

**REPUBLIC OF CAMEROON**

**Peace-Work-Fatherland**

**MINISTER OF HIGHER EDUCATION**

**FACULTY OF ENGINEERING**

**AND TECHNOLOGY**

**REPUBLIQUE DU CAMEROON**

**PAIX-Travail-Patrie**

**MINISTRE DE L’ENSEIGNEMENT SUPERIEUR**

**FACULTE D’INGINERIE**

**ET TECHGNOLOGIE**

**CEF440: INTERNET AND MOBILE PROGRAMMING**

1. **Different type of mobile app that exist**
2. **Review of programming languages used for mobile app**
3. **What’s mobile app frame work, list and compare them**
4. **Collect and analyze the requirement for a mobile app**
5. **Estimate the mobile app development cost**

**Presented by:**

**BITO’O BI HIEGA II DYLAN FLORENT FE20A024**

**CHEZEM DONGMEZA MIGUEL FE20A026**

**DJIOTSA DJOUAKE CHRISTIAN DARYN FE20A029**

**DJOUKENG NOUGNING YANNICK IVAN FE20A030**

**MADADJO KUITCHE MONICK CYBELLE FE20A060**

**INSTRUCTOR**: **Dr. NKEMENI VALERY**

**March 2023**

**MOBILE APPLICATIONS:**

**What is a mobile application?**

A mobile app (or mobile application) is a software application developed specifically for use on small, wireless computing devices, such as smartphones and tablets, rather than desktop or laptop computers. Mobile apps are sometimes categorized according to whether they are **web-based** or **native** **apps**, which are created specifically for a given platform. A third category **hybrid apps** combines elements of both native and web apps.

1. **Types of mobile application:**

There are three basic types of mobile application:

* **Native applications:** these apps are created for one specific platform or operating system.
* **Web applications:** theseare responsive versions of websites that can work on any mobile device or OS because they are delivered using a mobile browser.
* **Hybrid applications:** theseapps are a combination of both native and web apps, but wrapped within a native app, giving it the ability to have its own icon or be downloaded from an app store.

1. **Natives applications:**

Native Apps are written in a specific programming language to work on a particular Operating system. A majority of the smartphones either run an Android OS or the iOS if it’s an apple device. Native apps are generally built to make the most of all the features and tools of the phones such as contacts, camera (…). Because of their singular focus, native apps have the advantage of being faster and more reliable in terms of performance. They’re generally more efficient with the device’s resources than other types of mobile apps.

1. **Advantages of native applications:**

* Fast performance due to simple code specific to device and OS. (Because of their singular focus, native apps have the advantage of being faster and more reliable in terms of performance).
* Better use of OS and device specific functionalities.
* Interactive UI/UX. (Native apps utilize the native device UI, giving users a more [optimized customer experience](https://clevertap.com/blog/customer-experience-optimization/)).
* Lesser compatibility issues and faster to configure.

1. **Disadvantages of native applications:**

* Building OS specific apps can be time-consuming (the problem with native apps lies in the fact that if you start developing them, you have to duplicate efforts for each of the different platforms).
* OS specific programming languages like swift and java are hard to learn.
* Requires separate codebase to add new features. (Every time there’s an update to the app, the user has to download the new file and reinstall it.)
* It takes precious space in the device’s storage.

1. **Web applications:**

Web apps or mobile web apps can be accessed from an internet browser window. It does not require any storage space or installation process to use the app. They adapt to various screen sizes and devices easily. They behave similarly to native apps but are accessed via a web browser on your mobile device. One kind of web app is the progressive web app (PWA), which is basically a native app running inside a browser.

1. **Advantages of web applications:**

* Reduced business cost. (This cuts down on development costs.)
* No installation needed. (Because it’s web-based, there is no need to customize to a platform or OS)
* Better reach as it can be accessed from anywhere.
* Always up-to-date. (Just push the update live over the web. Users don’t need to download the update at the app store/play store).

1. **Disadvantages:**

* Web apps fail to work when you are offline. (Because they’re shells for websites, they won’t completely work offline)
* Limited number of functionalities as compared to Native apps. (There will be functionalities available within one browser and not available on another, possibly giving users varying experiences.)
* It takes a longer time to develop.
* Security risk. (Even if they have an offline mode, the device will still need an internet connection to back up the data on your device, offer up any new data, or refresh what’s on screen.)

