|  |  |
| --- | --- |
|  | Fitness App |
|  |  |
|  | Margaret and Shubha  README FILE  9/5/18 |

**THE BODY ACHIEVES WHAT MIND BELIVE FITNESS APP**

Main objective: To include diversity and provide app focusing on diverse need of society ex. Busy professional, senior citizen and young people.

1.Easy to follow program according to the needs

2.Use on online resource (video lead training) and getting feedback from the customer to make our plan more efficient

# FUNCTIONALITY

1.Flexibility and different planes for everyone

2.Keeping customer data for data processing

3.Body mass Index calculation

4.Attandence calculate

5.Analysising the data before and after the program

# *Instruction to use*

* Join us with your physical details.
* Plan be suggested, and you have option to choose.
* Enter the program and we are going to take your BMI (body mass index) before the program to see best possible plan according to your need.
* Taking the weight and attendance to see your improvement
* Provide feedback and encouragement along the way of our program

Please find the given flow diagram for better understanding of the program flow

## 

If not than keep working hard : growth mindset

If BMI is less than before ::: Good job

After fitness program BMI(Again taking inputs from user

**COMMENT:** Successful implemented and **tested** during the designing phase.

# Flow chart 2

**COMMENT:** Attendance calculator is implemented but has a scope of improvement for further modification.

# Flow Chart Three

Equal than or more

“Follow our program and growth mindset advice”

Less than before

“GOOD JOB”

‘

BMI calculation after the program

**COMMENT**: Starting phase of the app has calorie count but could be implemented because of the time limit. Something to incorporate for a **future use.**

# **Brainstorming process**

Initially upon beginning to brainstorm we thought of coding a simple brute force password cracker but soon felt it would be a more elegant program to write a fitness application instead. We chose a fitness app for the purposes of including people that would not usually be the target audience of a fitness application.

Senior citizens, working professionals and busy mothers. As well as general health and wellbeing for the general public. We also chose a diversity focused application in order to help senior citizens to become more social and improve their overall happiness and quality of life.

**COMMENT: Have image included in doc folder **

# **Challenges**

We initially faced a challenge of a file that was becoming too clustered and illogical with a large amount of code. We solved this challenge by sorting the **classes and tests into different files** and condensing the code in the program by linking to the files instead of hard coding in the methods.

Another challenge faced was how to exit the program if the wrong user input was passed in by user to select the fitness plan. we decided to go with the “abort” method with “puts” so user can get the informed message on the screen saying “wrong input is supplied “and program get terminate at that point itself.

Other example of challenges includes

1. Running of test driven development during the development of the app.
2. Debugging the program especially the edge cases where user input could be different than expected. There is lot of scope to test those kind of edge cases by doing the test-driven development.

* Ex. BMI calculation input metric system is in kg and cm, but there is possibility of user entering data in other metric, resulting in incorrect BMI calculation.
* Attendance calculator just manipulate lower case “y” as an input we can have more test-driven development to insure user get proper error message on screen.

# **Code structure**:

Final code is stored in weightloss\_app.rb file.

Code structure flows as below mentioned point:

* Collected data (user information) get stored in Fitness Program file and addition to that BMI calculation is store in same file.
* 3 Options were given to user to select from first is “Busy Bee” second is “old is Gold” and last one is “average joe”. Every plan has different ruby file stored according to the plan name.

(busy \_bee.rb ect.)

* If user has selected the first option in current ruby file, we can execute the code written for busy bee plan with the help of require \_relative functionality as same goes for other two option as well.
* At the end user get the feedback according to the amount of effort been put during the fitness plan.

