Run Together Project Final report

Group 8

Prepared by:

Madhava Sai Yamike, Matt Selvaraj, and Jibreel Mohamed

April 2022

Table of Contents

I Project Description

1 Project Overview

Run Together is an app that focuses on combining personal fitness and social networking functionalities. It is meant to be used by users who participate in group running to communicate with friends and it allows them to also track their running data. Users can find groups and friends through the following methods: in-app, Facebook, or SMS searches. The app will allow the users to schedule runs with their friends and groups though the chat poll features. The app focuses on synchronous group activities, but it also allows users to run in individually so users can use the same app everyday. By being able to run with friends, users will be able to reach their running goals by holding each other accountable to a schedule. Whether users go out for runs individually or in a group, they can set challenges or run-in charity runs.

2 Project Domain

Personal health and fitness is the major domain of Run Together. The app is designed to provide runners with a method to track their fitness progress while also staying in the loop with other runners. Overall, people should be able to use the app to maintain or improve their personal health and fitness.

3 Relationship to Other Documents

The final report will explain what components of the initial project report were implemented successfully and how they can be tested and verified. Additionally, we will demonstrate how the final product represents the implementation plan specified over coding scenarios one and two.

4 Naming Conventions and Definitions

4a Definitions of Key Terms

Individual Run: A run counter that is generated by the user that can be used to time their run.

Add Friend: Functionality used to add another user as a friend through the app.

Text Friend: Functionality used to text a friend on the app.

Check Messages: Functionality used to check existing messages on the app.

Check Friends: Functionality used to check your friends on the app.

Check Friend Requests: Functionality used to accept friend requests from other users on the app.

4b UML and Other Notation Used in This Document

This document follows UML 2.4 standards, the IEEE Citation Style, and the Volere snow card. Smaller uses of other notations are noted where they occur.

4c Data Dictionary for Any Included Models

User profiles contain a username, password, and email. The constraints are that all usernames and emails must be unique, and the password must be greater than 8 characters in length.

II Project Deliverables

Our Group has successfully implemented the features of the login/sign up function to access the app. Users can add/find friends to get to know each other. Users can send/receive text messages to any person in the friends list. Users can also schedule individual runs and reach their fitness goals.

1 First Release

The date of first release was February 25th 2022. The main functionalities of the system was the user can sign up or login into the app. The user can find and add the friend to the user friends list.

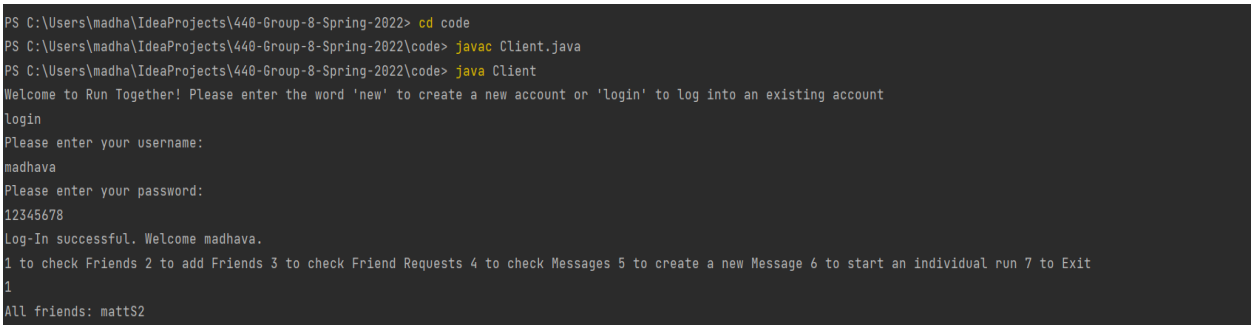

Figure 1. First release demo

2 Second Release

The date of first release was April 1st 2022. The main functionalities of the system was the user can send friend requests to the people who they know. Users can see the friend requests on their account and can accept it. Users can send/recieve a message from the user's friends list. The user can schedule an individual challenge run and keep track of their data.

```
Welcome to Run Together! Please enter the word 'new' to create a new account or 'login' to log into an existing account
login
Please enter your username:
mattS2
Please enter your password:
asdfjkl;
Log-In successful. Welcome mattS2.
1 to check Friends 2 to add Friends 3 to check Friend Requests 4 to check Messages 5 to create a new Message 6 to start an individual run 7 to Exit
4
Users you have texts with: madhava(Unread)
Please enter a user to text with:
madhava
Most recent message: madhava: Hi!
Please enter your reply:
Hello
Message sent successfully!
```

