Vaasan yliopisto
UNIVERSITY OF VAASA

PROJECT WORK

ICAT3130

Mobile Application Development

Examiner:

Sebastian Laitila

Students:

Fernando Pérez Parra - b113530

Ana Quintela González - b113438

TABLE OF CONTENTS page

LIST OF FIGURES page

1. INTRODUCTION

In the subject “ICAT 3130 Mobile Application Development” we have learnt to implement applications on mobile platforms. The main development tools in this course have been Java and Android, although, we also have learnt the concepts of application development over cross-platform.

In addition, we have extended our knowledge about Git, GitHub and GitLab, but the one that we have used has been GitHub (it is a development platform, from open source to business, where you can host and review code, manage projects and build software alongside 36 million developers).

All of this acquired knowledge has been the base to carry out our project. For its development, basically, we have used the program Android Studio, version 3.4, for Windows. The steps followed into it will be explained along the following sections.

But, before continuing, we will tell a bit about the aim of our application.

This application, called “Gotta guess ‘em all!” (due to the Pokémon sentence “Gotta catch ‘em all!”), is about guessing the name of the Pokémon after seeing the corresponding image. Three lives are available, so if the user fails the name three times, the game will be over.

Also, it will be another section with a list with all the images of the Pokémons used in the game with its correct names.

2. INTERFACE

Our project will be based in a Touch User Interface (TUI), which will allow communication between user and electronic device, as our mobile phone, by using the sense of touch through a sensible screen.

It uses the help of a Graphical User Interface (GUI) to represent graphically a control panel, that allows the user to interact with the said electronic device.

We will have five screens with the ones we could interact, which are a login, a main menu, a start of the game, a list with all the Pokémons and developer's info. Each of them will be explained below.

2.1. Menu

The menu consists in a screen with an image as background and four buttons over it.

Said image was downloaded and lately modified to adapt it to the style that we wanted in our application. In addition, we added the title and we extended the image to adapt it to the whole screen, by removing the status bar.

The buttons are located on the down part of the image. They were hidden by setting the attribute "background" as transparent and, in the same position, other images were placed to give it a better appearance. Other way to do it could be changing the attribute "visibility".

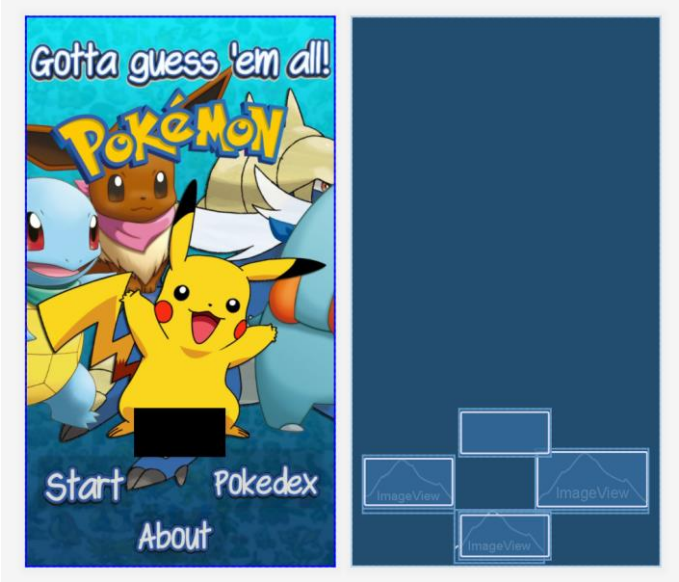

Figure 1. Black button before hiding it. *Source: own creation.*

Figure 2. Final menu screen. *Source: own creation.*

Each button, with no vibration, will take the user to a different screen already associated to it.

2.2. Start

This is the first of the buttons located in the menu. It is placed in the left down part of the screen, and after touching it, the screen referred to this section will appear.

On it, the user can find the following elements:

- Lifes: text block where the user can see the amount of lives that still has.
- Points: text block where the user can see the amount of points gotten after being right.
- Counter: hidden text block with a regressive counter.
- Image: image loaded of the Pokémon that the user has to guess.
- Text input: text block where the user has to write the answer.
- Button: button to validate if the answer is wrong or right.

Figure 3. Start screen. *Source: own creation.*

2.3. Login

After pressing the second of the buttons located in the menu, in the mid central part, a screen with two text blocks will be shown, where the user will have to introduce the email and the associated password, and it will be authenticated if touching the button associated this action.

There is also a button that allows the user to go back to the main menu.

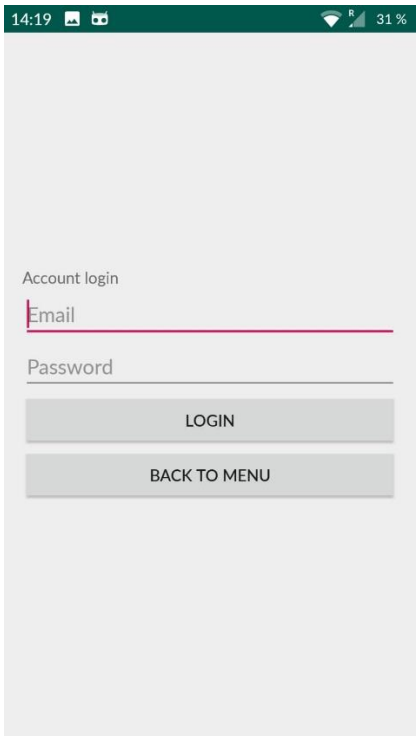

Figure 4. Log in screen. *Source: own creation.*

2.4. Pokédex

This is the third button of the menu, located on the right down part. Once it is touched, the screen associated with this section will be launched.

