

Mobile Apps Development

1.0



Table des matières

I - Chapter1 : Introduction to Mobile App Development	3
1. Mobile App definition	3
2. Mobile operating systems	4
3. Mobile OS market share worldwide in 2024	5
4. Mobile development constraints	6
5. How do mobile apps work?	7
6. Mobile app development approaches.....	8
6.1. Native approach	9
6.2. Cross-platform approach.....	9
6.3. Web approach	10
6.4. Progressive web approach.....	11
6.5. Hybrid approach	12
7. Development Frameworks & Tools	13
8. Conclusion	14
9. Appendix	14
10. References.....	14
11. Exercice : Evaluation test	15
Glossaire	16
Abréviations	17
Bibliographie	18
Webographie	19

Chapter1 : Introduction to Mobile App Development

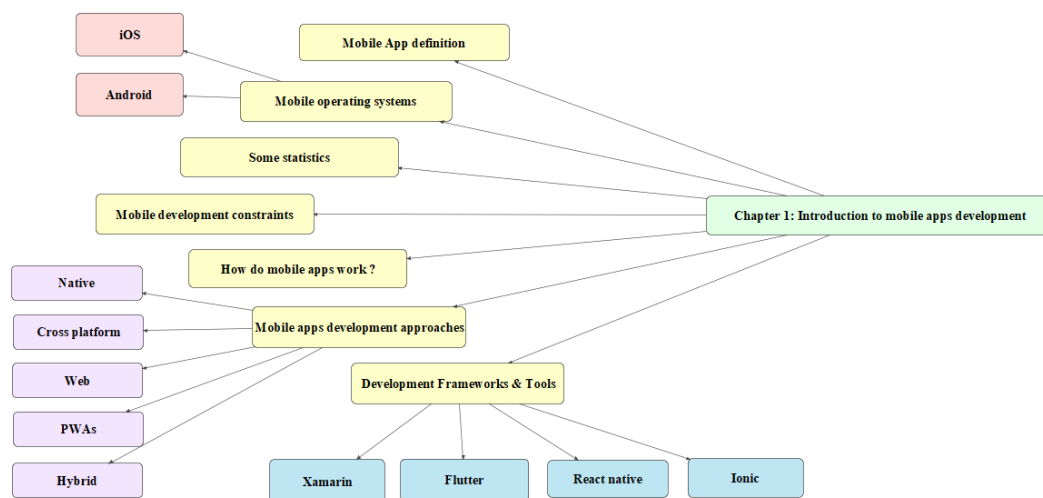


Objectives : this chapter aims to:

- Define what a mobile app is.
- List common mobile OS.
- Identify the mobile development constraints.
- Explain how mobile apps* work.
- Discuss the approaches, tools and frameworks used for mobile app development.

Plan :

1. Mobile App definition.
2. Mobile operating systems.
3. Some statistics.
4. Mobile development constraints.
5. How do mobile apps work ?
6. Mobile apps development approaches.
7. Development Frameworks & Tools.



1. Mobile App definition



- A mobile application (app) is an autonomous software application developed specifically for run on mobile device such as smartphones and tablets rather than desktop or laptop computers.
- Offers access to services in Web or Mobile version.
- Available on distribution platforms (Stores): **App Store, Google Play, Windows Phone Store.**

Examples of mobile apps:



Challenges :

🔍 Remarque

- **High performances:** running application without crashes or bugs and consuming little space in the device without affecting battery life.
- **Profitability:** the app's target audience, its monetization strategy, market competition and the overall effectiveness of its marketing efforts.
- **Scalability:** ability to handle increased workload or user demand => frequent updates.

2. Mobile operating systems

🔑 Définition

- Mobile OS* is an OS that helps to run other application software on mobile devices.
- It focuses on the management of :
 - Wireless connectivity,
 - Internet navigation,
 - Interface types,
 - Phone calls,
 - Embedded memory,
 - Process ...

Existing mobile OS:

iOS: was developed by Apple for its mobile devices such as iPhone, iPad, iPod.