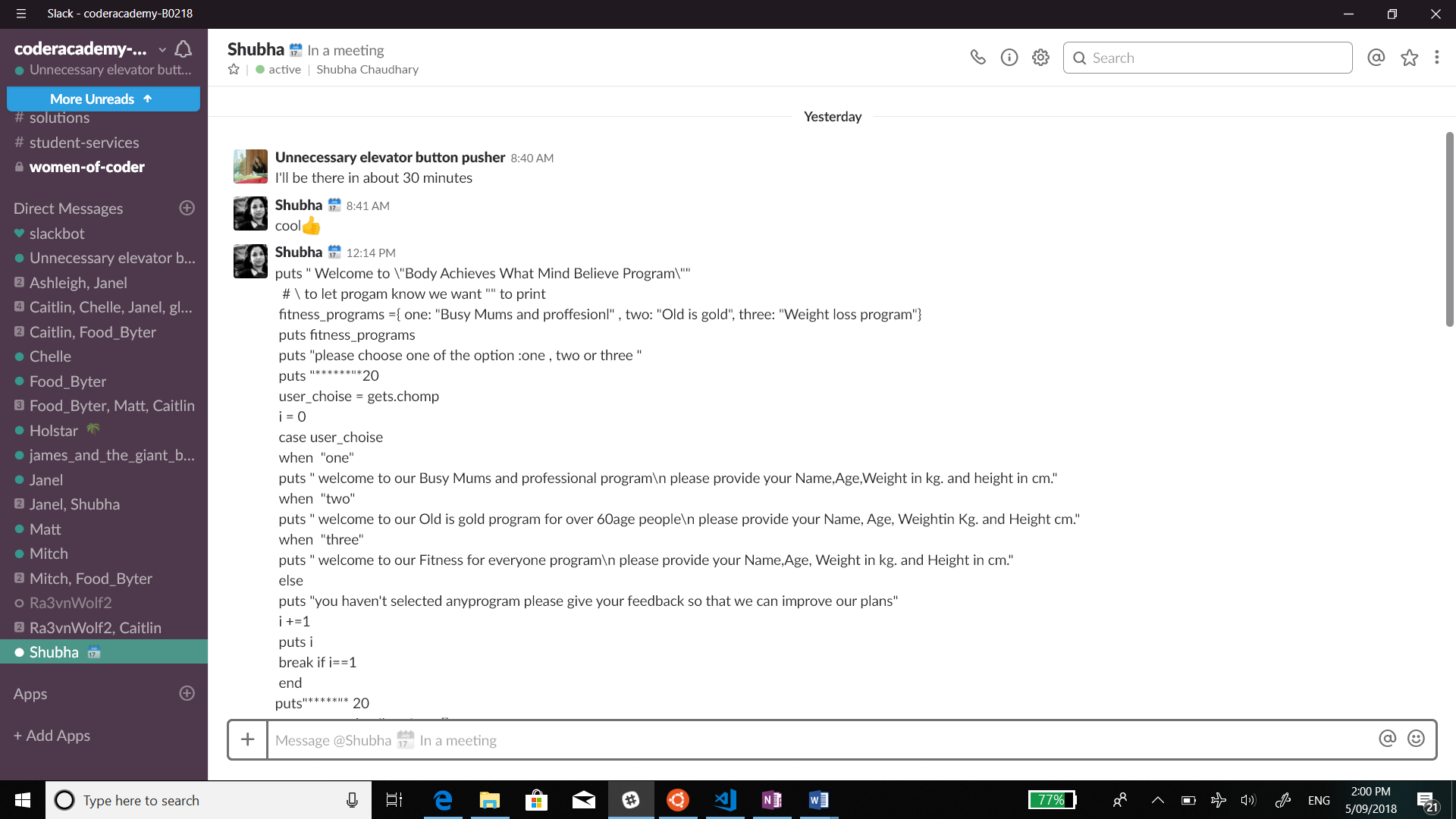
**Comment**: debug.rb file is for manual testing purposes.