On it, a list with the one hundred and fifty first generation Pokémons can be found, that the user could navigate thanks to the scrolling. It is shown an image of the Pokémon and its name.

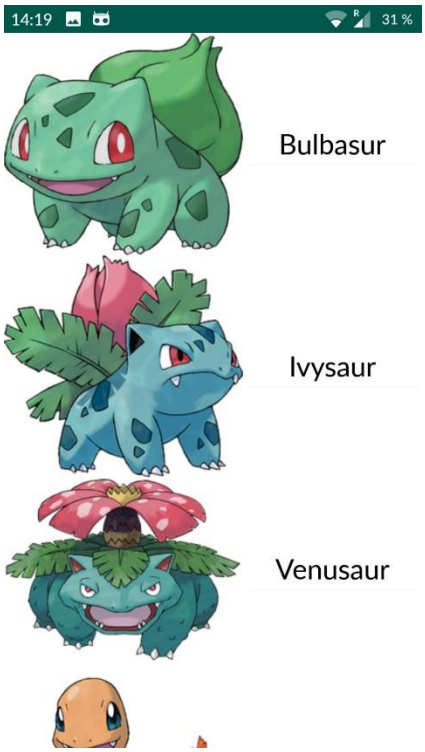

Figure 5. Pokédex screen. *Source: own creation.*

2.5. About

After the fourth and last button of the main screen has been touched, placed in the mid down part of said screen, the screen related with the information will appear.

The user could see the information about the developers: their names, the university where the study and the subject coursed in there.

That data is displayed over an image that we added as background. Also, under the names, there are two sprites of the Pokémon trainers.

Figure 6. About screen. *Source: own creation.*

3. USERS

The game has been developed for any public, with no age restriction. We remember ourselves and our older siblings playing the first-generation Pokémon games in the Nintendo, but nowadays, we also see a lot of teenagers and children still playing Pokémon. Due to this, we think it could be interesting for all of them.

In addition, elder people that may not be so interested concretely on Pokémon, they could show interest if they see the game as an application of memory, keeping up the brain fresh.

Figure 7. Application of memory. *Source: Pixabay.*

We decided to create a login with only email and password, as we can see in the figure below, even though we had more options such as using Facebook, GitHub, Google, Twitter, … But we wanted to give the user the biggest facilities, because maybe he has not any social media account, or Google account, or the age requested, etc., so choosing whatever email address he wants was the most adequate choice.

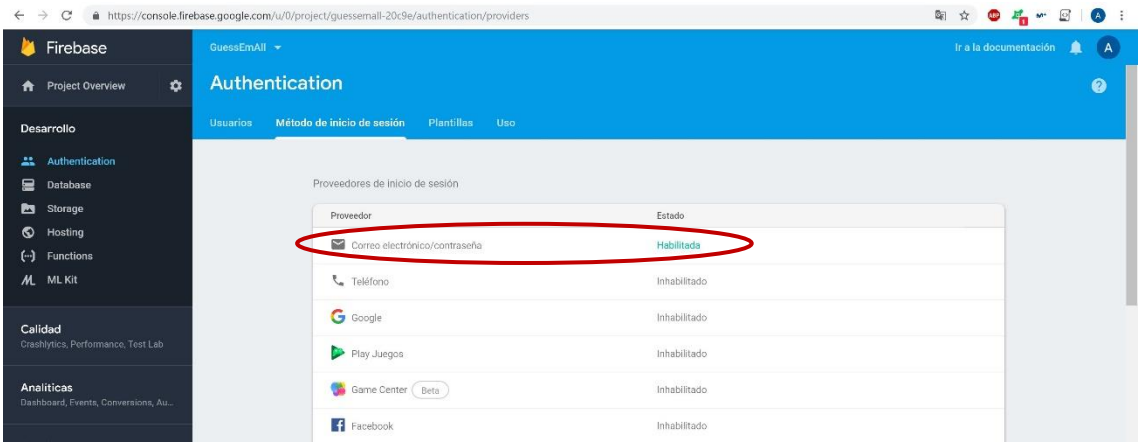

Figure 8. Email/password enabled. *Source: Firebase.*

Anyway, this login is optional, the user does not need to create or have any account. It will only be used, in the future, for keeping your scores.

We have used Firebase, which is a backend platform available for different platforms (Android, iOS, web), acquired by Google in October 2014 and improved also with the purchase of Divshot team.