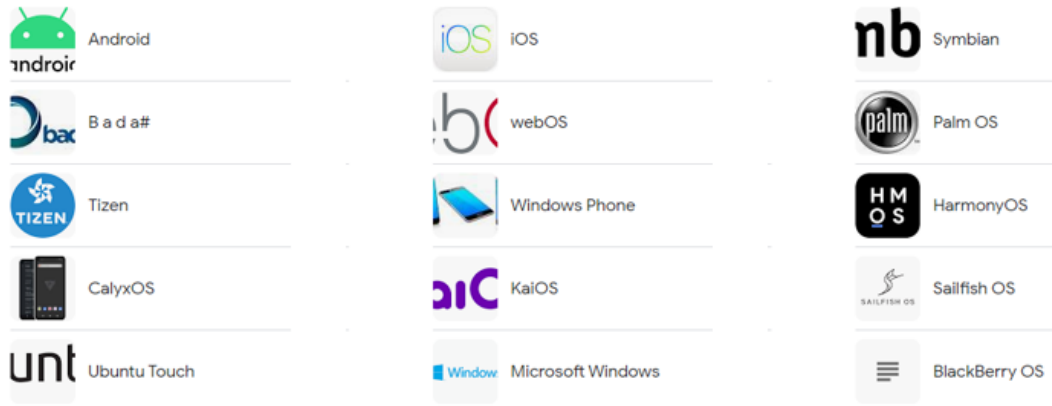
Windows Phone: was developed by Microsoft. It's not an open-source mobile OS.

Android: open-source based on Linux OS. It was developed by Android and re-purchased by Google in August 2005.

Symbian OS (Nokia), Blackberry OS (RIM), Bada (Samsung): less widespread on the market.

