WPI CS4518-A22 Mobile and Ubiquitous Computing

Sweat Life

Professor Guo

Jacob Chlebowski - jachlebowski@wpi.edu
Zaq Humphrey - zihumphrey@wpi.edu

Table of Contents

List of Figures

I. Introduction

Students often have trouble scheduling when or where they desire to pursue physical activity. One might think to use a notepad on their phone, or set a daily reminder, which can quickly clutter their screen and get messy. In order to solve this problem, we have decided to make an app that would take your workout regime and format it in an orderly manner. It would be able to schedule a workout on your in-app calendar and save you the trouble of inputting your routine on other applications. When looking for workout schedulers on your respective app store, you may come across other applications that require payment for additional perks. Our application, which will be 100% free of charge, is guaranteed to provide effortless scheduling so you can work around your other plans whether it's your education or your home-life.

In this report, you'll discover the architecture of our system and why we made it easy to navigate this application, *Sweat Life*. The home page (MainActivity) is the central point for users to access everything they need, without having to switch back and forth between other screens. The navigation bar on the bottom of the screen allows for seamless transitioning no matter where you are or what you're doing.

II. System Architecture

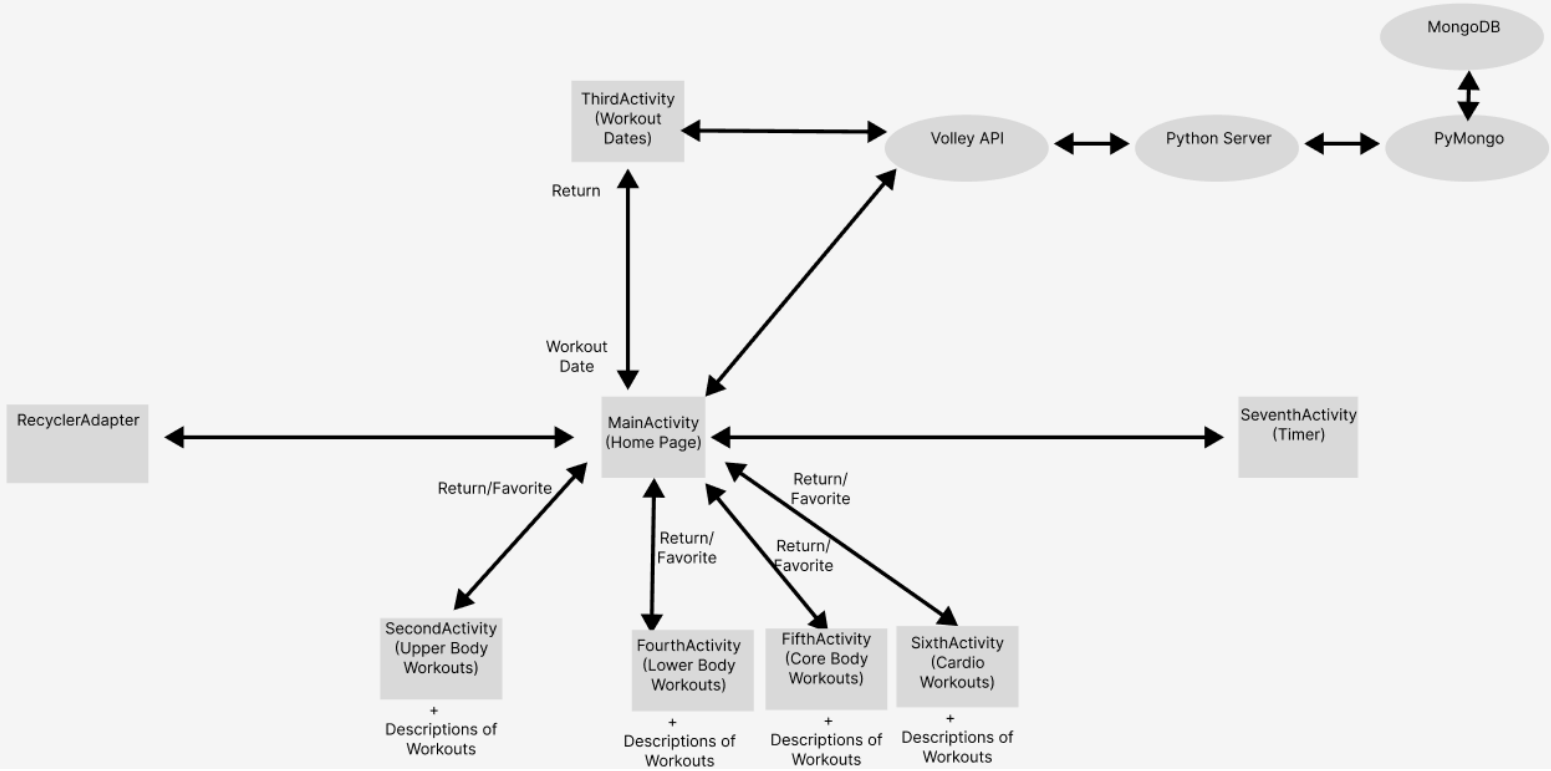

Figure 1: System Architecture Diagram

III. Formal User Stories

As a student whose academic schedule is packed, I want to implement my workouts into my outlook calendar. That way, I can stay on top of academics, activities, and my personal workouts. If *Sweat Life* didn't include the Outlook implementation, it would be difficult to switch between calendars. *Sweat Life* allows me to safely schedule around what is important to me in my student life.

As someone who is new to exercising and is unsure of the variety of existing workouts, I want a catalog of exercises with explanations of the workouts and what body part they develop so that I can properly create a workout tailored to my needs. Normally, looking up the workouts I am using would suffice and tell me the information I need. Having all the details easily accessible in one location saves a lot of time and caters to my demands.

As a student who wants to work out on their own schedule, keeping track of how long my workouts are is a necessity. Having a built-in timer into *Sweat Life* would allow me to effortlessly record the duration of my workouts. I can also keep track of my progress in a timed session or specific exercises at different time intervals. During a cardio workout for instance, I prefer to run in time intervals. If I want to, I can run for 1 minute, then slow down for another, then run an extra interval for my desired time.