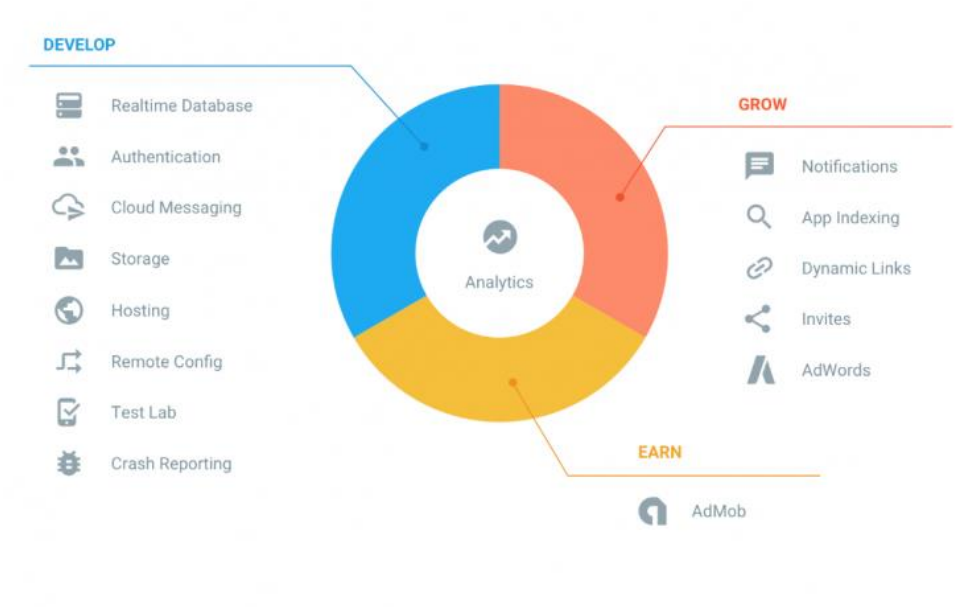

Figure 9. Firebase applications. *Source: elandroidlibre.elespanol.com*

Thanks to it, we can get that our application works with data in the cloud, due to Firebase provides an API to authenticate, save and synchronize data in the cloud in real time.

Its fundamental characteristics are divided into several groups, which can be grouped into:

- Analytics: Provides a free solution to have all kinds of measures (up to 500 types of events), to manage everything from a single panel.
- Development: It allows to build better apps, allowing to delegate certain operations in Firebase, in order to save time, avoid bugs and obtain an acceptable level of quality. Among its features include storage, testing, remote configuration, messaging in the cloud or authentication, among others.
- Growth: Allows users to manage applications and can also capture new ones. For this we will have features such as invitations, indexing or notifications.
- Monetization: Allows you to earn money thanks to AdMob.

But, although all those characteristics, the main reason of choosing this platform was that it was the one recommended in the guideline.

For creating the login, first of all, we created an account and a new project in the Firebase console.

After this, we added Firebase to our Android application, moving the *google-services.json* file into the Android app module root directory. This file contains configuration settings (URLs, API key, …) that we will need to communicate with the Firebase servers.

Next, we followed the steps to include the Google services plugin for Gradle which loads the *google-services.json* file that we downloaded. We also added some dependencies, as the library needed to use the Realtime Database and the one needed for Authentication.

Into our project in Android Studio, we also created two activities: one for the log in and another one for the sign up (but this last one is only commented and not actually working).

In there we will develop the code, that is already commented in the programming file.

Once is finished, if we have used the sign up, the user should have to register an account, that will pass through a validation such as having at least six characters long. When the registration would have been successful, in the Firebase console, we should be able to see the created user.

By using the log in, the user only needs to enter the email account and the password that he used when he created his own Firebase account.

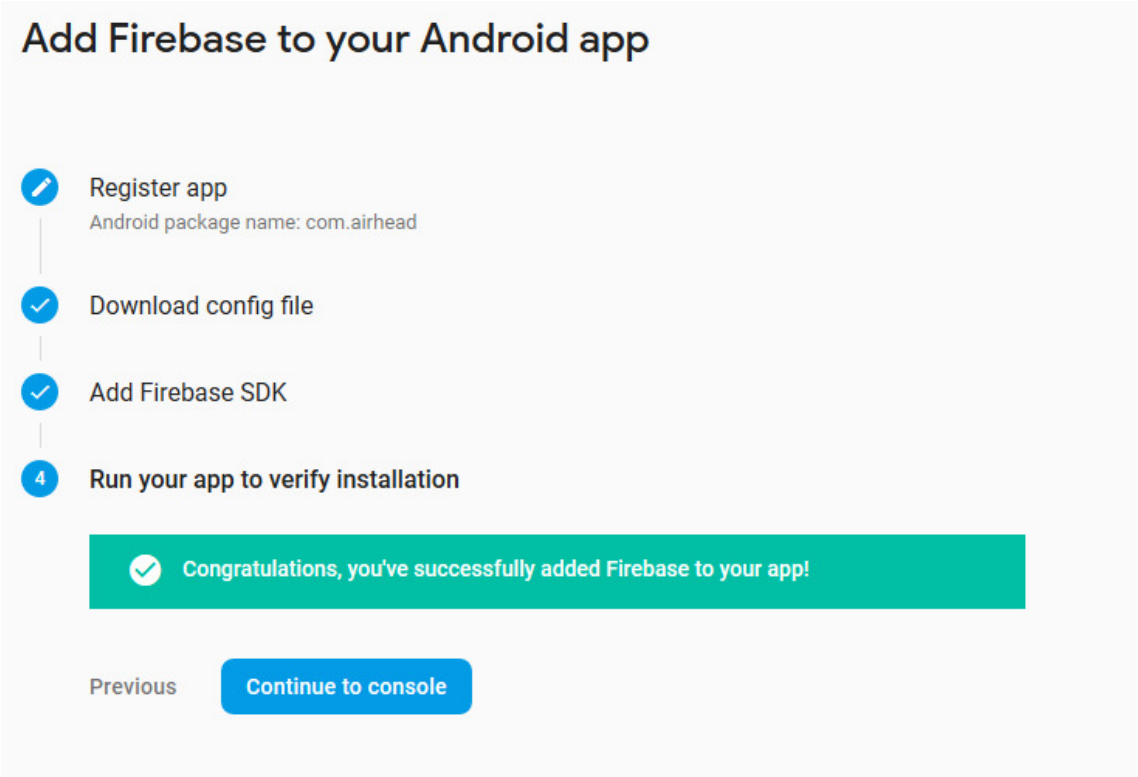

Figure 10. Confirmation of Firebase added. *Source: Firebase.*

4. FUNCTIONALITIES

We are going to explain the use of each screen and the items on it, while we follow a flowchart, that will help us to understand it better.

