ESAT
INNOVATION SCHOOL
ESCUELA SUPERIOR DE ARTE Y TECNOLOGÍA

Asignatura 2019-2020

Mobile
Briefing

Course: Programación Mobile
Associated Module: App Development Frameworks
Teacher: Rubén Blanco
Year: 2023/2024
Author: Manuel Alcázar López

Index

1. - App Project

The project is an app developed in Kotlin for Android, which shows a list of different songs and information about them, such as name, group, album and the image of the album. The project includes features like user login, browsing along the music list, changes between activities and app screens, and JSON parsing and web mocking to provide remote data access.

The structure of the project use the standard Android project layout, with the actrivities and fragments are dividev in different folders, the netwok API classes and setup into another folder, the different UI elements and resources into their respective folders and all the classes related to the music and song, or the information that is displayed in the application is also in its own folder.

The main charactersitics of the project are:

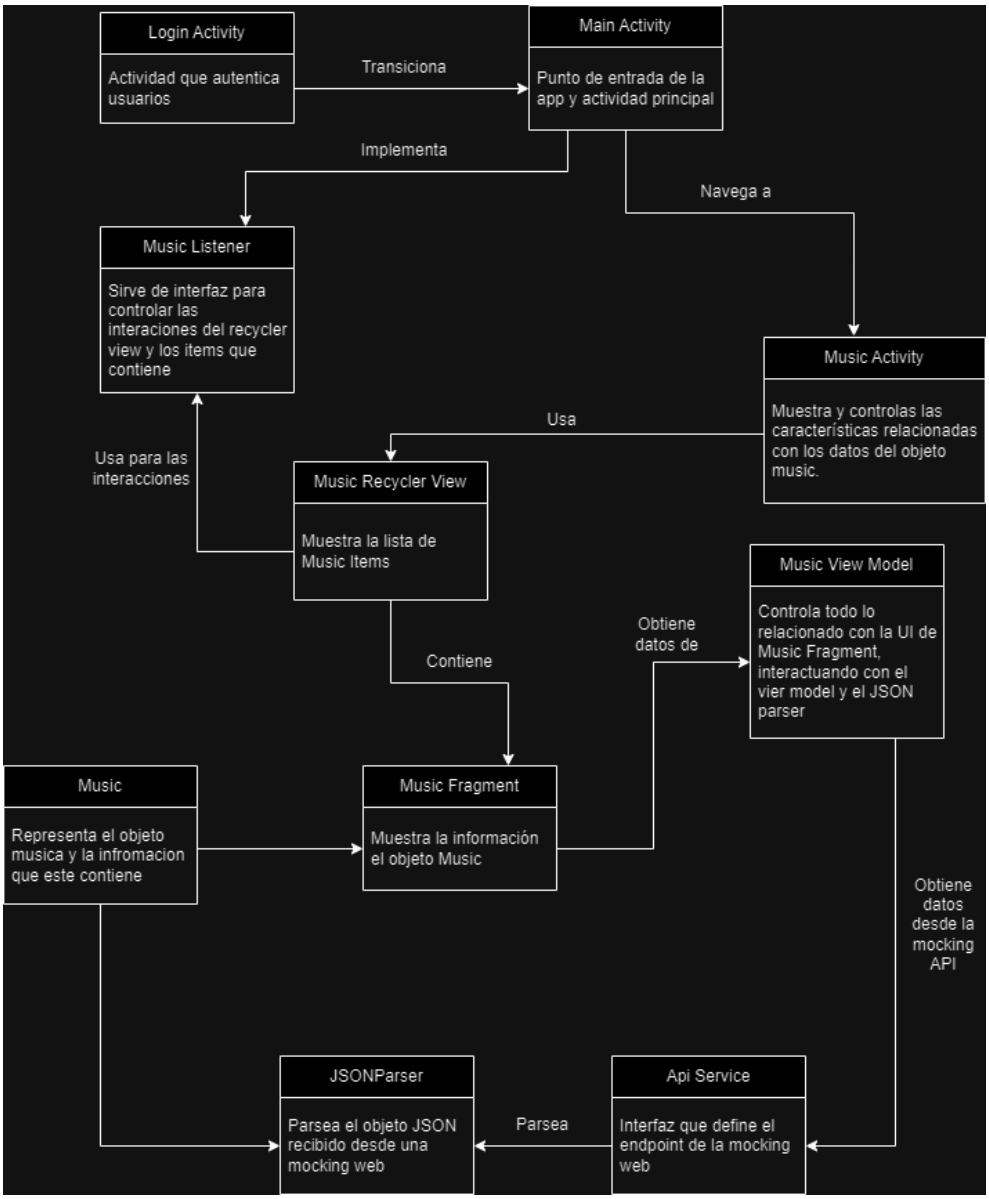
- **User Autentication:** with its own activity and resouces for the login interface, although the user is not persistant, and its hardcode; to access the user is "*wllop*" and the password is "*a123456**"
- **Browsing and Interaction:** achieve through the use of activities, fragments, view models, and recycler views; to display music and navigate through them.
- **Network mocking and JSON parsing:** using Retrofit dependency to handle network request, to fetch the music data for the application as a JSON Object, to be parsed into Kotlin code for the application.
- **Firebase Crashlyctics:** to provide and test the application from possible crashes and exceptions, and to analyse the project.