IV. Functional Requirements

a) UI Mockups

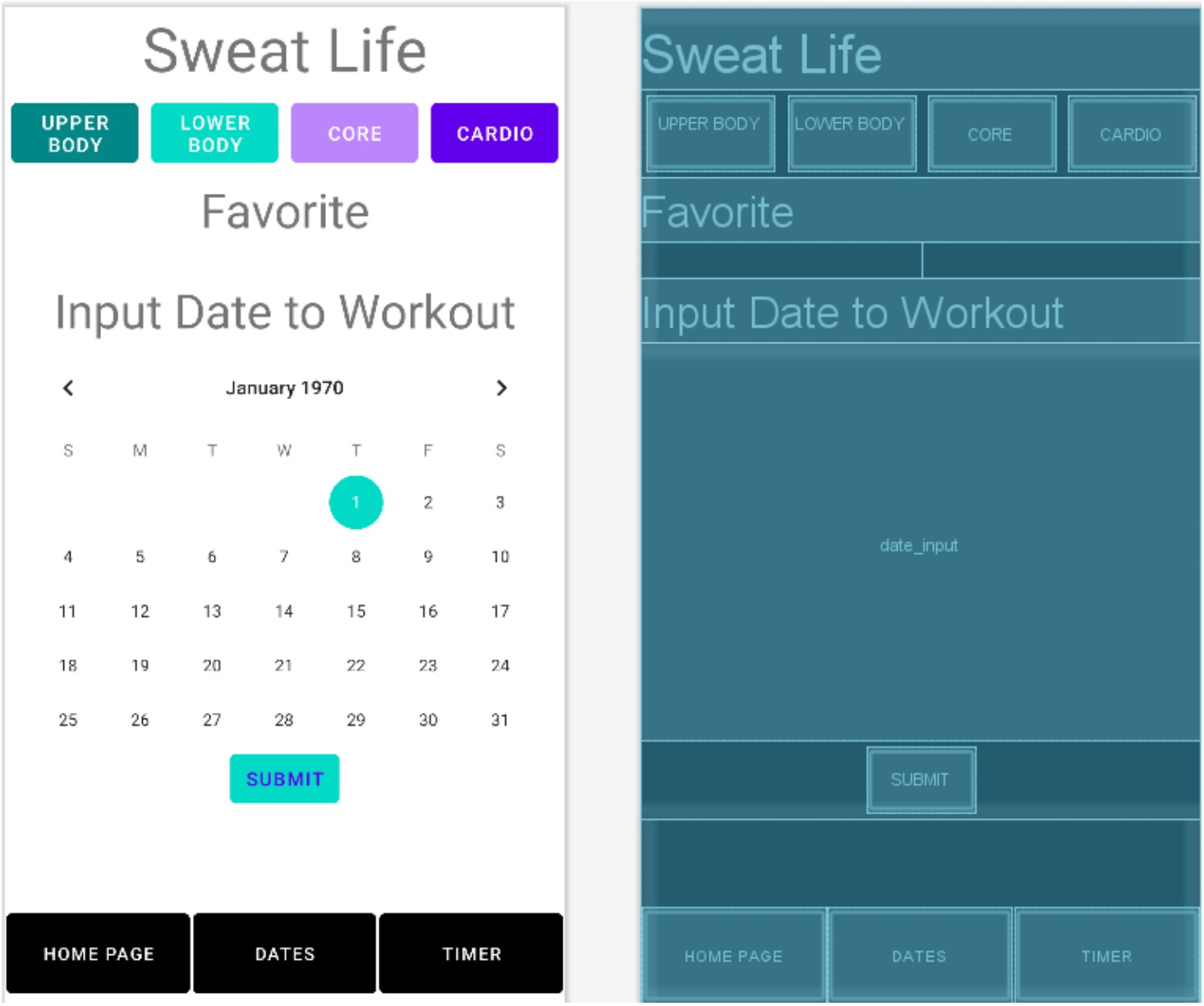

Figure 2: MainActivity ("Home