1. **Hybrid applications:**

Hybrid apps combine the best of both native and web apps. The hybrid apps are written using HTML, JavaScript, and CSS web technologies and work across devices running different Oss. Hybrid Apps are built on a single platform and distributed across various app stores such as Google Play store or Apple’s app store similar to Native apps and are best used when you want to build apps that do not require [high-performance](https://www.pcloudy.com/mobile-app-performance-monitoring-basics-to-advanced/), full device access apps. Examples of hybrid apps include **Facebook, twitter and LinkedIn.**

1. **Advantages of hybrid mobile applications:**

* . Easy to build. (Building a hybrid app is much quicker and more economical than a native app)
* Shareable code makes it cheaper than a native app. (a hybrid app can be the minimum viable product)
* Easy to push new features since it uses a single code base. (there is much less code to maintain.)
* Can work offline. (They also load rapidly, are ideal for usage in countries with slower internet connections, and give users a consistent user experience)
* Shorter time to market, as the app can be deployed for multiple OSs.

1. **Disadvantages of Hybrid apps:**

* Complex apps with many functions will slow down the app.
* More expensive than web apps
* Less interactive than native apps
* Apps cannot perform OS specific tasks (Hybrid apps might lack in power and speed,).

1. **Programming languages used for mobile applications development:**

There are several programming languages used for mobile app development, each with its own strengths and weaknesses. We have among others:

* **Javascript:** it is a popular language that is used for developing cross-platform mobile apps using frameworks like React Native, Ionic, and PhoneGap. It's widely used for web development and offers a large number of libraries and frameworks.
* **Java:** it is one of the most widely used languages for developing Android apps. It has a large standard library and is highly versatile, making it suitable for developing complex applications.
* **Swift**: it is a modern language developed by Apple for building iOS, iPadOS, macOS, tvOS, and watchOS apps. It's a fast and efficient language that is easy to read and write.
* **Kotlin**: it is a relatively new language that has gained popularity in recent years. It's an officially supported language for Android development by Google, and it's interoperable with Java, making it easy to use alongside Java in Android projects
* **Dart**: Kotlin is a relatively new language that has gained popularity in recent years. It's an officially supported language for Android development by Google, and it's interoperable with Java, making it easy to use alongside Java in Android projects
* **Python:**
* C#: is a language developed by Microsoft that is used for developing Windows, Xbox, and Windows Phone apps. It's also a popular choice for game development using the Unity engine.

In the table below, we can see the different programming languages and their features:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Programming**  **language** | platform | Operating  system | Level of difficulty | Framework | Programmer experience |
| javascript | Hybrid app  Web apps | Android and IOS | Easy to learn | React native | beginner |
| kotlin | Native app | Android | Easy to learn | Native android SDK | beginner |
| Java | Native app | Android | Easy to learn |  | intermediate |
| Swift | Native app | IOS | Easy to learn | Native IOs | beginner |
| Objective-C | Native app | IOS | Easy to learn | - | beginner |
| C# | Hybrid apps | Android and IOS | intermediate | Xamarin | intermediate |
| Dart | Hybrid apps | Android and IOS | Easy to learn | Flutter | beginner |
| Python | Hybrid apps | Android and ios | One of the easiest PL | kivy | beginner |
| C++ | Hybrid apps | Android and IOS | Hard to learn | Microsoft tool in VS | professional |

1. **Mobile applications development frameworks:**

A mobile App development framework supports and accelerates the process of application development. Although native apps offer an unmatched experience in terms of aesthetics and functionality, native apps are costly to build and maintain. The diversity of devices and platforms makes maximum user penetration challenging with native apps. The solution lies in creating a web, hybrid, or cross-platform mobile app for business with budget constraints by leveraging mobile app development frameworks. Now let us talk about the most known mobile application development frameworks.

1. **Flutter:**

This is a Google’s open-source framework for developing native Android and iOS apps using a single codebase. It is revolutionary software Development Kit (SDK) for cross-platform app development that stands out as it uses a unique approach for delivering apps with a native-like look and feel.

1. **Features:**

* Ability to develop high performance apps.
* It is comprehensive and precise.
* Flutter is a complete framework that includes widgets, a rendering engine, testing and integrating APIs and tools to help developers build and deploy aesthetically pleasing mobile apps.