The project uses the ViewModel classes and the Activities and Fragments, which is a common development pattern in modern Android applications. This helps to separate the logic data from the UI logic and provides better handling of the project. Also the project includes extre dependencies using Gradle, for is correctly functioning.

There is also an example test, which is typically used for testing the UI components and the interactions with the simulated or emulated devices, to ensure the app behaves as expected in a real-world environment.

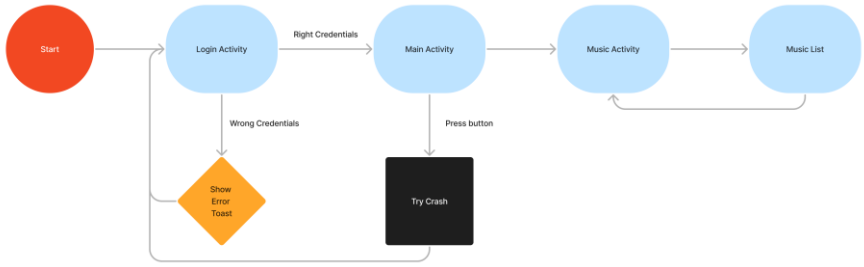
Overall, the project demonstrates a well structured Andoird application with a clear separation of classes and trasks following the concerns of modern Android development practices. The structure is modular, making it easier to maintain and extend in a future, and the presence and use of test and Firebase indicates a focus on ensure the app is realiable and functional.

The project‘s classes have the following functionality and relationship:



2. - UX Flow Chart

The navigation through the application can be displayed like this:



3. - Strategies and Solutions to the Development Problems

There were several problems that have appeared during the development of the application. In first place, the CPU of my compuer was incompatible with Intel HAXM, since I posses an AMD processor. To solve this problem I have to change the setup of the bios to be able to install the emulator accelerator. Even with the emulator accelerator, the IDE is in general slow and problematic. The UI can be overwhelming some times.

The main problem when developing the application is that I did lose the previous project I had made. And since there were a lot of time since I did not use the IDE and program in Kotlin, I basically had to learn everything almos from the beginning, fortunetly Kotlin is easy to learn and I rebembered some things that were key to develop the application, such as change the version SDKs, add dependencies, and how the activities interacts with each other.

