6x6 Fitness App

Documentation

Programming for Mobile Devices

By Darco Murja

Informatics in English Year 2, group 1

Phd. Lecturer Liviu Octavian Mafteiu-Scai

15.05.2021

Universitatea de Vest din Timişoara

Table of Contents

1. Abstract

During the second semester of the second year of University, I developed *6x6 Fitness App*, an Android native project for *Programming for Mobile Devices*. In this document I will present the goal of the application, the innovative features and programming related insight into the source code.

2. Introduction

6x6 Fitness App is a mobile application available for Android devices, designed to help users monitor their training and grow their fitness level in time by letting them plan ahead.
The application was built around the idea of scaling, being upgradable in the future in order to sustain a large number of users.

The application's logo, designed by myself

3. The Application's Goal

6x6 Fitness App is actively encouraging users to start a workout routine, especially in today's world pandemic crisis, when most of us lack any physical activity. While it was designed for common people, it does have an emphasis on more advanced athletes, as it features some more intense exercises. The application focuses on tracking users' past workouts and plan the future routines.

.

4. The Users

The main focus group is the common person who works at the desk or from home and requires some sort of light or more intense workout in order to get in shape or continue one's progression.

The app also hopes to aid the gym instructors to prepare workouts for their clients and ease their tracking of past workouts.

Lastly, *6x6 Fitness App* can be used by advanced athletes who require a strength focused workout in order to maximize muscle growth.

5. State of Art

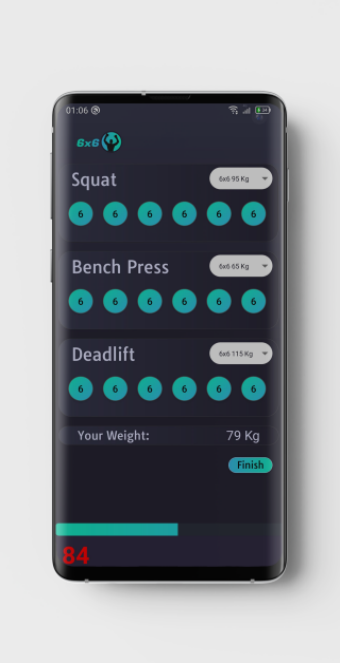
The Fitness field has seen an incredible increase in user float and revenue in recent years. During the first months of the Coronavirus pandemic, when most gyms had to close down, people all over the world started to workout at home. Fitness apps have acquired over 1 billion downloads in 2020 alone. Here are some of the most popular workout companion apps.

1.1. Stronglifts 5x5

Perhaps the most popular app when it comes to strength training and lifting very heavy weights, 5x5 is an app that combines the classic and proven workout routine used by professionals, with a new and modern UI.
The app is free, but some features like tutorials require an additional fee of 5$.
It was installed over 1 000 000 times on Android devices.

6. Original Features

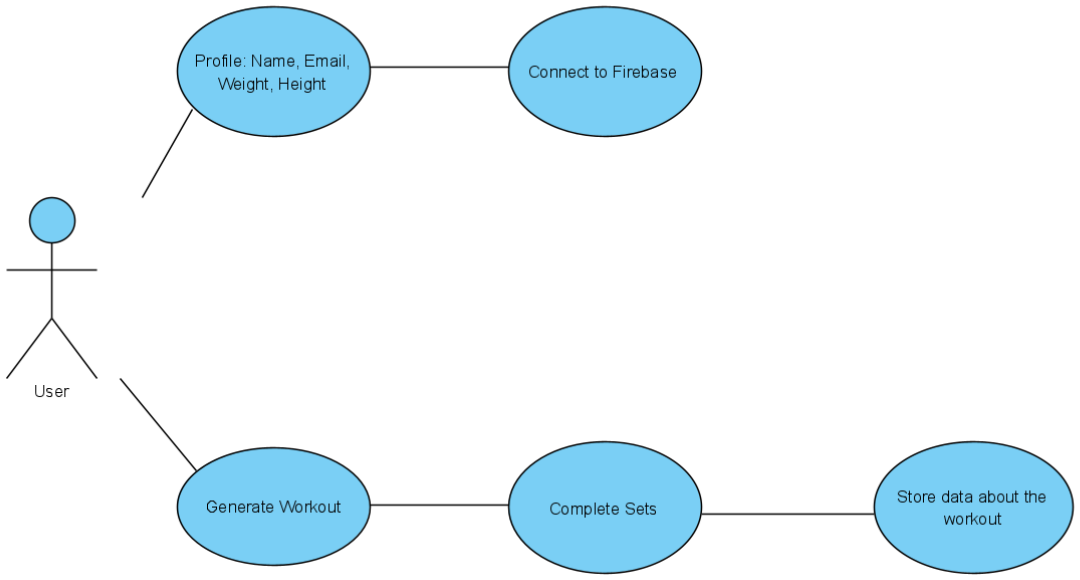
While the app resembles a lot of the main features of *Stronglifts*, it comes with new ways of user interaction, but most importantly, it resolves many of the bugs from the original app that makes is almost unusable.