Page" UI Mockup) - Easy Navigation

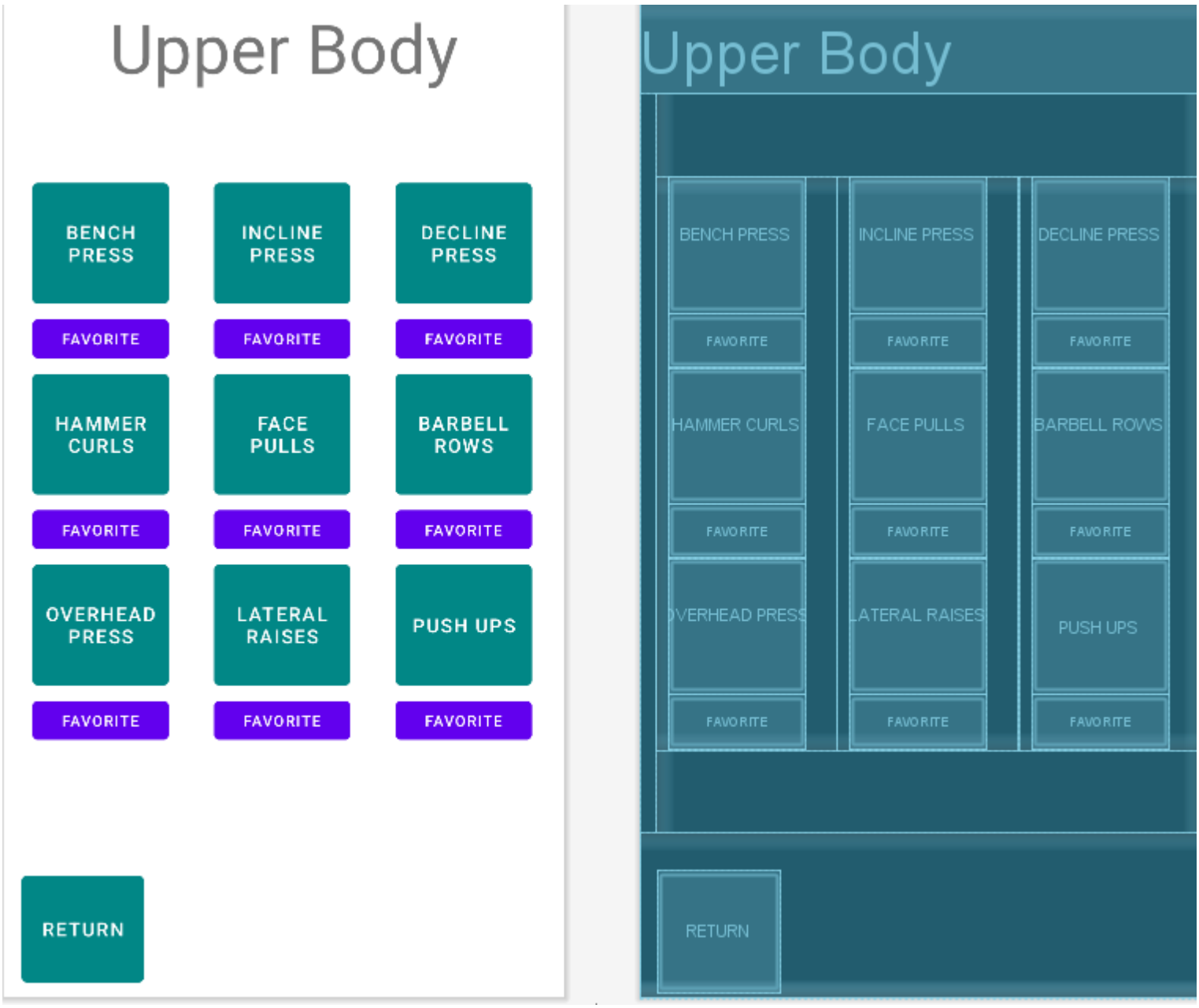

Figure 3: SecondActivity (UI Mockup)