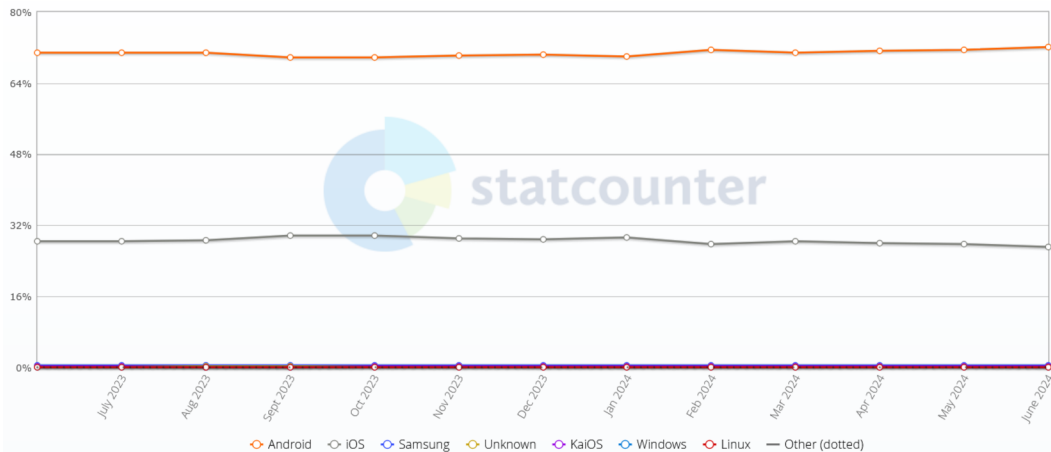
*Mobile OS market share in 2024 ***





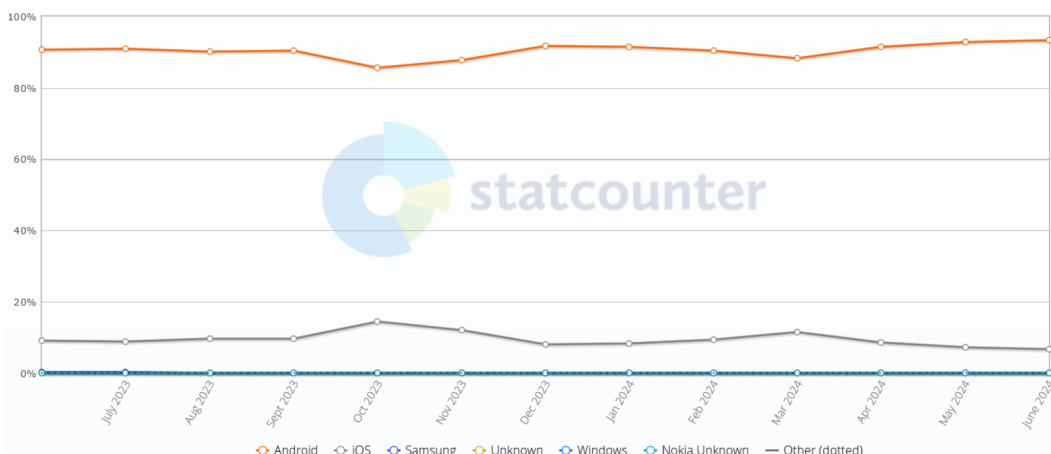
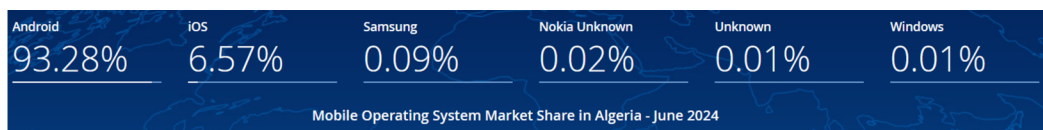
3. Mobile OS market share worldwide in 2024

In the world *






Worldwide market share

In Algeria *



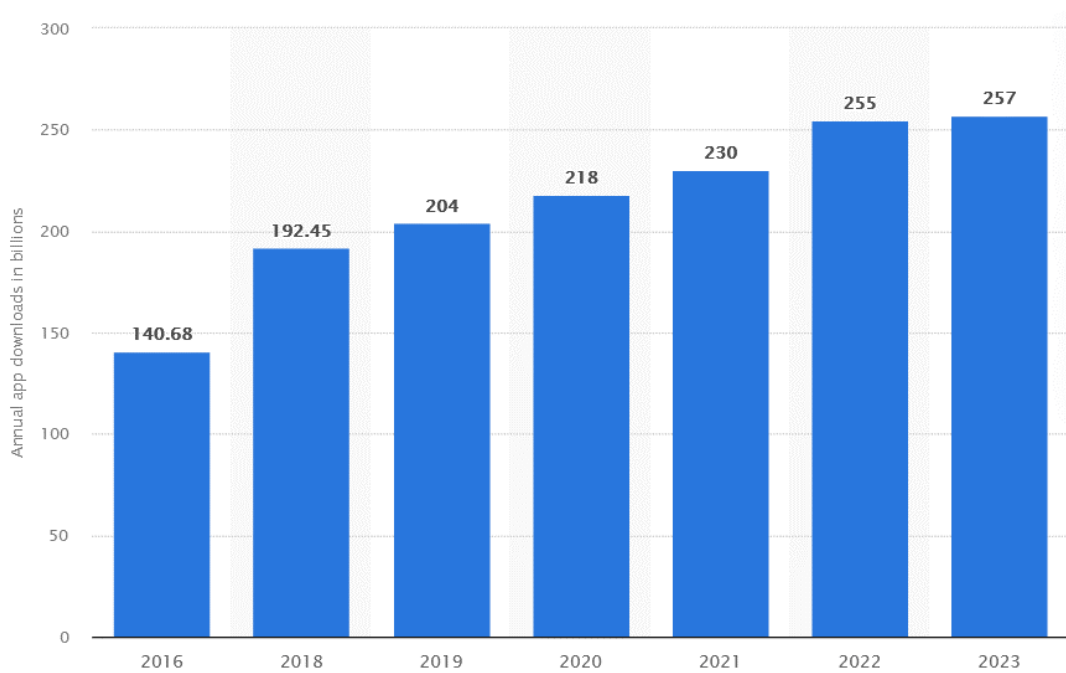
Algerian market share

By platform :

	Tablet 		Smartphone 		Desktop 
World	iOS 51.21%	Android 48.66%	Android 72.37%	iOS 26.98%	Windows 75.34%
France	iOS 55.9%	Android 43.75%	Android 62.12%	iOS 37.29%	Windows 74.71%

By platform

Number of mobile apps downloads worldwide 2016-2023 *



Number of mobile apps downloads worldwide 2016-2023

4. Mobile development constraints

- **Features of mobile devices:** limited memory, limited processing power, small screen size, different display resolution, wireless connectivity, limited bandwidth, heterogeneity of OS and technologies ...
- **The market concept:** reduce the Time To Market (TTM), the period of time from mobile app conception to it being available for purchase or use.
- **The platform choice:** knowing benefits and limitations of each platform with respect to application need and objectives.

