

Capstone Part A

TJ Riker

Business Problem	3
Software Development Methodology	4
Deliverables and Implementation	4
Meeting Requirements of Users	4
Programming Environment	5
Timeline	5

.

Business Problem

The problem that this application is trying to solve is to help consumers get a better understanding of their overall health and fitness, related to diet, weight, and water consumption. The expectation would be that after a user has a better understanding of their fitness habits, they could more easily take control of their health and work towards their goals of eating a certain amount of calories, protein, carbohydrates, fat, drink a certain amount of water, and reach their goal weight. The purpose of this project would be to help people be healthier and more informed on their diets and weight. The daily weigh in feature would inform the user on how their current weight is trending throughout the days and weeks. The user could also input the amount of calories they have eaten that day and the amount of water consumed. The point of this is to allow the user to be informed of what they are eating and be able to more easily make changes in their body composition. If they are tracking water, and they notice that they are not drinking as much per day as they would like, they would have a good tool to help increase amount consumed. Daily tracking of calories and weight will let the user understand how many calories per day they eat affects their overall weight. If they want to lose weight, they could reduce the amount of calories they are consuming, and by the daily weight tracking, they can see exactly how those calories are affecting their weight. This can all be done through guesswork and trial and error, but with an application to help track, the data is concrete and you will know exactly how much to eat and drink to reach a goal weight. This application would take the guesswork out of losing or gaining a specific amount of weight in a certain amount of time.

Software Development Methodology

The software development methodology being used to develop this application is the Agile method. I believe this method fits this type of application development the most, more than Waterfall method specifically. The agile method allows for quick changes and updates and allows for fast progression and sudden changes. As I am working through developing the application, I have new ideas and changes I would like to implement. The agile methodology allows me to do this without taking a huge hit to time and productivity. I can constantly iterate through new versions and make changes and updates as often as I need. The agile methodology also allows for testing whenever it seems fit. With all of the changes and updates I have been making, the ability to test whenever I please has been very helpful in making sure everything works smoothly and detecting bugs as soon as they appear.

Deliverables and Implementation

The deliverable of this entire project is to be a fully functioning android application. One that can track weight, calories, and water consumption. This application will not be deployed to the public app store, so I will be testing and implementing this app to an android phone through local deployment of an APK. The anticipated outcome is to be a fully developed application that works on any local device. The APK could be installed by anyone who has it for testing and usage.

Meeting Requirements of Users

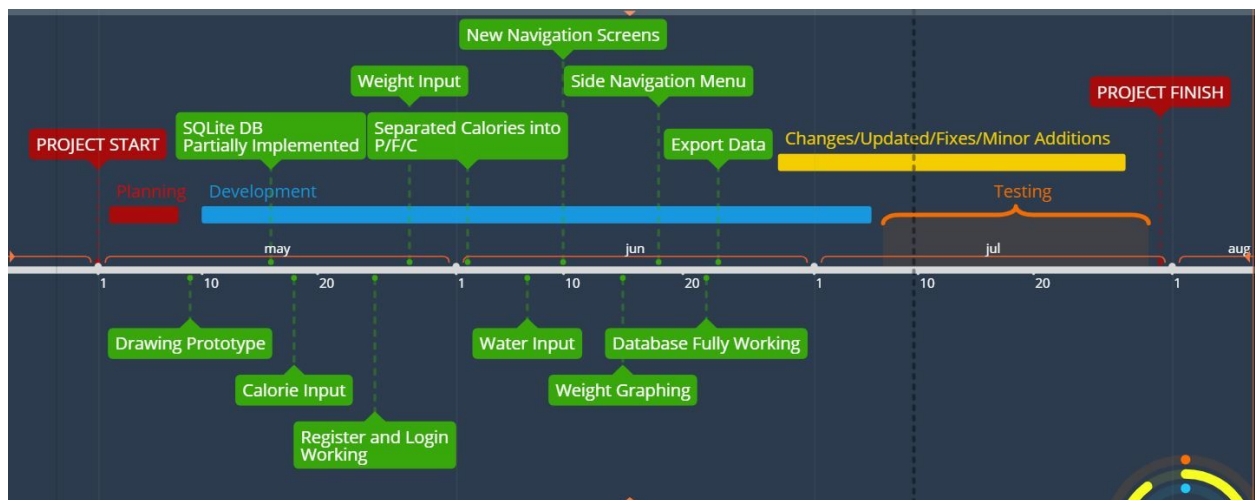
The requirements of the users would be met through updates and fixes. This is also why the agile methodology is a good idea for this type of application. The users of this application could suggest features to be added or bugs that need to be fixed. Then, because of the agile method, it is very easy for me to iterate new versions of the

application, creating new features, updates, fixing bugs, etc. Then a new version of the APK would be deployed to the users for further testing.

Programming Environment




















































The environment I have for developing this app is a fairly old pc. This is the main restriction I have when trying to develop. The pc is fairly old, so compiling times are usually quite lengthy. The PC I am using also has an AMD processor, which means that Android Emulation through Android Studio does not work. This requires me to use an older android phone that I purchased to test the application. Rather than being able to test multiple devices, frameworks, updates, etc, through an emulator, I have to test on one local device through the installation of an APK after every change. This makes testing take far longer, and much less efficient and effective. If I was to create another app, it would be incredibly helpful to have a PC with an Intel processor, purely for the use of android emulation.

Timeline



The only real dependencies for this application was the development of at least a basic working database first. Nearly everything in this application relies on a working database, so that had to be developed first. The resources applied to each task were basically all the same. When I started on a certain task, I worked on it when I could and

when I was finished, I moved onto the next task that I felt would fit in nicely next, though there was no real order that the tasks had to be implemented.

	PROJECT START	Description	1 may 2019	 
	Planning	Description	2 may 2019	8 may 2019  
	Drawing Prototype	Description	9 may 2019	 
	Development	Description	10 may 2019	6 jul 2019  
	SQLite DB Partially Implemented	Description	16 may 2019	 
	Calorie Input	Description	18 may 2019	 
	Register and Login Working	Description	25 may 2019	 
	Weight Input	Description	28 may 2019	 
	Separated Calories into	Description	2 jun 2019	 
	Water Input	Description	7 jun 2019	 
	New Navigation Screens	Description	10 jun 2019	 
	Weight Graphing	Description	15 jun 2019	 
	Side Navigation Menu	Description	18 jun 2019	 
	Database Fully Working	Description	22 jun 2019	 
	Export Data	Description	23 jun 2019	 
	Changes/Updated/Fixes/Minor Additions	Description	28 jun 2019	28 jul 2019  
	Testing	Description	7 jul 2019	30 jul 2019  
	PROJECT FINISH	Project must be turned in by this time	31 jul 2019	