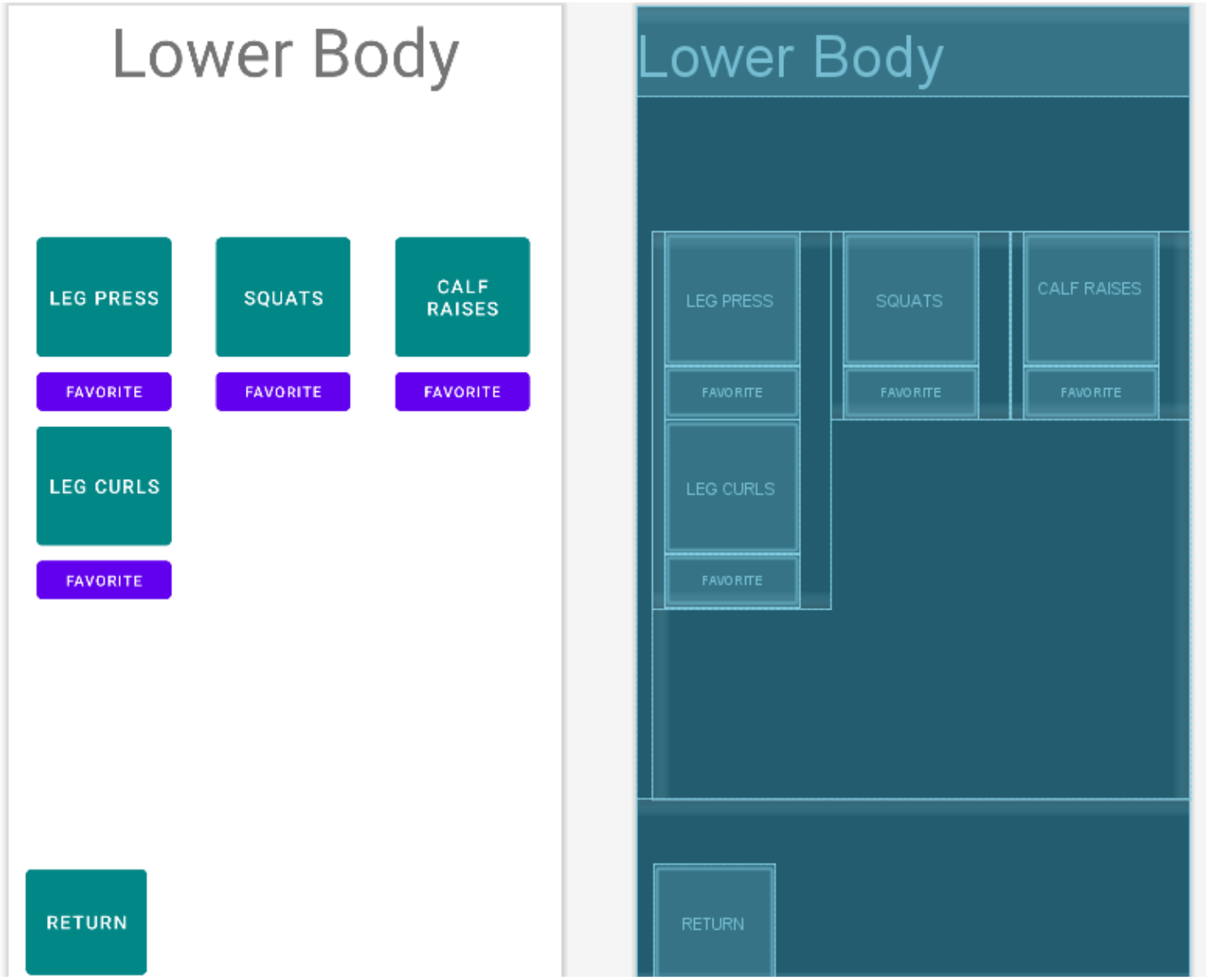

Figure 4: LowerActivity (UI Mockup)

