Sprint Review #4

LiftingPal

Sprint Review Report
Sprint 4
Version 1
December 5th, 2018

Software Engineer
Angel Chang
Abhi Inuganti
Robert Rozin
Andrew McLaren
Drew Kozak

Prepared for
CS 1530
Fall 2018
University of Pittsburgh

Revision History

Date	Description	Author	Comments
12/3/18	Version 1	Team	First Revision

Document Approval

The following Software Requirements Specification has been accepted and approved by the following:

Signature	Printed Name	Title	Date
	Angel Chang	Software Eng.	12/5/2018
	Abhi Inuganti	Software Eng.	12/5/2018
	Robert Rozin	Software Eng.	12/5/2018
	Andrew McLaren	Software Eng.	12/5/2018

	Drew Kozak	Software Eng.	12/5/2018

Table of Contents

1. Introduction

This sprint consisted almost entirely backend work. We implemented the algorithms that would generate the workouts for the different user skill levels as well as for raw or equipped advanced users. We also implemented the second questionnaire for users to fill out after completing their workout block. Finally, we created the user class in a different way than we originally plan, creating a user table in the database. We originally planned to complete 28 story points but we only are accepting 18 of those points as done. We could only get our database to function correctly when run locally, but that can be attributed to no one in our group having any database experience. Overall the last sprint brought the project together, finally making core components of the system output to the site.

2. Specific Goals

2.1 Implement Beginner Program Maker

2.1.1 Story Description:

Created a program for beginner lifters based on their pre-block questionnaire.

2.1.2 Story Acceptance Criterion

Create a block for beginner lifters based on the inputs stored in the SQL server.

2.1.3 Story Dependencies

Pre-block page and functionality

2.1.4 Story Challenges

It was difficult to link the HTML, the Java, and the SQL together. An external library had to be added for full functionality.

2.1.5 Story Assigned to

Robert Rozin
Andrew McLaren

2.1.6 Story Points

1

2.1.7 Status: Completed or not

Complete

2.2 Implement Intermediate Program Maker

2.2.1 Story Description:

Created a program for intermediate lifters based on their pre-block questionnaire.

2.2.2 Story Acceptance Criterion

Create a block for intermediate lifters based on the inputs stored in the SQL server.

2.2.3 Story Dependencies

Pre-block page and functionality

2.2.4 Story Challenges

It was difficult to link the HTML, the Java, and the SQL together. An external library had to be added for full functionality.

2.2.5 Story Assigned to

Robert Rozin
Drew Kozak

2.2.6 Story Points

2

2.2.7 Status: Completed or not

Completed

2.3 Implement Advanced Raw Program Maker

2.3.1 Story Description:

Created a program users who classified as Advanced and are not equipped on their pre-block questionnaire.

2.3.2 Story Acceptance Criterion

Create a block for Advanced Raw lifters based on the inputs stored in the SQL server

2.3.3 Story Dependencies

Pre-block page and functionality

2.3.4 Story Challenges

It was difficult to link the HTML, the Java, and the SQL together. An external library had to be added for full functionality.

2.3.5 Story Assigned to

Robert Rozin
Angel Chang
Abhi Inuganti
Andrew McLaren
Drew Kozak

2.3.6 Story Points

4

2.3.7 Status: Completed or not

Completed

2.4 Implement Advanced Equipped Program Maker

2.4.1 Story Description:

Created a program users who classified as Advanced and are equipped on their pre-block questionnaire.

2.4.2 Story Acceptance Criterion

Create a block for Advanced Equipped lifters based on the inputs stored in the SQL server

2.4.3 Story Dependencies

Pre-block page and functionality

2.4.4 Story Challenges

It was difficult to link the HTML, the Java, and the SQL together. An external library had to be added for full functionality.

2.4.5 Story Assigned to

Robert Rozin
Angel Chang
Abhi Inuganti
Andrew McLaren
Drew Kozak

2.4.6 Story Points

4

2.4.7 Status: Completed or not

Completed

2.5 Post-Block Questionnaire

2.5.1 Story Description:

The user will have to fill out a post-block questionnaire after completing their first block generated by the pre-block questionnaire.

2.5.2 Story Acceptance Criterion

A block questionnaire can be filled in by the user and the results will be stored

2.5.3 Story Dependencies

The user needs to finish the initial block that was generated by the pre-block questionnaire to work on the post-block questionnaire.