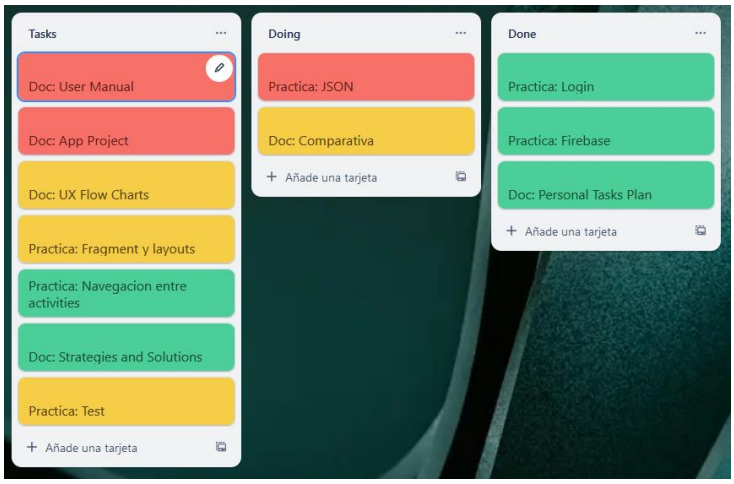
Related with this last one problem were the changes of the Android Studio's API versions. The new changes in Gradle dependencies made hard to add new modules and dependencies to the new SDK, since the way they are include, as the same as the name of the dependencies had change from the last time I used Android Studio. Although, I finally manage to find the dependencies I was looking for, by changing the SDK version to a previous one, and then updating it again to change the dependencies to the new ones.

Finally, the main issue come from the JSON mocking and parser. The issue radicates in that I did not know what was the structure of a JSON list object, and the parser could not conver the data to be use by the application, but finally searching how this structure was, I was able to solve the problem, but it was obnoxious.

4. - Personal Tasks Plan

To accomplish the different stages the project posses, this was divided into different tasks for a better managing and organazing. The tool used to create a personal tasks schema and follow the state of the stages is Trello.

The task are divided between the app tasks and the documentation tasks, and the color of the tasks reflects the amoun of time will take the task to be finished, being red the longest or hardest, and green the easiest.



Here is a link:
https://trello.com/invite/b/OzHOIrmp/ATTIe79d069da7fafd0ccd0fa3d673ab210fCF8C14B2/mobile

5. - User Manual

To use the app in first place we need to meet certain system requirements to install Android Studio and open the project files. These requirements are:

- **Operating System:** Window, macOS, or Linux.
- **RAM:** minimum 8GB, recommended 16GB.
- **Storage:** minimum 10GB free space.
- **CPU compatible with Intel HAXM.**

Before we can open the project we need to setup Android Studio, we can find the installer from the official website, where we can download the installer files. Then we need to install Android Studio following the setup instructions of the installer.

Once is finished, we can launch Android Studio and open the project. When the project is already open we may need to synchonyzne the Gradle files, we can do this going into the build.gradke.kts file located inside the src folder of our project, and click in “Sync Now” button displayed on the top of the build.gradle.kts file.

After syncing, we can build the project by navigating to “Build > Make Project” or pressing “Ctrl+F9” on Windows / Linux or “Cmd+F9” on macOS. This will start building the application, ensuring that there are no errors, if any error ocurs the detail will be displayed in the “Build” window to resolve the current issues.

Once the project is builded, we just need to run it using an emulator, but before we can run the application in an emulator we need to setup one, although we can use a physical device too. To set up an emulator we need to move to the Android Virtual Device (AVD) in the “Tools” section up in the IDE’s editor. If we want to connect instead a physical device we need to activate the “Developer Options” of our smart phone and then the “USB debugging”, which is a process that may vary from one device to another.

After everything is setted up, we can run out application by clicking in the “Run” botton in the top side of the editor or press “Shift+F10”. Android Studio will install the application of the selected device or emulator and will lauch it automatically.

Upon launching the application, the login screen will be displayed, the credentials for authenticate the users are hardcoded (“wllop” for the user and “a123456*” for the password), and there is no data persistence in the application. If we enter the wrong credentials a small toast will appear showing us that the credentials are incorrect, and therefore we have no logged in. If we ente the correct credentials and logged correctly we will navigate directly to the main screen.