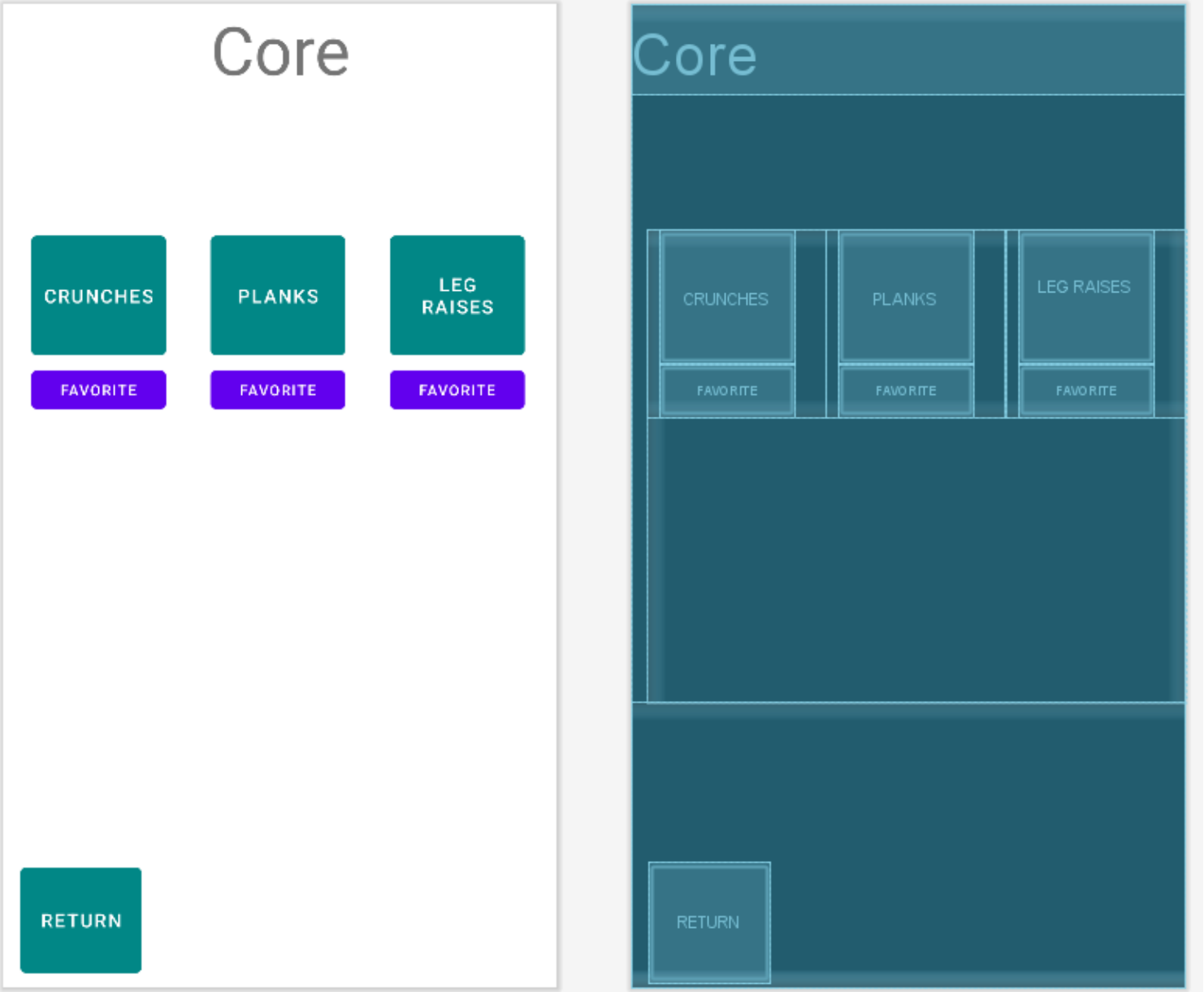

Figure 5: CoreActivity (UI Mockup)