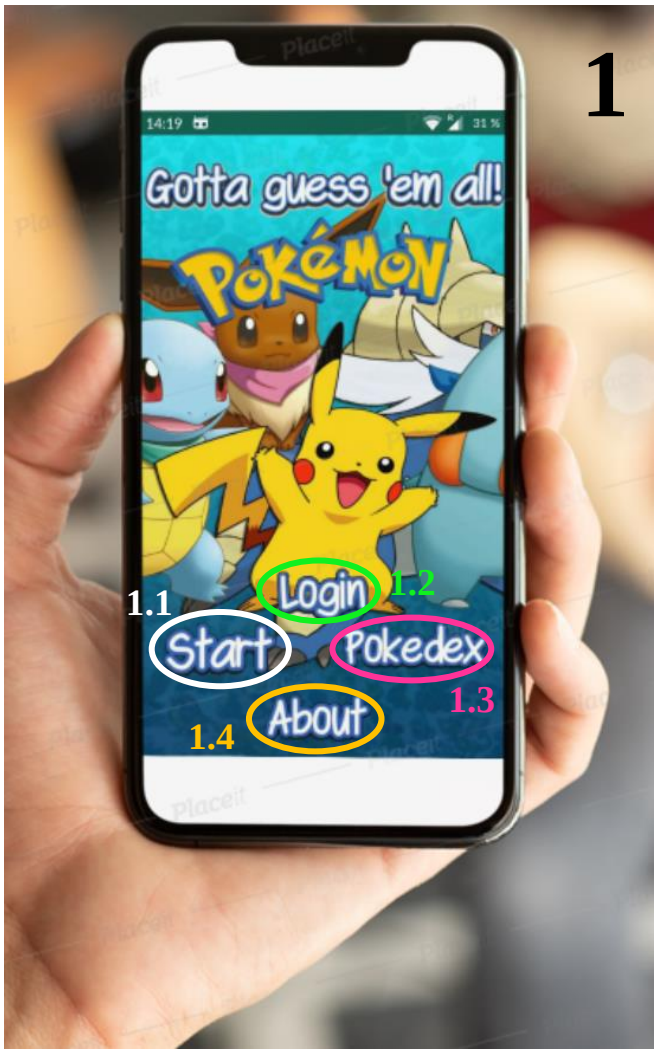

Figure 11. Menu with buttons (1, 1.1, 1.2, 1.3, 1.4). *Source: own creation in Placeit.*