# **User story**

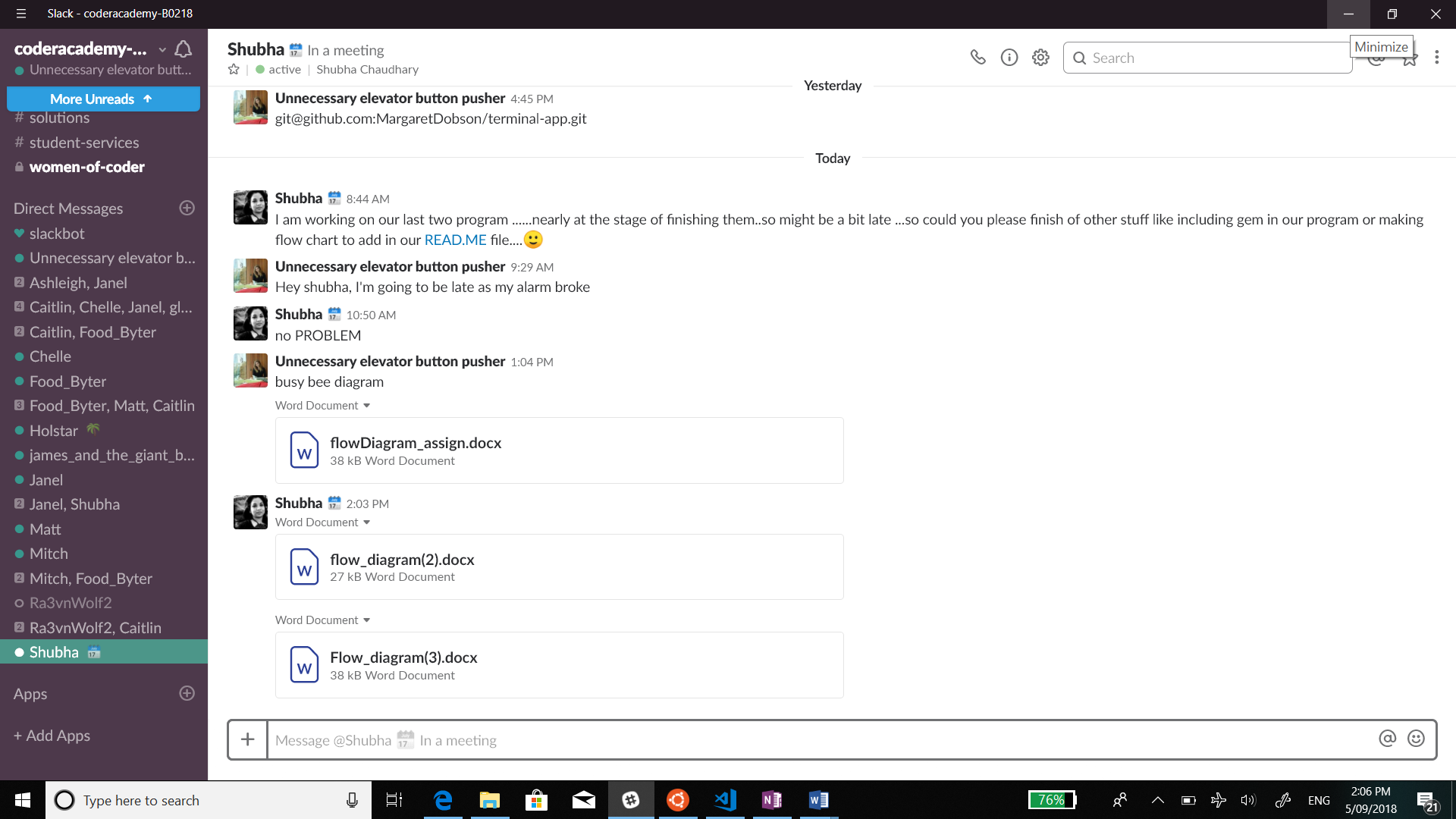
A user story that is of, for example a senior citizen or a busy professional. They feel that there is no technology that specifically includes them until they find our application. And use it to attend or try new programs they could potentially enjoy and give us feedback as to how the program has affected their life.

