenge was coming up with a way to create a timer. We knew from the beginning the creating a timer was going to be difficult. After researching ways to implement a timer I decided to use multithreading and a GLabel that would update every 1000 milliseconds. This solution was somewhat simple to implement by hardcoding into a class, but in order to get the timer to work properly with the custom workout portion of the project that Josiah was working on I needed to place the timer in a class.

2. Describe how the software engineering techniques you learned in this course helped you to address those challenges.

We used various software engineering techniques in order to simplify our code and address the above problems. One software engineering technique that helped with user choices problem was abstraction. We abstracted the variables of the user class that

we used to get the users choices. We also used abstraction with the timer by creating a timer class. Using the UML and class diagrams helped us stay on track when addressing the problems that we encountered when programming.

3. Describe what other knowledge you feel might have helped you with the project development.

I feel I had a good understanding of eclipse since we had a lecture on it, but it might have been even more helpful if we had a lab in which we just learned how to use eclipse commands. Also a greater understanding of both the awt and swing libraries would have helped me. More specifically, ways to implement media including animations, video, and sound. I could also have used a greater understanding of how to create interactions diagrams.

4. Discuss possible directions for the future work on this project.

For the future one possible direction for the project is to implement features that we didn't have time to implement during our initial project time. We also didn't get to add video to the workouts pane. Adding video would require use of the swing library. We also wanted to create a phone application, this was obviously a bit overzealous. In the future though the code we wrote for this application could be used in creating an Android application.

5. Give at least one piece of advice to give future COMP 55 students that you wish you would have known when you started that would help improve your future project.

Future COMP 55 students should spend a considerable amount of time creating their Requirements Analysis and System Design Report, specifically the UML and plan of work. The UML, while our group didn't follow it exactly as we thought, it gave us a good indication of what needed to be created. The plan of work is also important. Splitting up tasks early on and deciding on times when everyone can work on the project is crucial. I would also suggest that future COMP 55 students spend at least 30 minutes a day programming when that phase of the project starts for them.

Isabelle Tran:

Working in a group brought up many technical challenges I had never encountered up until this project. The ability to all maintain the same vision throughout the project may have been the biggest challenge. When individuals lack the same common goal coding began to go awry. The issues arose when group members needed to integrate aspects of another member's code realizing that we did not all have the same understanding of the functionality the code we were attempting to implement. Additionally, with each person comes a unique coding style that without proper communication led to improper implementation of certain code. Knowing and understanding all the code that you did not contribute is just as important as understanding your own code because it will allow for better coding practices on your end to provide a seamless transition between group members work creating a cohesive product.

There were many programming practices in hindsight that would have been very useful to our group had we had a better sense of what direction we wanted to head in. I believe with a more proficient use case diagram we could have foreseen a lot of the issues we had to address as

we were coding in order to maintain a user-friendly application. Additionally, our original UML reflected the lack of vision we had initially because we did not foresee all the behind-the-scenes classes we would end up creating. I think better communication for what we wanted our application to provide would have allowed us to make a better UML. Enumeration of strings helped heavily with this project because there were many references to type of workout, muscle group, and difficulty that could have been an issue had we left them as strings.

It would have been very helpful to have more tutorials on Git and all the benefits it provides for groups. The utilization of branches would have been helpful earlier on as our group had a lot of errors when working on the same class at the same time which would have been avoided with more tutorial on branches.

The future of this workout application is limitless. I was in the process of creating a workout suggestion feature that would encourage users after a set number of workouts at a set difficulty to attempt a higher difficulty level workout. Conversely, it would provide warnings to users who attempted too high of a difficulty too quickly to reduce risk when using our application. Our group decided to work on other aspects of the application rather than adding embedded videos, but I do see that as a potential for growth in this program. It would enhance user experience by providing them visual for proper form and coaching, providing them with a virtual personal trainer. Eventually, this application would be best used on a handheld device as users do not normally bring laptops to the gym.

Advice I give to others is to keep an open mind and vocalize your ideas and concerns. If you need help, do not be afraid to look dumb in front of your group members. It is better to be honest and up front about your capabilities and your commitment to the project than to take on too much and not deliver.

Kyle Hartman:

We ran into a couple technical challenges that were pretty tough to figure out, but were great features to include in our project, such as our timer, figuring out how to play and pause and even reset the timer was a pretty tough challenge that took hours of research to figure out and get it to work. My favorite challenge that we had but was great when we figured it out, was the notes section of our program, for a while we had issues with saving and loading in the text file and then we had problems with getting the text to wrap around the text box so that it didn't scroll on forever.

