Training Project Laboratory Report

Jena Woodroffe
Neptun: FNA1A3

Physiotherapy App

Description of the project:

This Android-based application aims to give physiotherapists a single platform from which to run their practice from. The final project will comprise of two applications - one exclusively for doctors and one for their clients, however for this semester's work, I will focus on the doctors' application. The app will contain the following features:

- ability to track appointments, create new appointments and notify the respective clients, display the upcoming appointments to the doctor
- tracking all information related to clients
    - their medical notes made by the doctor during appointments
    - their personal logs that they can make and update any time between appointments
    - text messages - the portal will have a chat portal where clients and physios can talk and sending photos and videos
    - their current exercise regime, the rehabilitation exercises they have been prescribed for the week along with their description, sets and reps

Tasks required for the week:

- o   finalise the use of Firebase authentication used for logging in
- o   successfully register new uses and add them to the Firebase store

Tasks done this week:

- ✔   finish the xml for the log in and sign up activities
- ✔   code the logic to change between the sign up screens
- ✔   create a singleton object used to store user's data when registering to the app
- ✔   validate user data during registration
- ✔   main home custom navigation bar, successfully switching between activities

Budapest University of Technology and Economics
Faculty of Electrical Engineering and Informatics
Department of Automation and Applied Informatics

H-1117 Budapest, Magyar Tudósok krt 2. Q.B.207
Tel: +3614632870, Fax: +3614632871
www.aut.bme.hu

Code Screenshots:

Loading page when app opens

My Practice

Email address

Password

LOGIN

Register your practice

Sign Up
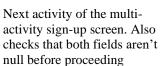
Name

Email address

Telephone number

NEXT

Sign up page – displayed when clicking *Register your practice*

- Has working back button that returns to the Login activity
- Clicking Next will verify if all of the data given is valid – checks if the Name is null or not, checks if the email address and number are correctly formatted

---

Provide IDs

Certification ID

Practice ID

NEXT

Next activity of the multi-activity sign-up screen. Also checks that both fields aren't null before proceeding

---

Create a password

Password

Re-enter Password

FINISH

Final activity of the sign up-activities. Finish button checks if the initial password meets all the requirements and if the two passwords match.

All of the Next and Finish buttons give the data they receive to the singleton DataRegistration class shown below

---

Custom navigation bar with the unified color of the app. It took a while to get the shape and icons correct, but most of the time was spent trying to override and remove default android styling that is applied to different components and activity changes

Budapest University of Technology and Economics
Faculty of Electrical Engineering and Informatics
Department of Automation and Applied Informatics

H-1117 Budapest, Magyar Tudósok krt 2. Q.B.207
Tel: +3614632870, Fax: +3614632871
www.aut.bme.hu

DataRegistration Singleton Object

The code below assigns the checking of valid data to this object. This ensures that if I ever need to register a doctor somewhere else in the project, the validation code can still be used. Each field checks the requirements for the argument given and throws errors if the value isn't ok. The messages the errors return are used in the edit text pop up errors that are shown in the results section

```kotlin
package com.example.mypractice

object DataRegistration {
    /*setters for the fields
    each setter checks the values to make sure they are valid
    returns an error with the appropriate error message if not
    */
    var name: String = ""
        set(value)
        {
            var temp = name.trim()
            if (value == "")
            {
                throw IllegalArgumentException("Name cannot be null")
            }
            else{
                field = value
            }
        }

    var email: String = ""
        set(value) {
            val emailRegex = Regex("^\\w+([.-]?\\w+)*@\\w+([.-
]?\\w+)*(\\.\\w{2,4})+$")
            if (value.matches(emailRegex))
            {
                field = value
            }
            else
            {
                throw IllegalArgumentException("Invalid Email")
            }
        }
    var phone: String = ""
        set(value) {
            var temp: String = value.replace("\\s+".toRegex(), "")//removing all
the spaces in the string
            var length = temp.length
            var alldig = temp.all{it.isDigit()}
            if (alldig&& length == 10)
            {
                field = temp
            }
            else
            {
                throw IllegalArgumentException("Invalid Number")
            }

        }
    var certID: String = ""
        set(value) {
            if(value.trim().equals(""))
            {
```

Budapest University of Technology and Economics
Faculty of Electrical Engineering and Informatics
Department of Automation and Applied Informatics

H-1117 Budapest, Magyar Tudósok krt 2. Q.B.207
Tel: +3614632870, Fax: +3614632871
www.aut.bme.hu

```kotlin
                throw IllegalArgumentException("ID must be given")
            }
            else{
                field = value
            }
        }
    var pracID: String = ""
        set(value) {
            if(value.trim().equals(""))
            {
                throw IllegalArgumentException("ID must be given")
            }
            else{
                field = value
            }
        }

    var password: String = ""
        set(value) {
            val specialCharRegex = Regex("[!@#\$%^&*()_+\\-
=\\[\\]{};':\"\\\\|,.<>\\/?]")
            val digitRegex = Regex("\\d")
            val pTemp = value.trim()

            //checking if the password matches the requirements for a safe
password
            var valid: Boolean = true
            if( pTemp.length < 8 )
            {
                throw IllegalArgumentException("Password too short. Must be 8
characters or more.")
                valid = false
            }
            if (!specialCharRegex.containsMatchIn(pTemp))
            {
                throw IllegalArgumentException("Password should contain a
special character")
                valid = false
            }
            if(!digitRegex.containsMatchIn(pTemp)) {
                throw IllegalArgumentException("Password should contain a
digit")
                valid = false
            }
            if (valid)
            {
                field = pTemp
            }
        }

    //adding the complete data to the database

}
```

Budapest University of Technology and Economics
Faculty of Electrical Engineering and Informatics
Department of Automation and Applied Informatics

H-1117 Budapest, Magyar Tudósok krt 2. Q.B.207
Tel: +3614632870, Fax: +3614632871
www.aut.bme.hu

Results Screenshots:

- for all of the sign up screens, errors are shown with their respective error messages in each field after the next button is clicked

Tasks to be done next week:
(Tasks you are planning to do in the next week)

- create all the necessary firebase databases needed for the project
- Work on adding clients – displaying them in lists and selecting them
- Creating the basic data structure for the project

Notes (optional) :
I apologise for the delay in uploading my progress to Github. I have been working on this regularly for the last few weeks, but struggled many times in understanding the basics of github and how to use it.

Budapest University of Technology and Economics
Faculty of Electrical Engineering and Informatics
Department of Automation and Applied Informatics

H-1117 Budapest, Magyar Tudósok krt 2. Q.B.207
Tel: +3614632870, Fax: +3614632871
www.aut.bme.hu

Budapest University of Technology and Economics
Faculty of Electrical Engineering and Informatics
Department of Automation and Applied Informatics

H-1117 Budapest, Magyar Tudósok krt 2. Q.B.207
Tel: +3614632870, Fax: +3614632871
www.aut.bme.hu