Figure 2. Second release message demo

```
Welcome to Run Together! Please enter the word 'new' to create a new account or 'login' to log into an existing account
login
Please enter your username:
madhava
Please enter your password:
12345678
Log-In successful. Welcome madhava.
1 to check Friends 2 to add Friends 3 to check Friend Requests 4 to check Messages 5 to create a new Message 6 to start an individual run 7 to Exit
5
All friends: mattS2
Enter a username to send a new message to:
mattS2
Please enter your message for: mattS2 and enter to send.
Hi!
Message sent successfully!
```

Figure 3. Second release message demo

```
29
28
27
26
25
24
23
22
21
20
19
18
17
16
15
14
13
12
11
10
9
8
7
6
5
4
3
2
1
Run Complete!
```

Figure 4. Second release individual run demo

3 Comparison with Original Project Design Document

Our prototype provides a mix of both social media and fitness features such that it is representative of a minimum viable product. The prototype does not feature group runs, charity runs, or login API integration, because we believed that they were not essential features to the core of the minimum viable product. Additionally, our third group member was unable to write a single line of code over the course of the 16-week semester, leaving our group at essentially two coders (we are a group of three). As such, our prototype does not feature a graphical user interface.

III Testing

1 Items to be Tested

Use-case ID 1: Sign-up

Use-case ID 2: Log in

Use-case ID 3: Message Someone

Use-case ID 4: Start a run

Use-case ID 5: Add friend

2 Test Specifications

ID#1 - Sign-up

Description: This product must allow a new user to sign up

Items covered by this test: Use-case ID #1

Requirements addressed by this test: Registration (*Group10Developmentfinal*)

Environmental needs: NA

Intercase Dependencies: NA

Test Procedures:

- run program
- enter 'new'
- enter a valid username, email, password

Input Specification: Username must be available, password must be at least 8 characters long, email must contain '@'.

Output Specifications: “New user created”

Pass/Fail Criteria: New user data is loaded in the database

ID#2 - Sign-in

Description: This product must allow an existing user to sign up

Items covered by this test: Use-case ID #2

Requirements addressed by this test: Registration (*Group10Developmentfinal*)

Environmental needs: NA

Intercase Dependencies: Must be existing user (ID #1)

Test Procedures:

- run program
- enter “login”
- enter the username
- enter password

Input Specification: Username and password inputted must be an account that has been created.

Output Specifications: “login successful”

Pass/Fail Criteria: menu options should be visible.

ID#3 - Message Someone

Description: This product must allow a user to send and check messages

Items covered by this test: Use case ID: 3

Requirements addressed by this test:

Message someone (*Group10Developmentfinal*)

Environmental needs: NA

Intercase Dependencies:

- User must be logged in: **ID#2-Sign-in**
- User must be friends with the user being communicated with: **ID#5 - Add Friend**

Test Procedures:

- run program
- enter ‘4’ or ‘5’
- enter the username of a friend
- type in a message
- send

Input Specification: Username inputted must be a friend of the user.

Output Specifications: “Message sent successfully”

Pass/Fail Criteria: Message data is loaded in the database

ID#4 - Start a Run

Description: This product must allow a user to start a run

Items covered by this test: Use case ID: 4

Requirements addressed by this test: Start a Run (*Group10Developmentfinal*)

Environmental needs: NA

Intercase Dependencies: User must be logged in: **ID#2-Sign-in**

Test Procedures:

- run program
- enter ‘6’

Input Specification: NA

Output Specifications: “Starting 30 second run”

Pass/Fail Criteria: A 30 second timer should start

ID#5 - Add Friend

Description: This product must allow a user to add another user as a friend

Items covered by this test: Use case ID: 5

Requirements addressed by this test:

ID#14 - Friends List Capacity (*Group10Developmentfinal*)

Environmental needs: NA

Intercase Dependencies: User must be logged in: **ID#2-Sign-in**

Test Procedures:

- run program
- enter ‘2’ or ‘3’

Input Specification: user must input a valid username to add as a friend

Output Specifications: “Friend request sent” or “Friend added”

Pass/Fail Criteria: Friend list must be updated in database

3 Test Results

ID#3 - Message Someone

Date(s) of Execution: 4/22/2022

Staff conducting tests: Matt Selvaraj

Expected Results: Message sent Successfully

Actual Results: the message was sent to the user being communicated with

Test Status: Pass

ID#4 - Start a Run

Date(s) of Execution: 4/22/2022

Staff conducting tests: Jibreel Mohamed

Expected Results: 30-second timer to start

Actual Results: 30-second timer started

Test Status: Pass

4 Regression Testing

Not Applicable.