2.5.4 Story Challenges

It was difficult to link the HTML, the Java, and the SQL together. An external library had to be added for full functionality.

2.5.5 Story Assigned to

Robert Rozin
Abhi Inuganti
Angel Chang

2.5.6 Story Points

4

2.5.7 Status: Completed or not

Completed

2.6 User Class

2.6.1 Story Description:

Create a user that would store all of the answers to the questionnaire for the program makers.

2.6.2 Story Acceptance Criterion

A database stores users and all of their answers to the questionnaire.

2.6.3 Story Dependencies

N/A

2.6.4 Story Challenges

No one had created a database from scratch before, but some of us had experience with SQL. Generating the user tables with the proper columns was not too difficult.

2.6.5 Story Assigned to

Robert Rozin
Angel Chang

2.6.6 Story Points

3

2.6.7 Status: Completed or not

Completed

2.7 Program Block Navigator

2.7.1 Story Description:

Allow users to view the current work out and past work outs they had done.

2.7.2 Story Acceptance Criterion

A page is generated that stores all of the work outs a user has done and allows for easy navigation.

2.7.3 Story Dependencies

2.1, 2.2, 2.3, 2.4, 2.5

2.7.4 Story Challenges

Due to most of the team being very new to HTML, it was a learning experience and an uphill battle to get a lot of things done. We simply didn't have the technical knowledge or experience to do this.

2.7.5 Story Assigned to

Robert Rozin
Angel Chang
Abhi Inuganti
Andrew McLaren
Drew Kozak

2.7.6 Story Points

3

2.7.7 Status: Completed or not

Completed.

2.8 Cause HTML to run the ProgramMakerRunner class

2.8.1 Story Description:

The HTML needs to execute the ProgramMakerRunner.java file once the questionnaire is filled out so the block does not need to be generated by hand every time.

2.8.2 Story Acceptance Criterion

The program block is automatically generated when the submit button on the questionnaire is clicked.

2.8.3 Story Dependencies

2.1, 2.2, 2.3, 2.4, 2.5

2.8.4 Story Challenges

Had never linked HTML and Java before, so it was a learning experience. Unfortunately, we ran out of time and had never created a java applet either.

2.8.5 Story Assigned to

Robert Rozin
Angel Chang
Abhi Inuganti
Andrew McLaren
Drew Kozak

2.8.6 Story Points

4

2.8.7 Status: Completed or not

Not completed, did not have enough time. Possibly look into making an applet so the html can run the file.

2.9 Move the Database Online

2.9.1 Story Description:

Get the database running off an ip that isn't localhost.

2.9.2 Story Acceptance Criterion

The database doesn't need to be ran locally for the program to work.

2.9.3 Story Dependencies

N/A

2.9.4 Story Challenges

No one in our group had implemented a database on our own before, so it was an entirely new experience for us. Unfortunately, this story remained unfinished due to it.

