Computer Project I

Fall 2017

Two-Factor Authentication

Project Report

Group #19

Hüseyin Tosun 150120121
Berkan Dinar 150120102
M. Habib Beyatlı 150120048

14.10.2017

1. Introduction

In the context of this project, we created a system that uses two-factor authentication mechanism via a username, a password and a biometrical authentication element. Face recognition is chosen for this purpose.

Needed face recognition algorithm is downloaded from a GitHub repository[1], implemented and edited for making it suitable for the mechanism. The project was expected to last three weeks and made by three attendants.

In the project, there are 8 classes that used for operations related to the project. These classes are MainActivity, SignUpActivity, AddPersonPreviewActivity, TrainingActivity, SignInActivity, RecognitionActivity, WelcomeActivity and HashFunc. 2 other classes are used as well. However, These classes did not require any changes.

2. Classes

MainActivity

The classes that used for the main operations of the project. All calls that send from buttons are handled here. The activity connects to other activities for operations such as sign in and sign up.

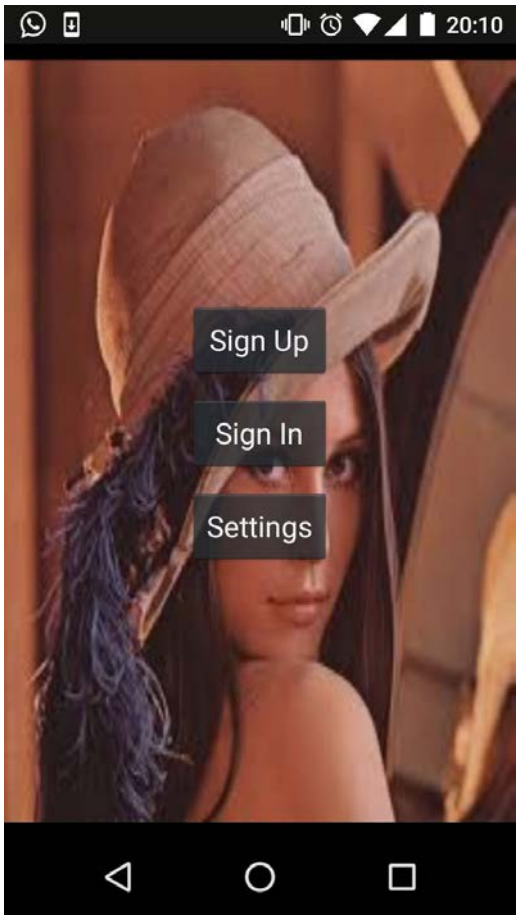

Figure 1: Main Page

SignUpActivity

This is the class that handles sign up operations. In this class, the entered username and password are taken from the user interface and sent to AddPersonPreviewActivity. The controls for duplicated username and lengths of username and password are done here as well.

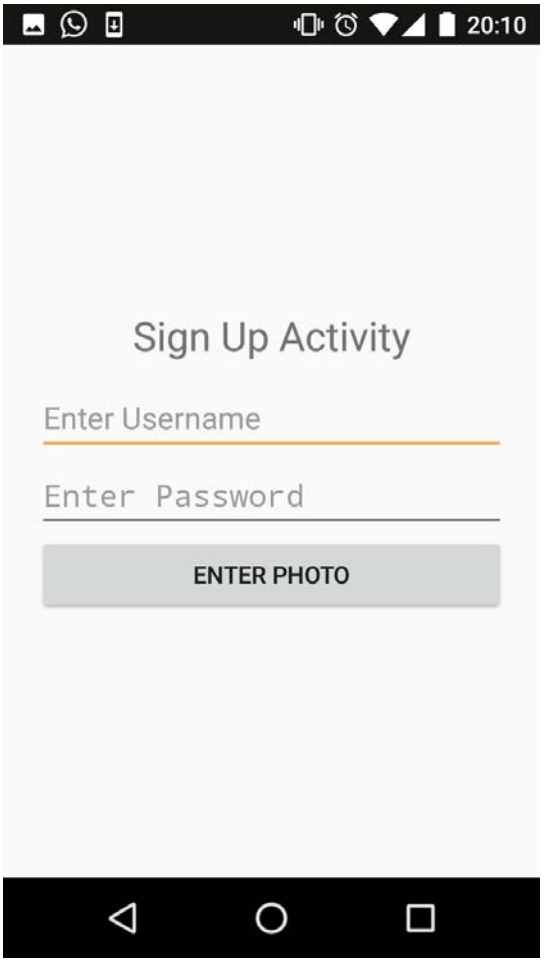

Figure 2: Sign Up Page

AddPersonPreviewActivity

This is the class which deals with the face detection and user registration operations. In this class, the face of the user is detected and stored with username and password. Username and password are recorded as the name of face picture files. For example, if a user's username is "tahir" and hashed password is "456203", the name of the file would be "tahir½456203½photo_number.png".

TrainingActivity

This is the class where the training operations for learning the entered faces takes place. It checks all recorded face files and learns their features. After the training operations, it directs to the main page of the application.