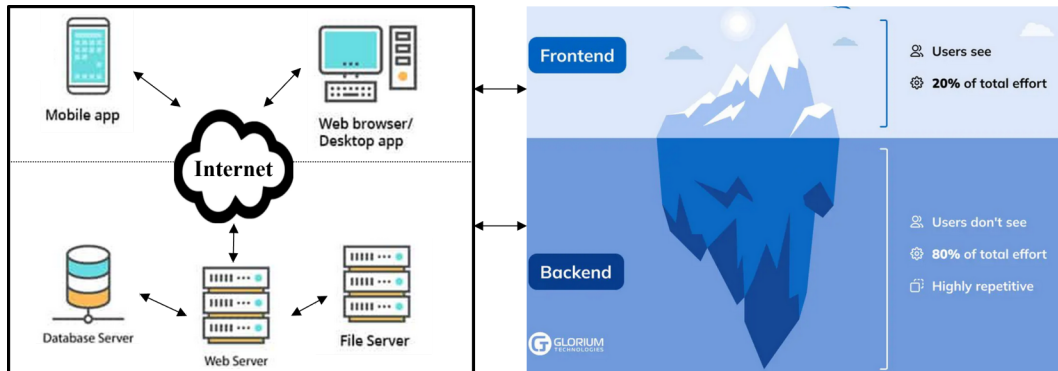
More constraints :

- Who is the target audience for the mobile application?
- What will it be used for?
- By which mobile device and on which operating system?
- Does the application need to integrate native functionalities ? (e.g. GPS, Camera, ...).

5. How do mobile apps work?

Mobile apps consist of two parts:

1. **Frontend (client side):** is where users interact with the software.
2. **Backend (server side):** is responsible for the application's logic and data handling. It acts as the brain of the app.

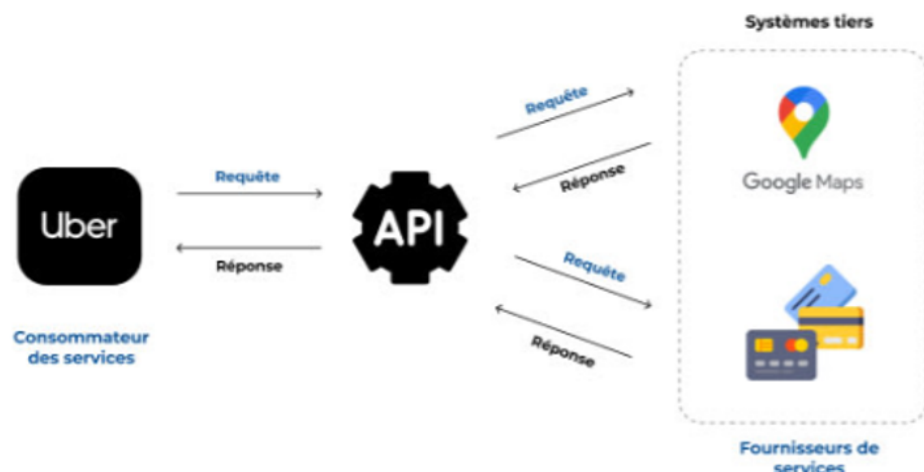
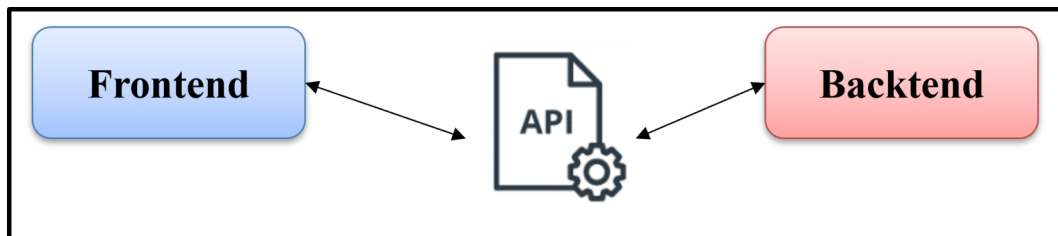


Mobile backend is an app designed for communication among machines and servers, it runs on a remote server without graphical interface.