The multiple diagrams such as the UML class diagram and the interactions diagram were very helpful in understanding how our project would work and function. These diagrams allowed us to plan what we wanted to do and as we were working on the project and adding new things or removing old stuff, it was great to be able to have a diagram of sorts to help us along the way and make sure we remained on track and that we didn't stray too far from our original goal.

On that note, I think more discussion on how these diagrams work and how they are supposed to help us would be great, I felt myself googling too much outside of class to understand what these diagrams were supposed to help us with. The book was helpful, but having more in class discussion on these diagrams would help fill the gaps the book left out. Another thing that could be added is maybe more group work before the start of the group project, I feel that would allow each member to get a better feel for each member of the group.

A great direction would be to further improve how the user is able to make a custom workout with more features and easier selection of the workouts. Maybe include multiple text

files the user can access and use. I would love to add animation to the workout images we have, to make it easier for the user to visualize what they are supposed to do. Better looking statistics would be a huge plus, I felt like the colors of the histogram bars didn't really go well with the look of jump start.

One piece of advice I would give is to be open with your group and make sure that you all are on the same page and that you all feel like each of you is contributing fairly. Communication is key to any group project. If one of you is not on the same page or isn't really contributing as much as they should be then it needs to be addressed and talked to within the group.

Josiah Yoshimura:

One of the biggest challenges I faced was finding a solution that fit with our framework. A good portion of the solutions I came across all used Swing. Though the acm graphics library, and awt seem fairly easy to use; it seems that swing might be a slightly more difficult, but more robust option. Our notes pane needed a text area in which the user could type up something and save it. This was implemented using an AWT text area. The text area had its own constructor that specified where it should go, but since our overall framework used ACM it did not matter where we specified to place the text area. We had to use the ACM functions to specify where we wanted the text area. Looking back this seems fairly obvious, but at the time it was quite the headache.

Additionally, some of the most challenging parts of the project were figuring out how to make pieces of the project work together. The single responsibility principle is quite helpful for simplifying classes. It allowed me to divide some of our classes up and then figure out how to

connect the pieces. A good example was the statistics pane. At first I thought it would handle everything from the data to the histogram. In the end it had a histogram object which obtained its data from methods from a file IO class. Another class with statistics functions also acted as a helper.

I think the biggest thing that could have improved the project would be taking the planning more seriously. Though I did check to see if there were solutions to the things we wanted to implement, I wish I would have thought more deeply about the relations between the components of our project. In the future it would be interesting to try and figure out a more sophisticated way of storing data. We used an almost overly simple file, but if we had store the file with a way to distinguish between data from different times it would make for a more interesting statistics pane. Perhaps one could just store multiple files, but it feels like a waste of files and unnecessary clutter.

References

<http://www.makeoverfitness.com/>

Workout graphics came from this website

<https://docs.oracle.com/javase/tutorial/uiswing/components/dialog.html>

Dialog boxes incorporated in workout

<https://docs.oracle.com/javase/tutorial/uiswing/components/icon.html>

Custom icons used for dialog boxes

<http://stackoverflow.com/questions/4172940/how-to-set-background-color-of-a-button-in-java-gui>

Adding color to dialog boxes

<http://stackoverflow.com/questions/14393423/how-to-make-a-countdown-timer-in-java>

We used this link to help us figure out how we would approach creating a timer.

<http://stackoverflow.com/questions/15438818/printing-a-word-every-x-seconds>

We used this link to figure out how to print a JLabel every number of seconds.

<https://www.cis.upenn.edu/~bcpierce/courses/629/jdkdocs/api/java.awt.TextField.html>

Resources to implement text field that will eventually save user input to a file. Used this for the notes pane.

http://www.tutorialspoint.com/awt/awt_textarea.htm

Using a TextArea from AWT library was better. Used this for the notes pane.

<https://examples.javacodegeeks.com/core-java/util/timertask/java-timer-and-timertask-example-tutorial/>

Was used to figure out how to create a timer.

<http://www.theguardian.com/lifeandstyle/2011/jan/07/fitness-regime-stay-motivated>

Used in order to find information on workouts and terms.

<http://www.bodybuilding.com/>

Used to find workouts that we could use in our application.

<https://human.biodigital.com/signin.html>

Used for more information on terms and workouts.

http://www.tiim.info/the_nxt_project/web/design.php

<http://www.ibm.com/developerworks/rational/library/content/RationalEdge/sep04/bell/>