IV Inspection

1 Items to be Inspected

Use-case ID 1: Register

- Creator: Matthew Selvaraj
- Inspector: Jibreel Mohamed

Use-case ID 2: Log in

- Creator: Jibreel Mohamed
- Inspector: Matthew Selvaraj

Use-case ID 3: Message Someone

- Creator: Jibreel Mohamed
- Inspector: Matthew Selvaraj

Use-case ID 4: Start a run

- Creator: Matthew Selvaraj
- Inspector: Jibreel Mohamed

Use-case ID 5: Add friend

- Creator: Matthew Selvaraj
- Inspector: Jibreel Mohamed

Data requirement ID 6: Social data

- Creator: Matthew Selvaraj and Jibreel Mohamed
- Inspector: Jibreel Mohamed

2 Inspection Procedures

Inspectors were expected to inspect code while in contact with the coder. The inspector was to perform the features associated test, and then report success or failure. Debugging was then to proceed as a joint effort between the coder and inspector. This will be done over one or two online meetings conducted via zoom or discord. Meetings are expected to go no longer than 90 minutes.

3 Inspection Results

Use-case ID 1: Register

- Creator: Matthew Selvaraj
- Inspector: Jibreel Mohamed
- Inspector found code to be working

Use-case ID 2: Log in

- Creator: Jibreel Mohamed
- Inspector: Matthew Selvaraj
- Inspector found a bug between the login and home screen.
- Debugging then occurred as a joint effort. Bug was resolved within the client.java file.

Use-case ID 3: Message Someone

- Creator: Jibreel Mohamed
- Inspector: Matthew Selvaraj
- Inspector found a bug in message history.
- Debugging then occurred as a joint effort. Bug was resolved within the message database file.

Use-case ID 4: Start a run

- Creator: Matthew Selvaraj
- Inspector: Jibreel Mohamed
- Inspector found code to be working

Use-case ID 5: Add friend

- Creator: Matthew Selvaraj
- Inspector: Jibreel Mohamed
- Inspector found code to be working

Data requirement ID 6: Social data

- Creator: Matthew Selvaraj and Jibreel Mohamed
- Inspector: Jibreel Mohamed
- Inspector found code to be working

V Recommendations and Conclusions

So far, all items covered have passed the testing phase and inspection process. Moving forward, we will have to implement a UI to go with the code and test for fine-tuning in the future and make sure that the amount of bugs in the code is close to none.

VI Project Issues.

1 Open Issues

(1) How will the google maps API be implemented into the project?
(2) The pandemic had made travel restrictions to reduce the spread of COVID -19.

(3) How will we implement what areas are being restricted to roam people out?
(4) How will group run and schedule event functionalities work?
(5) How can the Users know about their fitness data? How can we display them?

2 Waiting Room

One of the potential features that are not included in the current release is the implementation of Google and Weather API. The Implementation of these API will help the users to navigate in their desired path. The weather API will help the users to know the weather conditions and take necessary precautions. The Google API will help the users to take the path which they are comfortable with.

3 Ideas for Solutions

There are many tools in the real world which help us with cross platform development such as using libraries like flutter. Due to the already existence of fitness apps in the market it is possible to reduce the development costs through these tools.

4 Project Retrospective

One of the difficult parts in developing this project idea is to have a basic idea of what we need to implement inorder to make it functional. In the first release we focused on the Login/signup page for users in order to access the app. In the second release We had a feature to add or send requests for friends. We have worked on how we can send/receive messages between two friends. There are many challenges to overcome to implement the individual run feature. Implementing API and creating a UI are some of the parts which need to be improved in the future. Synchronized team meetings really helped on the way we developed this project. Tools such as Google drive, Git and Zoom were really helpful to collaborate, share and submit the work for this project.

VII Glossary

Average Pace: The pace the user is running at throughout all the completed miles so far.(*Group10Developmentfinal*)

Challenge Run: A route that is generated by the user that can be used to invite other users to

run to achieve goals like highest speed or lowest time.(*Group10Developmentfinal*)

Charity Run: A run that will be pledging a certain amount of money to a designated charity

depending on completion or amount run.(*Group10Developmentfinal*)

Normal Run: A route that is generated by the user that can be used to invite other users to run

at their own pace.(*Group10Developmentfinal*)

VIII References / Bibliography

[1] C. Khoshaba, W. Davidson, C. R. Gomez, and J. Koreth, "Group10Developmentfinal." UIC, Feb-2020.

IX Index