From this screen we can access to the displayed elements and browse and navigate through them, we can also tap on them to view more details. We also have the chance to crash the application, although this is use only for testing the Firebase Charslytics.

6. - Android vs Xcode

6.1 Android Studio:

Android studio is the official IDE (Integrated Development Enviroment) of Google, for its operating system Android. It’s based on IntelliJ IDEA, for Android development and incorporates code editing and developer tools. Android Studio supports different features and tools such as:

- **Android Emulator.**
- **Composite.**
- **Gradle-based build system.**
- **ProGuard integration.**
- **Code templates.**
- **GitHub integration.**
- **Support for Android Wear apps.**
- **Java, Kotlin, and C++ support.**
- **Debugging.**
- **User Interface Desing.**

Android studio was announced on May 2023, and released in December 2014, as a replacement for Eclipse Android Development Tools as the primary IDE for native Android application development.

6.2 Xcode:

Xcode is the Integrated Development Enviroment created by Apple, to develop applications and software for Apple’s different operating sytems and devices such as, macOS, iOS, iPadOS, watchOS, tvOs, and the new visionOS. Xcode was first released in 2003, and its latest version was released in September 2023.

It posses a wide set of features for developers:

- **Wide variety of programming languages.**
- **Emulators and simulators of the different Apple devices.**
- **Reality composer, for 3D and AR experiences.**
- **Fat binary or universal binary to ease transitions between different devices architectures and CPUs.**
- **Custom Machine Learning.**
- **Debugging tools,**
- **Intuitive UI desing.**
- **Cloud services.**
- **Profiling tools.**

6.3 Comparison:

Both Android Studio and Xcode posses the common features present in any IDE (Integrated Development Enviroment) since both are used for developing applications for different platforms:

- **Debug Tools.**
- **Profiler Tools.**
- **Memory Analysis and Tracking Tools.**
- **Code Editing**
- **Device Emulators and Simulators.**
- **User Interface.**
- **API Documentation.**
- **Templates.**
- **Pre-built components.**

All of these development tools are high quality and helps developers to save time and effort. It also provides ease of use and better user experience when developing complex apps, that with all the tools provided can be profiled and optimizaed for specific platforms and allows developers to deliver bug-free applications.

But since both IDEs are targeting very different platform, their common tools provides different features and the IDEs also posses' uncommon features and characteristics such as:

- **Platform:** as mentioned before they target different platforms, Android Studio is primarily used for developing applications for Android, while Xcode is focuses in Apple's devices such as iOS or macOS.

- **Programming Language:** one of the most significant differences between Android Studio and Xcode is the amoun of programming languages they support. Android Studio supports primarily Kotlin and Java, along with C++, but Xcode supports Switf, C, C++, Objective-C, Objective-C++, Java, AppleScript, Python, Ruby, Rez, Ada, Go, Perl, D and C#, and other languages like GNU Pascal and Free Pascal.

- **User Interface:** both have different UI that provides different user experience. Android Studio uses the "LayoutEditor" a drag-and-drop layout editor for visually desing the user interface of the apps, allowing developers to quickly develop their applications. Xcode uses a combinations of drag-and-drop, tools and code, called "Interface Builder", that allows more app development complexity in an easier way.

- **IDE Desing Framework:** Android Studio and Xcode have a very different philosphy about how to structure the code and navigation of the apps. Android use partition, the app is break into activities and fragments. The activity is the equivalent to one app screen, since the application can posses multiple screens, each of the activities uses fragments, to navigate between the activities. Instead, Xcode uses different types view controllers, which can control the entire screen or one part of it. These controllers can be managing in a several ways, through code, images, or XML files and more.

- **Availability:** both IDEs are proprietary software, but Android Studio is avaliable for Windows, macOS, and Linux, making it cross-platform. Xcode is only available for Apple devices. This makes a great difference, since Apple devices are expensive to obtain compared to the other devices, due this, Android Studio is a more available and flexbile option for developers.