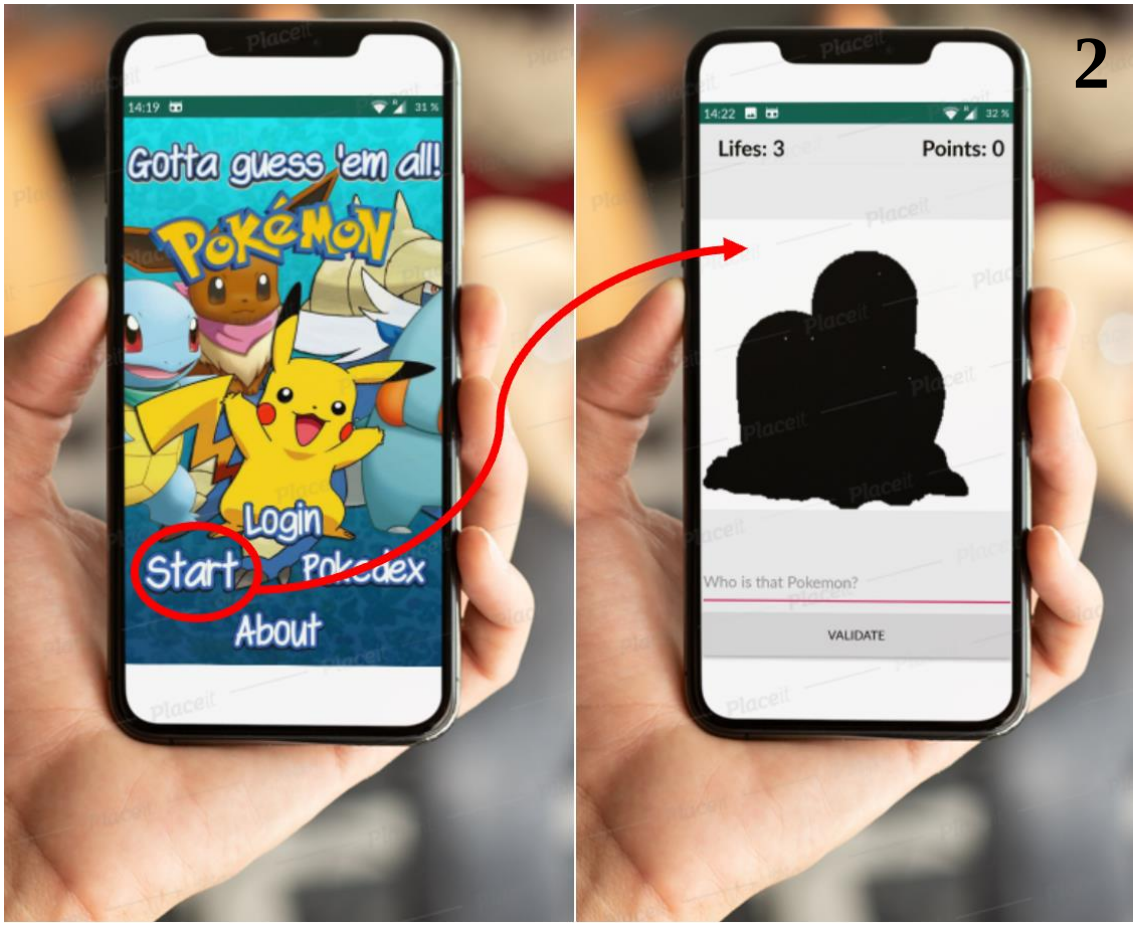

Figure 12. From menu to Start (1, 2). *Source: own creation in Placeit.*

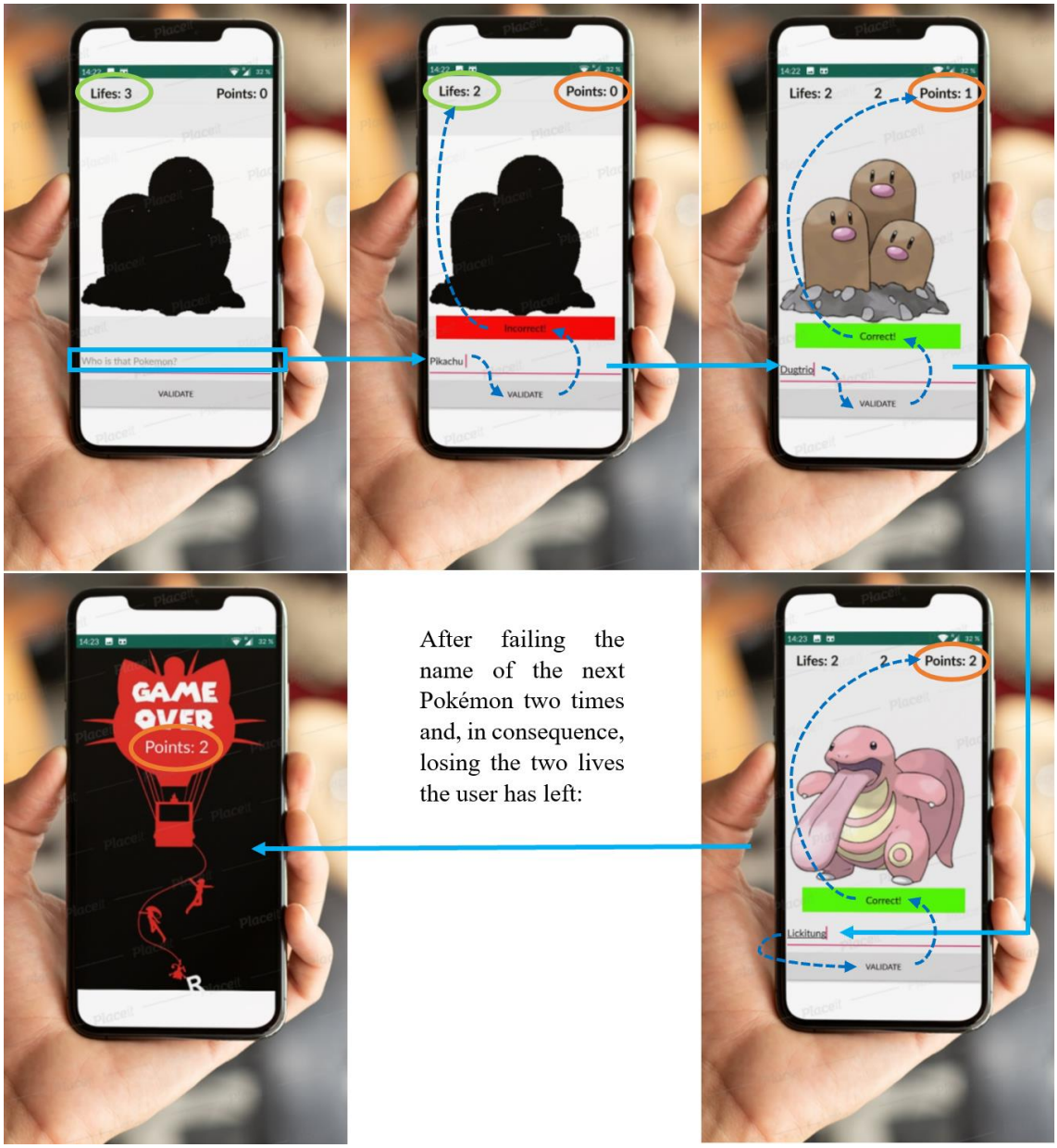

Figure 13. How the Start works (2). *Source: own creation in Placeit.*

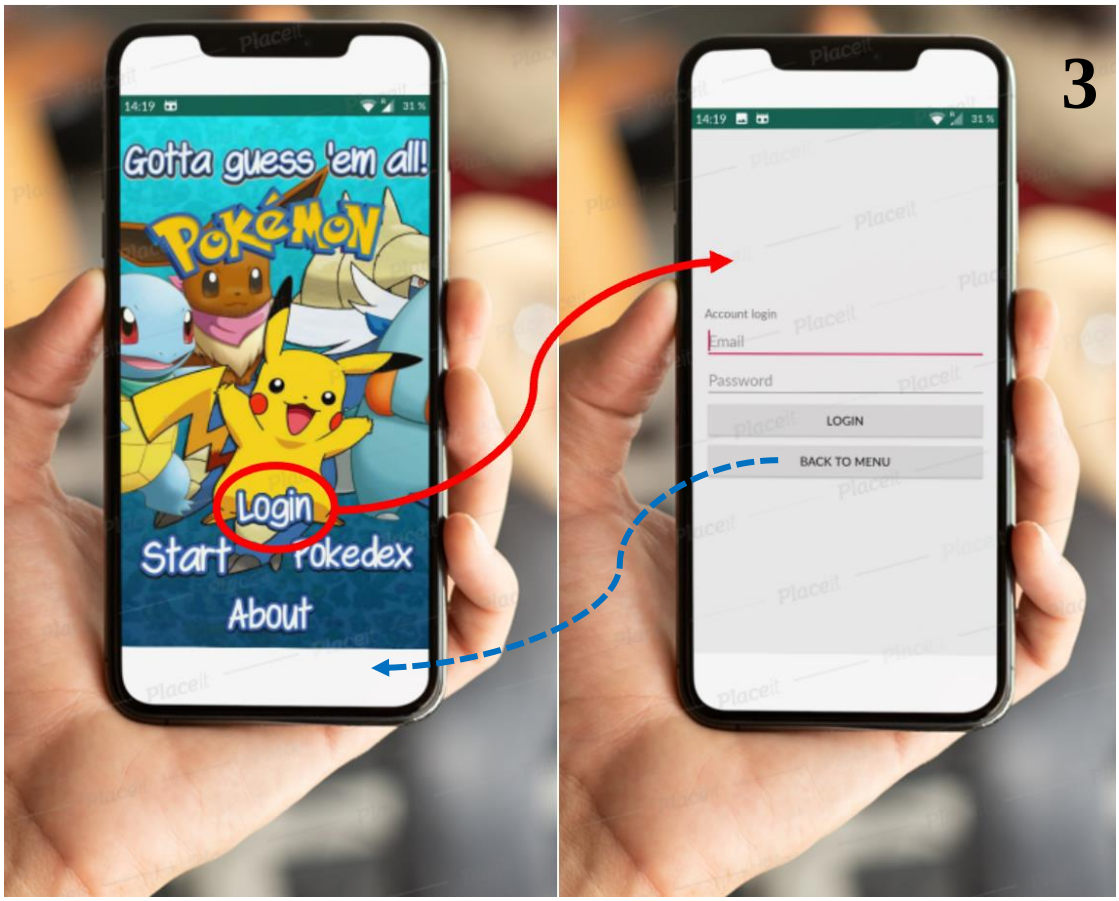

Figure 14. From menu to Login (1, 3). *Source: own creation in Placeit.*

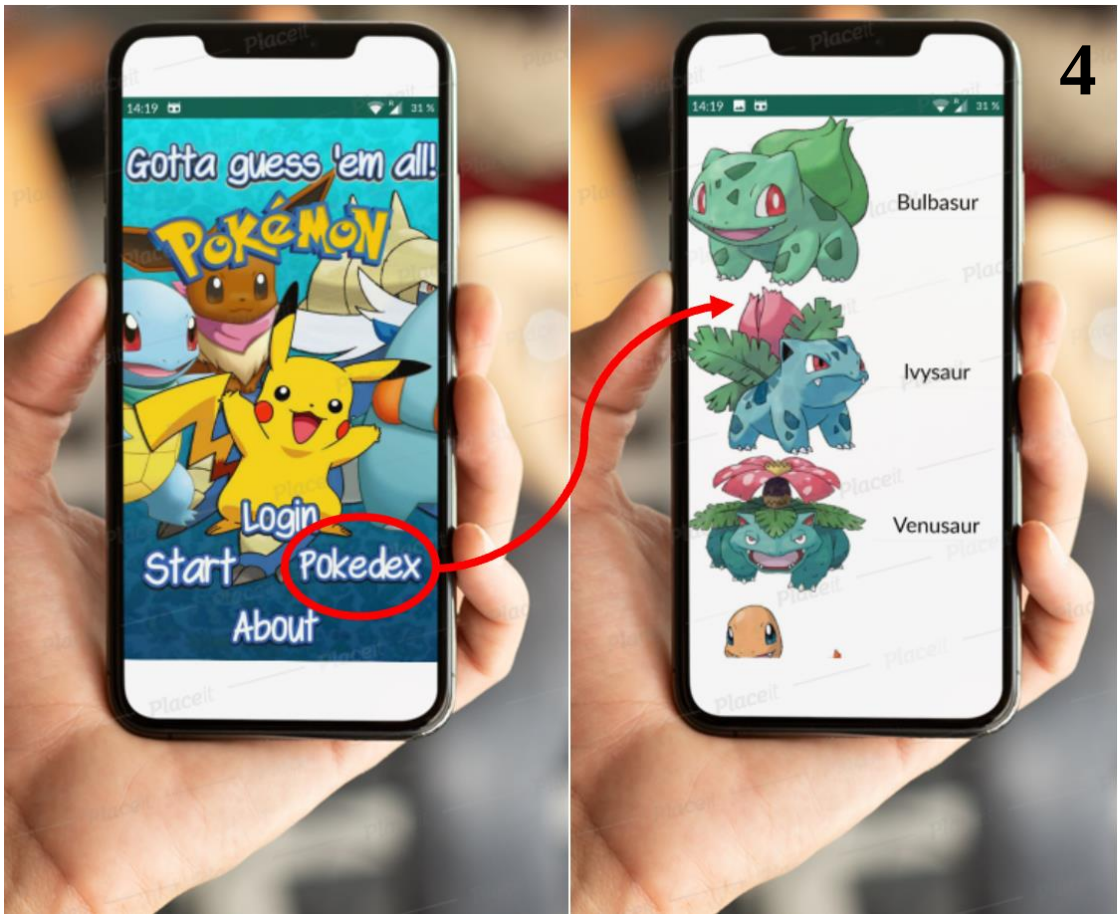

Figure 15. From menu to Pokédex (1, 4). *Source: own creation in Placeit.*