Examples of applications built using Flutter development include **Google, Alibaba, and Abbey Road Studios**.

1. **React Native:**

Developed and maintained by Facebook, react native is an open-source, cross-platform app development framework that has rapidly become the developer's top choice. It helps to develop Android and iOS mobile apps.

Mobile apps of top brands such as Tesla, Airbnb, Skype, and Amazon Prime are the best examples of React Native apps

1. **Features:**

* Faster development and deployment are the primary reasons behind the universal popularity of React Native.
* Reusable components, integration with third-party plugins, and component-based GUI development for front-end applications.

1. **Ionic**

This is an open-source framework built with Apache Cordova (Phone Gap) and Angular, allowing developers to build Android and iOS apps

The framework helps the developers build robust and feature-rich native applications.

An advantage of this framework is that it allows the developers to use several UI elements

Such as filters, forms, views; navigation menus and action sheets, in the application design.

This is a complete framework that enables developers to build Progressive Web Apps (PWAs)

and hybrid and cross-platform mobile applications.

1. **Xamarin**

It is a Microsoft-owned open-source framework for developing native and high performance Android, iOS, macOS, tvOS and watchOS apps leveraging.

1. **Features:**

* The framework empowers business to provide native performance and unique user experiences to end-users.
* Allows code sharing on multi-platforms in xamarin, an abstraction layer controls the communication between the underlying platform and shared code.
* the framework allows the developers to build an ecosystem with APIs, back-end, components.

1. **Native iOS(Swift):**

This is a programming language developed by Apple and it is used to build iOS apps build time is short, an extensive selection of debuffing tools in a dedicated XCTest testing framework as well as small size of apps and super-fast performance, both of which are common traits for all native development frameworks.

Apps build wih swift are **Uber, VSCO cam, yahoo Weather.**

1. **Features:**

Leading companies prefer Native Scripts because it has features like:

* appropriate backend support,
* business support
* complete access to iOS APIs and Android,
* cross-platform application
* a native user interface without web views.

1. **Apache cordova:**

Formally known as PhoneGap, AC is a developer-friendly mobile app development framework.

It is a cross-platform app development framework that leverages CSS3, HTML5 and JavaScript to build mobile applications. Cordova plugins enables the developers to access device hardware capabilities such as GPS, Camera and accelerometer to deliver a native-like app experience features are fast development process with a single codebase and third-party app management tools.

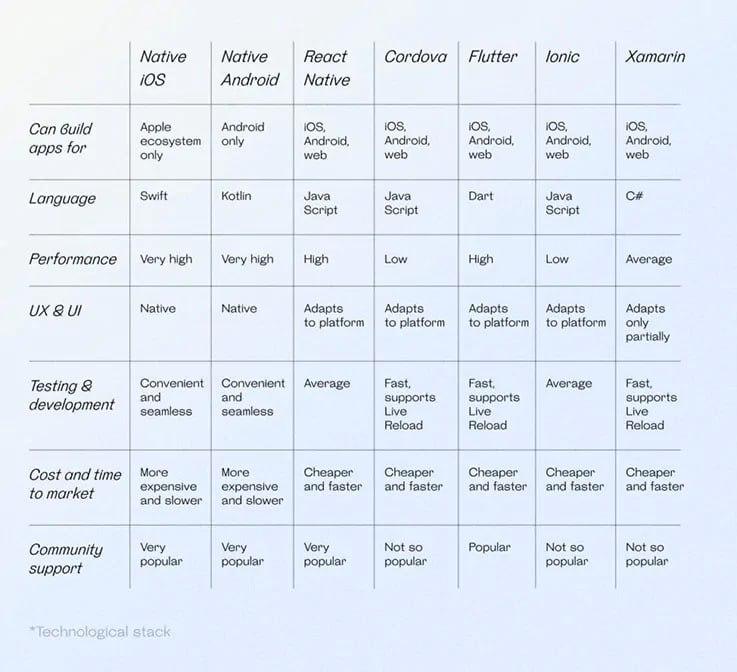
1. **Native Android SDK (Kotlin)**

It was first introduced by google in 2009 and was updated in 2017 and became the main language for building native Android apps. Android SDK allow developers to write lower-level-code, which make it easier to control the environment, build complex features and debug apps.

apps built with native frameworks do not require an interlayer of code to access device hardware

such as camera and microphone.

