**Run Smart**

V. J. R. Koh, J. Z. Yeo, Q. Z. Lian

1. INTRODUCTION

Our proposed solution caters mainly to the different age group who wishes to exercise and escape the confines of self-isolation at home after the lockdown period due to Coronavirus, Covid-19. Leading to a rise in physical activity behavior and people desire to head outdoor and exercise. The most common form of exercise which most people choose to happen to be running given the accessibility of the sport to run in any environment. People no longer need to travel specifically to the stadium or the park due to the accessibility of tracks surrounding our neighborhood in Singapore due to our government effort. However, there is a major flaw in running outdoors given the ever-changing weather conditions. Given Singapore weather conditions, it is difficult for individuals to predict the weather especially during the monsoon season. Hence, our application aims to resolve the troubles of individual who may face such issues and improve their running experience by offering them a smart running app which will warn user ahead of time if it is about to rain if they wish to run in this short duration of time. In addition, there is a built-in feature of a timer that counts down from the duration which the user set initially. We decided to narrow the scope of our users to Changi for a start mainly to cater to runners in our school where there aren’t much sheltered jogging paths nearby. With this application, we hope to improve the user experience of runners in Changi by providing them with a wonderful running experience.

1. APPLICATION FEATURES

Our Smart running application is simplistic which makes it easy for users of all ages to navigate while managing the application. The application comprises of a world time at the top to allow user to keep track of the time easily. To begin, the user will first input in the underlying text input beneath the clock in minutes before pressing the submit button. Once the user presses the submit option, the system will cross-check the final time with the data extracted from the weather API according to the duration preset by the user to verify if the weather is clear or rainy during that duration. A prompt message will pop out to suggest to the user whether the weather is suitable to run or not. Depending on the weather, the application background will also change accordingly to the weather indicated from the weather API. E.g., If showering rain is detected from the time which the user intends to run between, the background will change into a showering rain graphic. Upon closing the pop-up message, the display will reset back to its original background. Apart from this, there is a timer at the bottom that allows the user to keep track of the duration of the run.

The user is also able to set the run time duration in terms of hours, minutes and seconds using the “Set Duration” button at the bottom of the countdown timer. The user would not be able allowed to exceed 1 hour and 30 minutes for safety reasons such as changes in the forecast. Setting the duration through the countdown timer would result in a change in value to the input box under the clock, which the user could instantly use to check the forecast using the submit button.

If the user first inputs a number in minutes in the textbox, the user must press the Set Duration button before pressing play to start the countdown.

If the user presses Set Duration immediately, they would be able to type out the timing that they wish for the hour's component, minutes component and seconds component. After that, they would need to press the Set Duration again before pressing the play button to start the countdown.

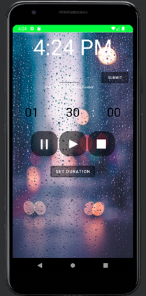
The user may only reset using the stop button after pressing the play button, be it when they have paused it or when they have not started the countdown.

The photos below show screenshots of our application running in the virtual android emulator:

A picture containing text, monitor, electronics, screen

Description automatically generated A screenshot of a cell phone

Description automatically generated with medium confidenceA picture containing text, electronics, monitor, display

Description automatically generated

1. FRAMEWORKS & LIBRARY USED

For world time, TextClock widget is used to display the current time as a formatted string in a 24-hour format system setting. As for the API, retrofit and Gson libraries were used. Retrofit library is used as it abstracts HTTP calls and the serialization process, thus making it easier to use. Gson is chosen as the deserializer to convert the JSON response object into Java objects.

.4. ANDROID TESTING UNIT

For android Testing unit, our group tested the time conversion function to verify whether the system can convert the user input successfully from minutes to hours and minutes which is represented in a string format.

5. MOST INTERESTING/CHALLENGING PART OF THE APPLICATION

The most challenging part of the application was the logic behind processing the data returned from the API (e.g., the constraints and which data to select).The API that we chose to use is a 2 hour weather forecast from <https://data.gov.sg/dataset/weather-forecast> using date parameter to retrieve all of the forecasts issued for that day and is updated half hourly from NEA. While a JSON schema was given on their website, the results obtained were different from our expectations. This is because we expected only 1 set of data for each specific 2-hour time period. However, there is more than 1 set for each period which could be due to updates from NEA thus we decided to use data of the latest data set available when user presses the button.

Initially, we wanted to allow users to set their run duration to any duration length that they want and have the app present the weather at each specific segment of their run. However, due to limitations of the dataset this was not possible since the weather API can only forecast up to 2 hours into the future from a set start time ( every 30 minutes from 12 am).For instance, the valid period of weather could be from 12 am to 2 am, 12.30 am to 2.30 am , 1 am to 3.30 am etc. Thus, there could be a possibility where user queries data at 12.20 am and the weather data presented is only applicable for the next 1h 40 minutes until 2 am thus if they were allowed to select their own duration for e.g., 2 hours (to check weather from 12.20 am to 2.20 am) then data may not be applicable since the next set of data will also only be uploaded at 12.30 am. Hence, we set the maximum duration they can select to be 1h 30 mins since new set of data for the next period is uploaded every 30 mins.

Additionally, there is no list of possible string values for the forecast field on the website , thus our code needs to handle all classifications of string values that suggest rain. Thus, we did a search for the substring ‘showers’ in the string value we obtain and if there is we return true else we return false for any other possible strings to suggest that it is not rainy .There is also a need to handle the case where the API is unable to fetch data either because the query parameter is wrong or due to network issues. For this, we decided to return error.

1. PROJECT MANAGEMENT: HOW TASKS ARE DEFINED & DISTRIBUTED AMONG GROUP MEMBERS

In the beginning, our group gathered to discuss the scope of our project to determine the direction in which we aim to target. After finalizing the general overview of the project, the task was split evenly among the team members into smaller portions in the following structure. Different people were tasked to complete certain task, we had members handling the API interface, configuring of the timer system and lastly the user experience which allows to users to communicate with the system to get information of the weather during that time and having the graphic display change according to the weather condition retrieved from the API.

1. CONCLUSION

In conclusion, our team managed to develop a smart running application that predicts the weather conditions from the duration of time that the user presents. Together with a built-in feature of a timer that allows the user to run while keeping track of the time when using the app. This application is also user-friendly due to the simplistic design of the application, making it easier for users of all ages to operate the module. Moving forward, we wish to cater to a larger group of audiences, beyond Changi to the rest of Singapore or other parts of the world. In addition, we hope that this device will be able to allow user to select duration more than 2 hours and with more added features to the application, we aim to improve the overall user experience of everyone using the application.

1. ACKNOWLEDGEMENT

The Team thanks TikTok presenters for their patience and guidance, sacrificing out-of-office hours including during weekends in organizing the TikTok camp. Their experience and constructive criticisms have guided the Team’s direction.

1. REFERENCE

**Link to RunSmart GitHub:** https://gitfront.io/r/user-6240300/P42qhBMBM1ZF/RunSmart/