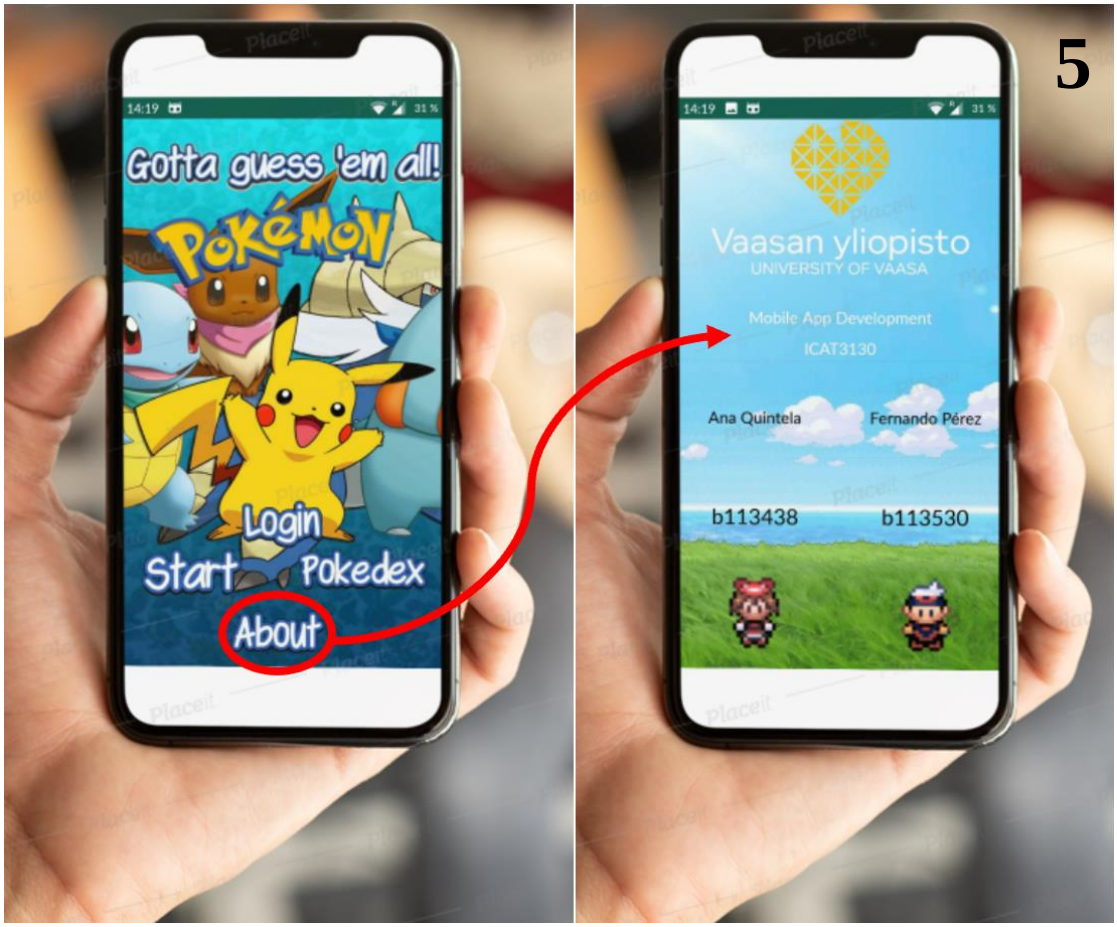

Figure 16. From menu to About (1, 5). *Source: own creation in Placeit.*

As we have already commented before, the first screen is the menu. *(1)*

In here four choices can be taken, due to there are four active buttons over the background picture.

If the user clicks over the button "Start" *(1.1)*, he will be directed to another screen *(2)* where he could see in the top bar the available amount of lives (is starts in three) on the left and the amount of gotten points (it starts in zero) on the right.

Under this, an image is launched. The image shows one Pokémon, and it will be the one that the user will have to guess the name.

Said name will have to be introduced by keyboard in the text block located just under the picture.

When the name is written, it needs to be validated by touching the active button "VALIDATE" that is below the text block.