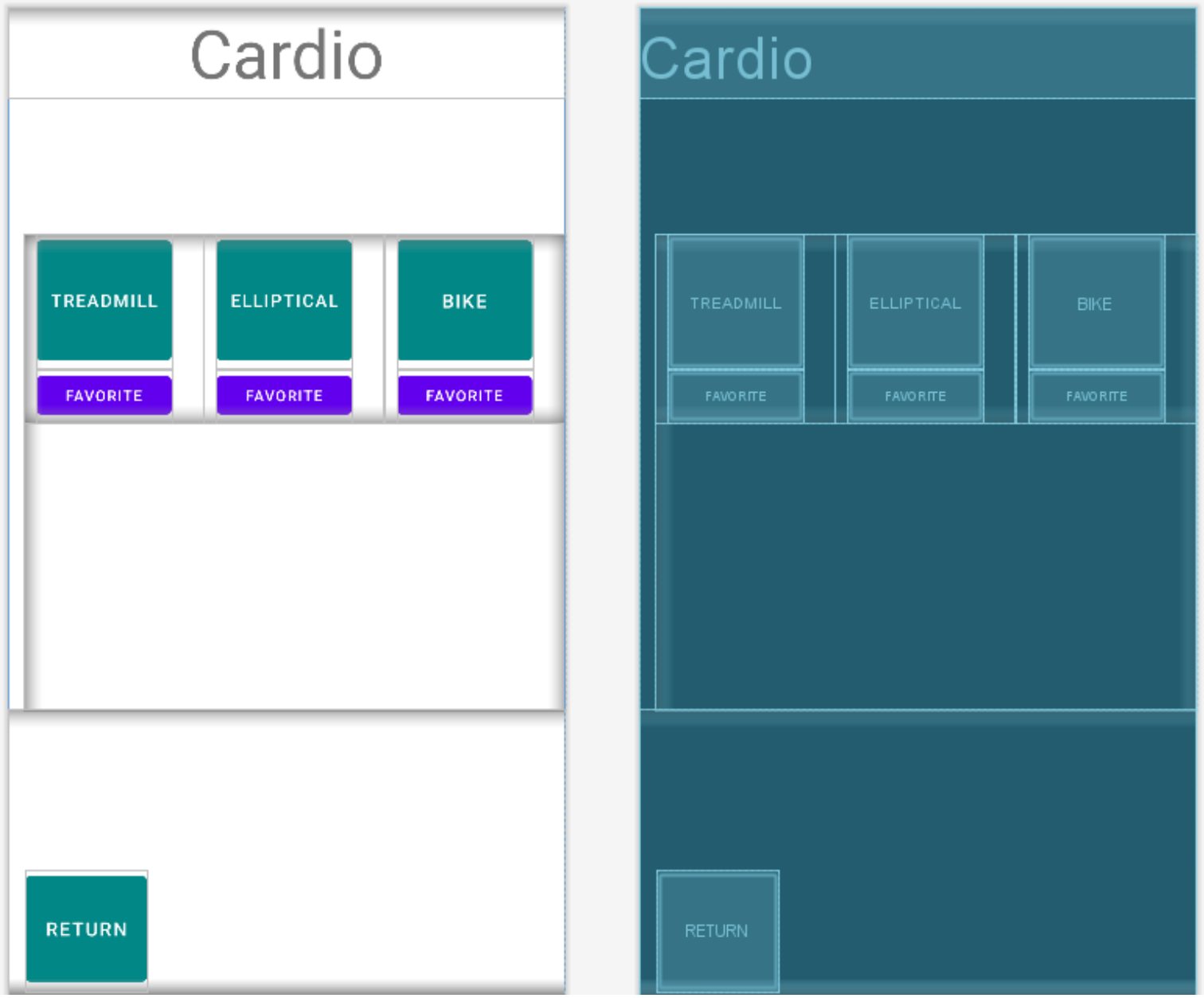

Figure 6: CardioActivity (UI Mockup)