SignInActivity

This class does the operations for the authentication which is the main goal of the project. It takes the username and password. After the “Enter Photo” button clicked, it directs to RecognitionActivity class and sends the username and password information there.

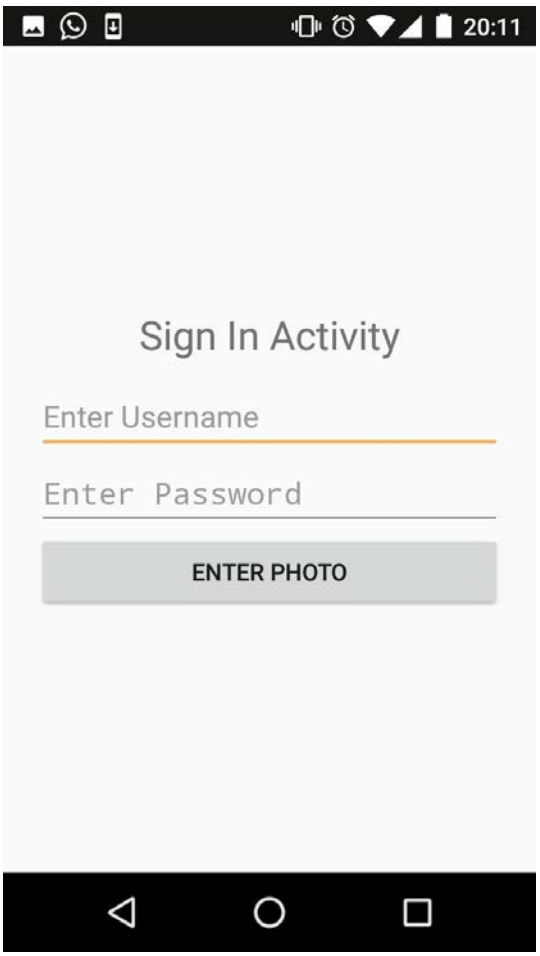

Figure 3: Sign In Page

RecognitionActivity

In this class, the camera starts working automatically and seeks for a face. If a face is caught, it compares it with recorded face files. If a match occurs, it sends the username, password and matched username to the welcome page.

WelcomeActivity

This class is the class that controls for the authentication made. It parses the matched username and splits it into two parts: username and hashed password. Then, it compares this hashed password with the hashed version of the user-entered password. It also compares the username that comes from the recognized photo and the user-entered one. If a match occurs, it opens the personal page of the user. In a mismatch situation, it returns back to the sign in page with an error message.

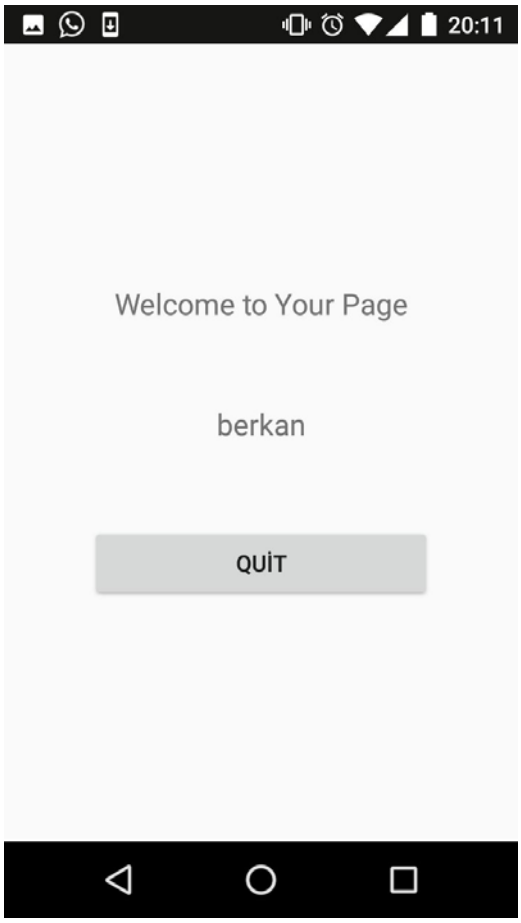

Figure 4: Personal Page

3. The Faced Problems

The first problem that we faced was importing necessary plugins for Android Studio. These import operations caused a significant loss of time at the beginning part of the project. Different projects require different development environments and necessary things included step by step.

The second problem was about finding a suitable face recognition library for our authentication project. There were a lot of face recognition libraries but many of them did not fit the operations that we wanted to do. These libraries also required several other libraries to work and this situation made it hard to work with them.

The third problem about the project was doing the authentication operation during the face catching phase, because the camera conflicts with our operation when it is open. Therefore, we gave the mission of authentication operation to the welcome activity.

The last but not least of our problems was storing the collected user info permanently. Two of the major solutions to this problem are using a database or a text file. We tried both of them; however, both operations caused several problems in Android and Java environment. Thus, we created a unique method which stores the username and hashed password in the name of face picture files.

4. References

[1] sladomic. (2017, 10 14). *Android Face Recognition with Deep Learning Test Framework*. Retrieved from github: https://github.com/Qualeams/Android-Face-Recognition-with-Deep-Learning-Test-Framework