(The main workout screen)

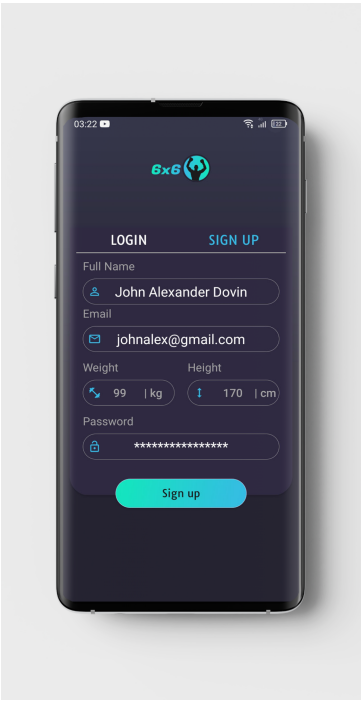
Every set is represented by a circle that has written how many repetitions the user has made. Once you activate a set, if you have not finished all 6 required repetitions, you can click again and the counter will subtract one. Once you complete a set, the timer will reset at 300 seconds.

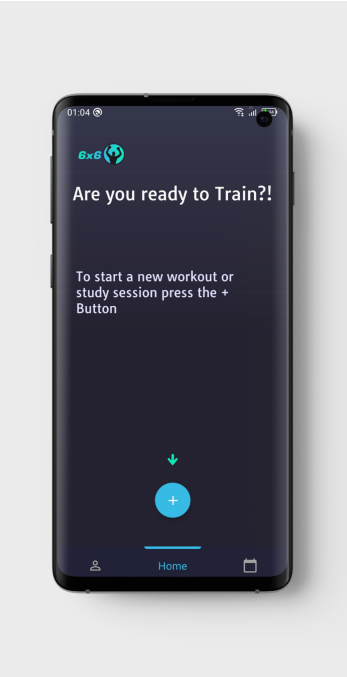
Functionality



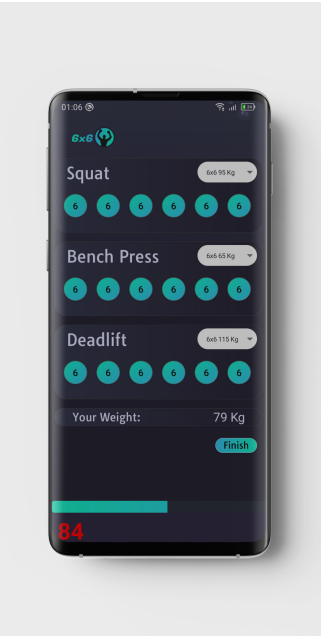
7. User Interface and Interaction

Navigation between screens is done using a custom navigation bar. When you first open the app, you are welcomed by the LogIn Screen where you can also choose to sign up.





(Home Screen -start the workout)



The Workout Activity is where you document your workout, see the exercises you have to follow and find out how much time you should rest between sets.