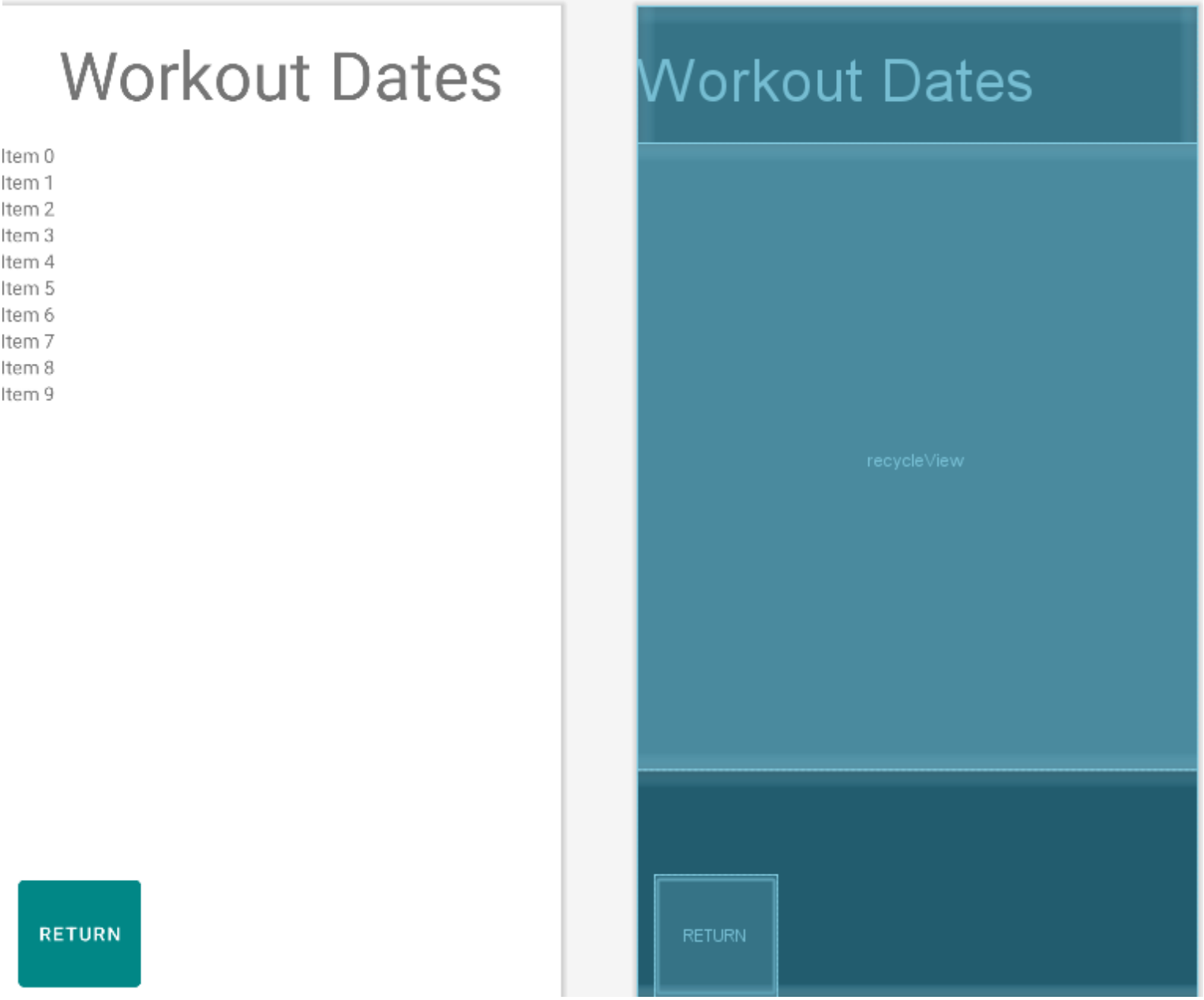

Figure 7: ThirdActivity (UI Mockup)