2.9.5 Story Assigned to

Robert Rozin
Angel Chang
Abhi Inuganti
Andrew McLaren
Drew Kozak

2.9.6 Story Points

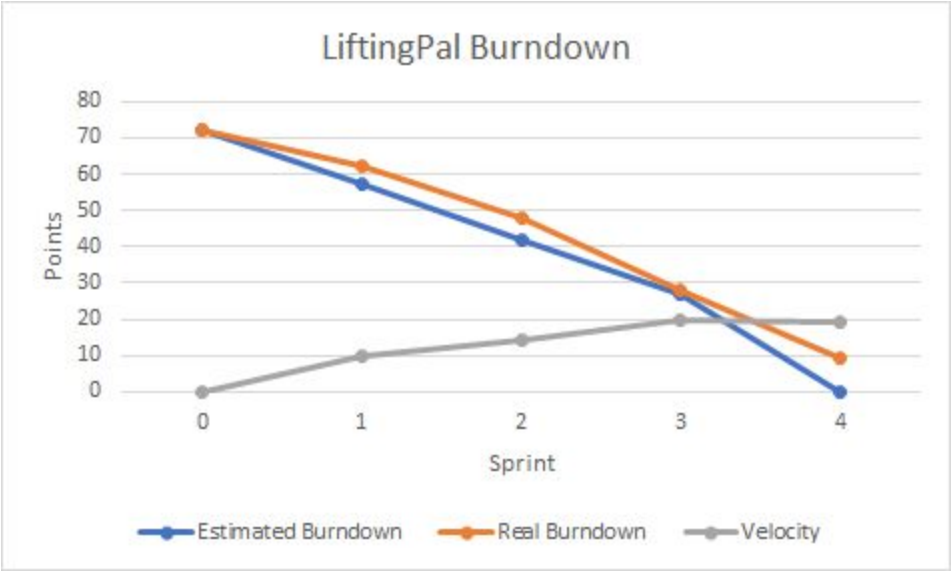
3

2.8.7 Status: Completed or not

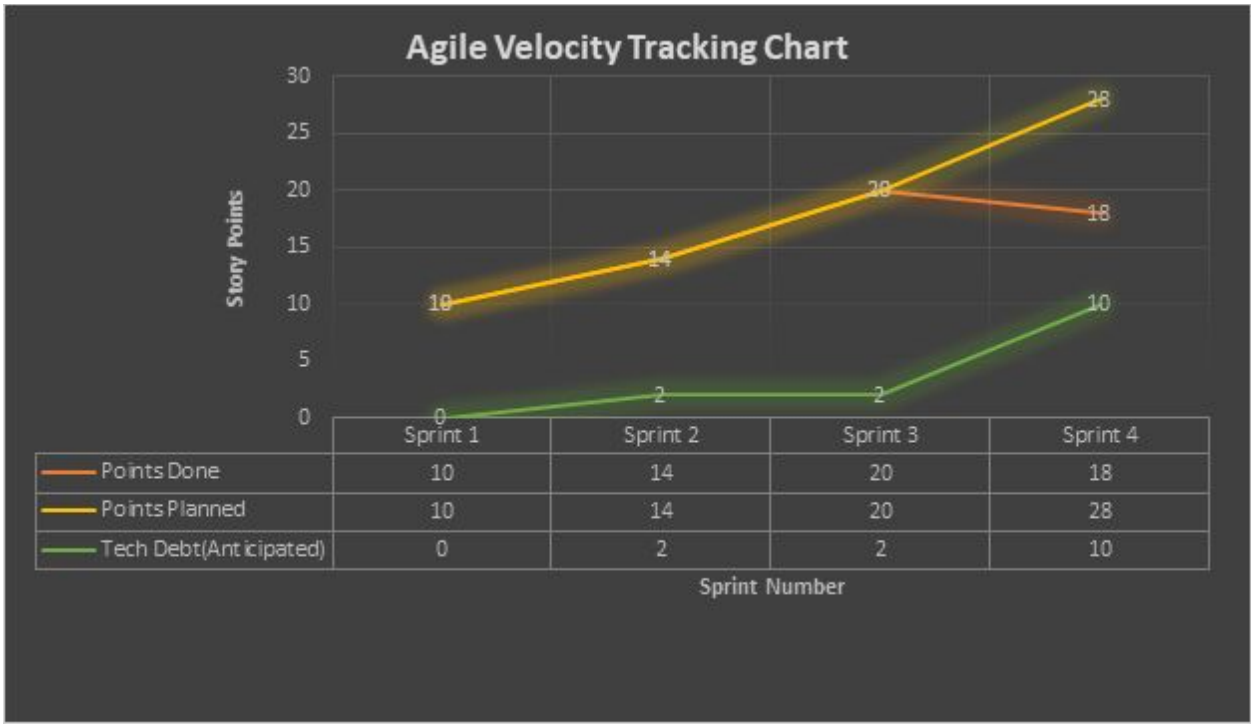
Not Completed. Will do more research to fix this and implement a proper online database.

3. Analytics

3.1 Sprint/Product Burndown Chart (sample chart shown below)



3.2 Sprint Velocity Chart



4. Conclusion

In conclusion, we have finished up our backend as much as we could during this sprint to meet the deadline for this sprint. We have implemented our relevant algorithms to generate the workouts that are suited for specific sorts of users based on their skillset. We have changed some of our initial ideas during this sprint due to a lack of time, and because we felt some of our older ideas are simply infeasible at this point in time. The user class is an example of this occurrence. We created a user table inside our database rather than creating a user class in Java as originally planned. We have finished 18 points from our initial 28 story points that we have hoped to finish. We did not plan on hosting our database locally but that was always our backup plan in case we could not meet the deadline and programming issues. Much of the backend work in this sprint was especially difficult and time consuming as no one in our group had much experience with databases. All in all, we have finished our projects and successfully made it to the deadline with functionality despite having to give up on some early targets that we have set for ourselves due to complexities and deadlines.