8. Technical Documentation

6x6 Fitness App is developed in Kotlin, using Android Studio .
The following screenshots are of from the Workout Activity:


```
private fun startSets(countDownText:TextView) {
    val squatArray = arrayOf(set1,set2,set3,set4,set5,set6)
    val benchArray = arrayOf(bench1,bench2,bench3,bench4,bench5,bench6)
    val deadliftArray = arrayOf(dead1,dead2,dead3,dead4,dead5,dead6)

    for ( i in squatArray){
        i.setOnClickListener { it: View!
            setClick(i,countDownText)
        }
    }

    for ( i in benchArray){
        i.setOnClickListener { it: View!
            setClick(i,countDownText)
        }
    }

    for ( i in deadliftArray){
        i.setOnClickListener { it: View!
            setClick(i,countDownText)
        }
```

For every button in each array, we call the function setClick, that takes a button and its TextView.

```
    }

private fun setClick(set: Button,countDownText:TextView){
    set.setTextColor(rgb( red: 255, green: 255, blue: 255))
    var i = 0
    if(set.text.toString() == "0"){
        i = 7
    }
    else{
        i = set.text.toString().toInt()
    }
    activateSet(set, i: i-1, countDownText)
}
```

The setClick function sets the color of the text to white and adds the right number to the Button's text. It also calls the activateSet() function.

```
@SuppressLint( ...value: "SetTextI18n")
private fun activateSet(set:Button, i : Int, countDownText:TextView){
    set.setBackgroundResource(R.drawable.round_button)
    if(i == 0) {set.text = "0"
        set.setBackgroundResource(R.drawable.round_inactive_button)
        set.setTextColor(rgb( red: 255, green: 0, blue: 0))
    }
    else         set.text = i.toString()
    if(i == 6)   set.setTextColor(rgb( red: 0, green: 0, blue: 0))

    timer.cancel()
    startCounter( mCounter: 180,countDownText)
}

private fun setUpSpinners(){
    val list:MutableList<String> = ArrayList()
    for (i:Int in 20..170 step 5){
        list.add("6x6 $i Kg")
    }

    val adapter = ArrayAdapter( context: this, R.layout.spinner_text, list)
```

The activateSet() function first sets the Background of the button to the drawable, the blue and magenta gradient. Depending on the number it receives as a parameter, it sets the color of the text to white, green, grey or red. Once you activate a button, the counter resets with the function StartCounter(), but not before we make sure the old one is canceled not to crash the activity. The setUpSpinners() gets the spinner ready, using a custom adapter to create the right list of texts when you open the menu.

```
private fun startCounter(mCounter:Int, countDownText:TextView){
    var counter = mCounter
    timer= object : CountDownTimer((counter*1000).toLong(),  countDownInterval: 1000) {
        override fun onTick(millisUntilFinished: Long) {
            countDownText.text = counter.toString()
            progressBar.progress =(100- (counter * 5)/9)
            counter--
            if((counter == 90) or (counter == 25)){
                countDownText.setTextColor(Color.parseColor( colorString: "#FF0000"))
                vibrate()
            }
        }
        override fun onFinish() {
            counter = 180
            countDownText.setTextColor(Color.parseColor( colorString: "#ffffff"))
            vibrate()
        }
    }.start()
}
```

The startCounter() function takes a number, an int which by default is 300, and a TextView that represents the timer itself. Using the timer object we take the int value and transform to milliseconds, and on each tick we update the text to the respective number. When it finishes the timer, or it reaches 90 or 25 seconds respectively, the phone will vibrate.

```
private fun  vibrate(){
    val v = (getSystemService(Context.VIBRATOR_SERVICE) as Vibrator)

    if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.O) {
        v.vibrate(VibrationEffect.createOneShot( milliseconds: 1000,
            VibrationEffect.DEFAULT_AMPLITUDE))
    }
    else {
        v.vibrate( milliseconds: 1000)
    }
}
```

Vibrate() is a function that shakes the phone as if it received a push notification. As long as the build version is higher than the version number of the codes, it can use the default Android function VibrationEffect.DEFAULT_AMPLITUDE.

9. Conclusion and the future of 6x6 Fitness App

6x6 Fitness App is an application that can fit the gap between the beginners' and the pro athletes' markets, sitting in the middle where it can be used by millions.
The next step is to implement features for gym instructors, to help them track multiple clients' workout routines. This way it can be used by gyms and have a license based selling point, where gyms are given access to its full potential by paying a yearly subscription.

10. References

https://developer.android.com/docs/
https://www.youtube.com/channel/UCnKhcV7frITmrYbIU5MrMZw
https://www.youtube.com/watch?v=F9UC9DY-vIU
https://www.youtube.com/watch?v=L8Xq15NTuCc