We can summarize all of this in the following table:



IV- **How to collect and analyze the requirements needed for a mobile app development:**

**The Software requirements analysis** is the process of identifying, defining and documenting the requirements of a software system. Categories of Requirements:

* **Functional Requirements**: function the software solution is required to perform.
* **Technical Requirements**: technical issues to take in consideration for the software solution to be well implements.
* **Transitional Requirements**: implement in case an extra product is needed.
* **Operational Requirements**: operations to be carried out in the backend for proper functioning of the system.

1. **Objectives of a Software Requirements Analysis**

The main objective is to identify the requirements of users and transform them into specific requirement that are measurable, realizable and contribute in the generation of income.

1. **Requirement Analysis Process**

* Identification and definition of the problem and the Stakeholders and end users
* Collections of requirement form the stakeholder/end-user and other that can be affected by our solution
* Solve ambiguity and incoherent problem of requirements
* Validation of requirements, which consist of testing and evaluating the requirements to check if the system satisfies the expectations.
* Organizing the requirement.
* And make the final document that describes what our solution is going to do (software documentation).

And then, the use of UML (Unified Modeling Language) diagram to illustrate and analyze graphically the requirements and show out the relationship between the actors, system and external system that interact with the software app.

1. **Requirement gathering techniques**

* **Interview**: consist of meet or contact potential user and organizations that will need our software and listen to what they are in need of.
* **Surveys**: consist of sharing a list of question to stakeholders to know what they need.
* **Workshops:** consist of regrouping the end users and organizations to discuss about the requirements.
* **Prototyping**: consist of releasing a prototype of the system and look at the reactions of the people and adjust/improve the requirement from their views of thing.
* **Use case analysis**: consist to identify the different actors/users that will interact with the software system and the different task and scenarios that will be executed which leads to adding some requirements for the new system or to validate the requirements for existing systems.
* Compare the actual system with the expected system and analyze what functionality is lacking.

1. **Requirement analysis techniques**

* **Business Process Model and Notation** (BPMN), used to a graphical view to show out the business process by managing the activities process between the actors.
* **Flowcharts:** used to describe the sequence flow and control logic of a related set of activities.
* **Gap analysis**: used to determine the present state of a product by evaluating the gap’s in a product’s performance in order to check if the requirements meet the expectations.
* **Gantt charts**: represent activity/task according to the scheduled timelines.

**V-** **HOW TO ESTIMATE A MOBILE APP’S DEVELOPMENT COST**

The price of building an app depends highly on its functionality and the wages of those who are going to build it. Every feature takes a number of hours to program, and the more complex the app, the higher will be the price of building and maintaining it after release. Below are some factors that influence the cost of a mobile application:

**-Features**

**-Platforms**

**-Design**

**-App Maintenance**

**-Team development size**

**-Team location**

1. **Features**

Every feature of an app has a more or less fixed price, which depends on the number of hours it takes to program it. The more complex the feature, the more expensive it is going to be.

Here is some example of features an app might demand:

• **Augmented reality**

**• Artificial intelligence**

**• Machine learning**

1. **Platform**

The number of platforms you want your app to be available on also influences the development cost, especially with native development, since iOS and Android apps are completely different products made by independent developers or teams

1. **Design**

The design also affects the application’s cost because as the complexity of the design grows, the application’s cost grows also.

For example we might want a system with animations and beautiful transitions; therefore the developers will spend more time working on it.

1. **Application maintenance**

Every app requires updates to maintain complete compatibility with new devices and Operating Systems, as well as fixing bugs. It also influences the app development cost.

Application maintenance includes:

- Operating system updates

- Library, framework, and tool updates.

Also we will have to rent a server since our app needs to store data.

1. **Team’s size and location**

How many people develop your app should be determined according to the project complexity. The more complex the project will be, the more you will need workers and the pricier the development cost will be.

The geographical location of the team plays an important role in the development costs. For example, Africa and India are the two cheapest markets

VI- **CONCLUSION:**

In summary, there are basically three categories of mobile apps which are Native apps, **web-based apps** and **hybrid apps**. Also there are several mobile programing language and mobile app development frameworks such as Flutter, React native, Ionic, JavaScript, swift, xamarin, etc… In order to develop a good mobile application, we have to know and analyze its requirements