b) Use Case Diagram

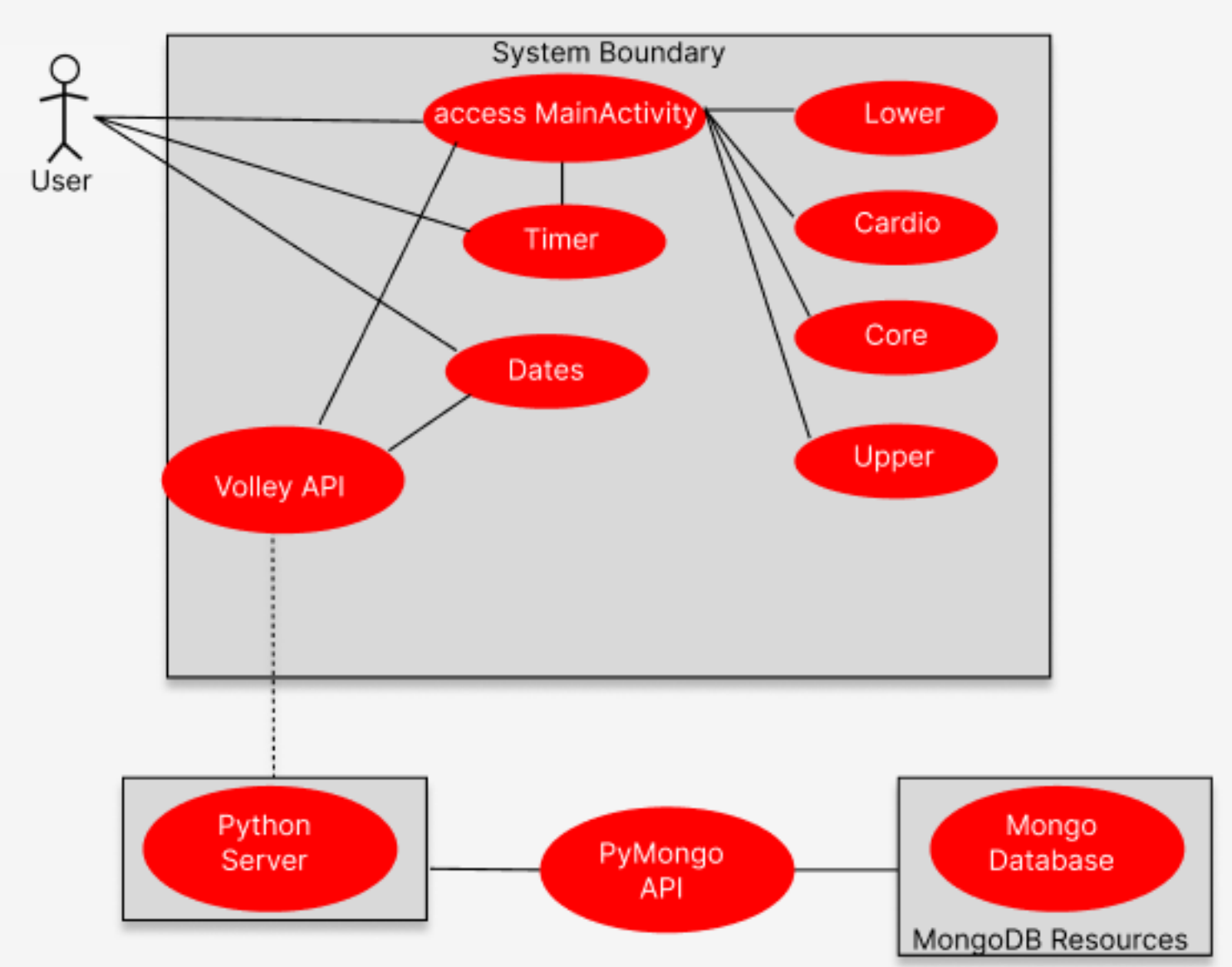

Figure 8: Use Case Diagram

V. Non-functional Requirements

a. Performance Considerations

A particular scenario where there might be a performance issue would be if someone was to create an excessive amount of workout dates on the app. This would cause the program to draw a large number of resources to be able to display the dates as well as use a lot of memory. A solution to this problem is to restrict how many dates the user can input. The user would be limited to a maximum of 10 workout dates and would be able to add or remove as many dates as they want as long as it's inside of that limitation. Another scenario in which a performance issue arises is when there are many workouts listed in the children activities and the app would take a long time to be able to open the children activities. To solve this problem, we will limit the number of workouts in each of the children activities to one page. In the future, more workouts could be added and unloading one page while loading the next page would be a good solution, but for the duration of this class the workouts will be limited.

b. Security and privacy considerations

A user might be worried that *Sweat Life* will be stealing data from them behind the scenes. This will not be the case, as our application will not request or steal any information. The application can be used completely anonymously. Even when inputting the workout dates into the outlook calendar activity, the API will be using the Volley and MongoDB database, meaning that absolutely no information will be needed from the user. In regard to information, however, there could be a scenario where someone wants to log in to make their account more secure. Our app does not currently

have authentication, but that is definitely something to consider in the aspects of both security and privacy. In terms of our database implementation, MongoDB safely stores all data from the user in a database that no one can access.

VI. References

Buttons : android developers. Android Developers. (n.d.). Retrieved September 14, 2022, from https://developer.android.com/develop/ui/views/components/button

Create dynamic lists with RecyclerView : android developers. Android Developers. (n.d.). Retrieved September 14, 2022, from https://developer.android.com/develop/ui/views/layout/recyclerview

Layouts : android developers. Android Developers. (n.d.). Retrieved September 14, 2022, from https://developer.android.com/develop/ui/views/layout/declaring-layout

Linear Layout : Android developers. Android Developers. (n.d.). Retrieved September 14, 2022, from https://developer.android.com/develop/ui/views/layout/linear

Microsoft. (n.d.). *Free personal email and calendar from Microsoft*. Outlook. Retrieved September 14, 2022, from https://outlook.live.com/

Relative layout : Android developers. Android Developers. (n.d.). Retrieved September 14, 2022, from https://developer.android.com/develop/ui/views/layout/relative

Scrollview : android developers. Android Developers. (n.d.). Retrieved September 14, 2022, from https://developer.android.com/reference/android/widget/ScrollView