If the name is correct, the points will be added to the counter of the right top corner and another Pokémon picture will appear. However, if the name is not correct, the user will lose one life, and the number of them will decrease in one unit, as he could see in the left top corner, and the same image will remain until the user guesses it or loses all the lives (at this point, the Team Rocket will announce the user the *Game Over* and another game will have to be started).

Supposing that the user had been selected the activity associated to the button "Login" *(1.2)*, a different screen would be shown *(3)*.

In here we have one option: initiating with an existing user, although in the future, a second option will be included, creating a new user account.

The email and the password will be validated. If the data is correct, the next screen will be launched; if not, a pop-up message will be displayed explaining the reason of why was not possible to log in and, then, the user could try again.

This will allow the user to access to his account, where in future versions, he could see the points gotten during all the games.

By other hand, if the user had been clicked on the button "Pokedex" *(1.3)*, it would be displayed another screen *(4)* where, by scrolling up and down, he could see a list with all the Pokémon images that appear in the game with the corresponding name (that will be the correct answer to the game). In this way, the user can study or check the solutions.

Imagine that the user had been touched the button "About" *(1.4)*, in this case the last screen will be launched *(5)*.

All the information about us, the developers of the application (Fernando Pérez and Ana Quintela), the subject and the university will be shown.

To give it a better look, an image of one of the Pokémon landscapes is displayed, and two spirits of the trainers, that we have made correspond with each one of us.

From all these screens, touching the back button of the used device, it will go backward from screen to previous screen.

Finally, we have made an installable version of the application, an apk file. So, the user only needs to wait a few seconds and it will be ready.

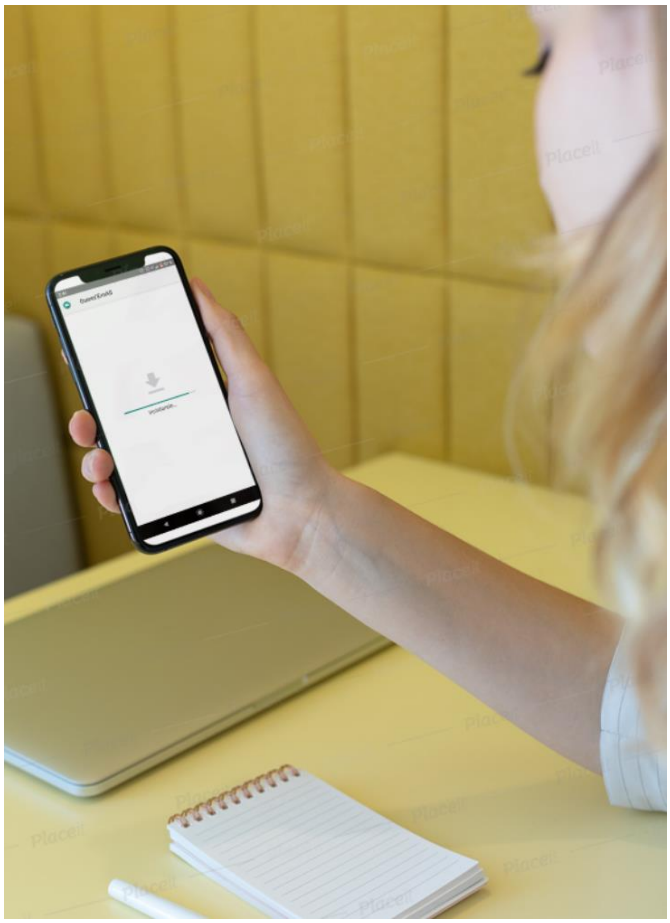

Figure 17. Installing apk file. *Source: own creation in Placeit.*

5. CONCLUSIONS

After developing this project, we have familiarized with new language programs as Java, since until now we have only used C and C++. Even though it is quite similar to them, plenty of differences can be found.

In addition, we have used it into Android Studio, where we have mixed it with XML (Extensive Markup Language) to get the desired result. The XML has been in charge of the data (the design of the layout is so much easier form here) and Java of the logic, as it could be interactions between the app, elements…

Also, we have learnt more about JSON (JavaScript Object Notation), which helped us to transmit and exchange data as an easier alternative to XML. And speaking about help, we also should speak about Firebase, that facilitate the creation of the login and authentication.

Thanks to all of this, now we feel more comfortable and confident in the field related with the development of applications. Even if we still need to improve and learn a lot about it, this has been a good start to get closer and increase our interest about it.

To conclude, we would like to propose some improvements that could be implemented in the future:

- Add the scroll in all the screens, so if the user turns the device to horizontal the application would work perfectly. Right now, it is only thought to be used in vertical (except for the Pokédex that is already done).
- Update the list with more Pokémons or creating a map with the regions and the Pokémons that can be found there.
- Finish the registration option.
- Once the user has logged in, find online the record of points and may share it with friends.
- Optimize the Java code in the list by using loops and reduce the wait time.

6. REFERENCES

<URL:https://placeit.net/>

<URL:https://pixabay.com/es/>

<URL:https://developer.android.com/studio>

<URL:https://firebase.google.com/>

<URL:https://www.youtube.com/watch?v=EdtlQc1RTVo>

<URL:https://www.youtube.com/watch?v=oi-UAwiBigQ>

<URL:https://www.giantbomb.com/profile/wakka/lists/the-150-original-pokemon/59579/>

<URL:https://www.youtube.com/watch?v=Lx1e_fdnNxA>