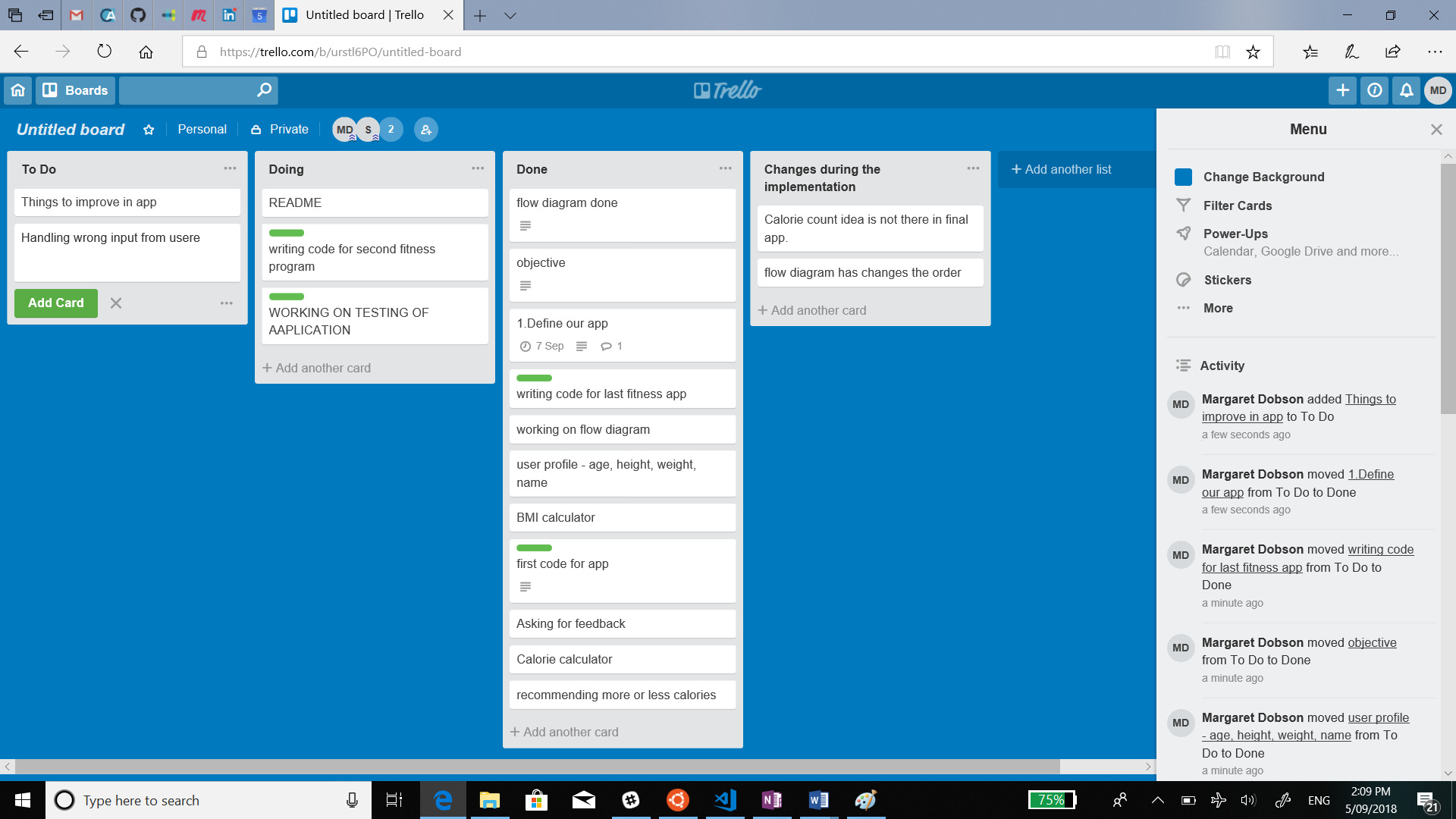
# **Communication Process**

We communicated mainly in person, on slack and also communicated through Trello



# 





* Going through the tools brainstorming
* Slack communication
* Git link

# **Time issues**

These are issues that we had due to the two-day time constraint

* We were going to add a calorie count tracker but didn’t have enough time.
* Writing more test for the app. Specially for the **edge case**.
* More structured and better coding practice to include be the next possible step after having running app.

# **resources**

Github link: <https://github.com/MargaretDobson/terminal-app>

* cloning : git@github.com:MargaretDobson/terminal-app.git)
* Main code: Weightloss\_app.rb .
* Different plans and data: busy\_bee.rb,old\_is\_gold.rb,average\_joe.rb,fitness\_programe.rb
* Debugging: debug.rb
* test file: busy\_bee\_test.rb,old\_is\_gold\_test.rb,average\_jo\_test.rb,fitness\_programe.\_test.rb
* image: download.jpg for gem