- **Target Limitations:** there are different limitations related to the app distribution and emulation tools these IDEs provides. Xcode's emulators are limited to Apple devices, while Android Studio allows testing the apps in various Android devices with different operating systems and hardware profiles. Also Android allows to distribute the apps through different channels, from Google Play to any other third-party app store, Xcode, instead, only allows distribution through the Apple's App store.

In conclusion both IDE offers their pros and cons. The flexibility and accessibility of Android Studio allows for more market share and wider app distribution, but since there are a lot of different devices with different features with Android as operating system, the developers may have issues while developing and distributing their apps. In Xcode, although is has more limited market share, and only one channel of app distribution, since is focused in Apple devices, the development, and optimization of the applications is easier.

Another important different between Android and Xcode relies on security, Android applications are not encrypted, but obfuscate, and the operating system is also more vulnerable to cibersecurity threats, compared to iOS or Xcode applications which are encrypted.

6.4 Evaluation:

The environment used for the development is Android Studio for its availabity and ease of use. The application is developed according to the Android Studio's framework, using its main characteristic of code partition using activities and fragments. The project uses extra modules and dependencies such as:

- **Firebase Crashlytics:** mainly used to test possible crashes in the application.
- **Retrofit:** for JSON parsing and web mocking to load data into a list.
- **Glide:** for image load ad managing for Android.

These dependencies also use different frameworks and API's. Althogh the application is scalable in the long term, the different updates of the Android Studio's API, makes the project unease to maintain in the long term, since the changes from the version of Android Studio are so different, some dependencies and code parts needs to be change from one version to another, which is an issue for developers once the application have become complex and bigger.

6.5 Analysis:

The using of Android Studio has been a trouble since the beginning. My computer has an AMD processor that is not compatible with Intel HAXM, used for accelerating android device emulators, which has make the development process slow. Also, Android Studio consumes a lot of resources, and has the same as before, the lack of RAM, has also make the development slow.

Also, Android Studio's user interface can be overwhelming if do not have the knowledge. Overall, the development with Android Studio has been slow and obnoxius. This project has consumed more time than expected, because the code is neither complex nor large.

7. – Xcode Implementation Approach

Bibliography

CONTRIBUTOR, TechTarget. What is Android Studio?: Definition from TechTarget. *Mobile Computing* [online]. 23 January 2023. [Accessed 18 May 2024]. Available from: https://www.techtarget.com/searchmobilecomputing/definition/Android-Studio

Android Studio. *Wikipedia* [online]. 16 May 2024. [Accessed 18 May 2024]. Available from: https://en.wikipedia.org/wiki/Android_Studio

What is Xcode: Features, installation, uses, advantages and limitations. *BrowserStack* [online]. 25 April 2023. [Accessed 18 May 2024]. Available from: https://www.browserstack.com/guide/what-is-xcode

Xcode. *Wikipedia* [online]. 14 May 2024. [Accessed 18 May 2024]. Available from: https://en.wikipedia.org/wiki/Xcode

5 major differences between IOS and Android App Development. *EGO Creative Innovations - we design apps that your users love* [online]. [Accessed 18 May 2024]. Available from: https://www.ego-cms.com/post/5-major-differences-between-ios-and-android-app-development

Android Studio vs. Xcode – Pros and cons of each application development platform. *iubenda* [online]. [Accessed 18 May 2024]. Available from: https://www.iubenda.com/en/help/127491-android-studio-vs-xcode-pros-and-cons-of-each-application-development-platform

Android Studio vs xcode: What are the differences? *StackShare* [online]. [Accessed 18 May 2024]. Available from: https://stackshare.io/stackups/android-studio-vs-xcode

ESAT INNOVATION SCHOOL | ESCUELA SUPERIOR DE ARTE Y TECNOLOGÍA