Mobile backend takes care of:

- Data processing and storage independent of a smartphone's capabilities.
- Data sync and sharing across multiple devices.
- Content updates within the mobile app.
- Authorization and authentication that control access to data.
- Heavy processing operations.

Developers access to mobile backend via **APIs** (*Application Programming Interface*). **



API is an interface that refers to a set of protocols, rules and tools that allows different software applications to interact and communicate with each other.

Types of backend:

1. Third-party SaaS : *

- Is available solution on the market i.e. ready-made, plug-and-play solution.
- You plug in your frontend, subscribe to their plan, and release your app.
- Exp: AccuWeather API*, Shopify

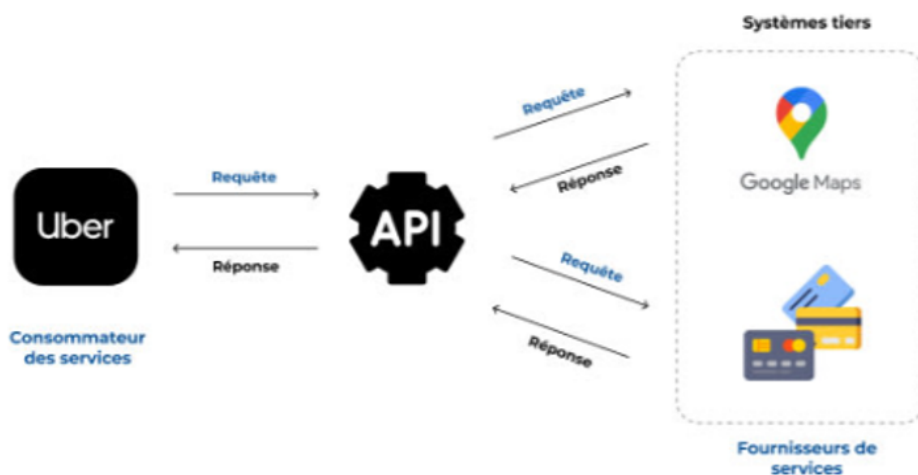
2. MBaaS Solutions : *

- Gives a basic structure that can mold based on a specific requirements.
- Allows the use of pre-developed components with a backend to offer capabilities to store user data, login/ registration features, authentication, and push notifications.
- Exp: Firebase

3. Custom Backend :

- Designed to meet the unique requirements of a particular application.
- Full control, flexibility, performance, security, scalability.
- Zero dependencies and min API* connections.
- Exp: Node.js, Ruby on Rails

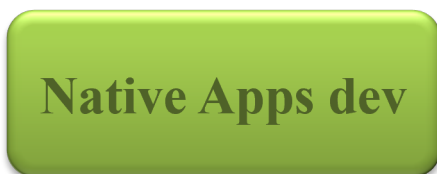
Example of SaaS backend :



6. Mobile app development approaches

Mobile app development approaches include both :

- The technologies used to build mobile apps.



- The solutions that make other apps work well on mobile devices.



6.1. Native approach

Native apps :



Définition

- Native development means creating a mobile app that is tailored and dedicated to a specified platform like iOS, or Android. It requires using the native language of the mobile device e.g:
 - Java or Kotlin for Android*,
 - Objective-C or Swift for iOS*,
 - C# for Windows Phone.
- Use the internal memory of the mobile device.
- Apps are available on stores dedicated to the target platform.

Characteristics :



Access to the OS native functionalities (contact, calendar, ...) and hardware resources (GPS, camera, sensor, Bluetooth, ...).



Provides high performance & fast execution as it's built specially for the OS (easy communication with the device).











Offers more control and customization over the User Interface (UI). Takes full advantages of the platform's design and features (Optimized UI* for the platform).



Uni-platform: generalization => duplicate efforts for each platform => time consuming & high cost.

Technology examples :

Operating systems	IDE	Language	Store	Open source
 Android	 Android Studio	Java / Kotlin	 Google Play	Oui
 iOS	 Xcode	Objective-C / Swift	 App Store	Non
 Windows Phone	 Microsoft Visual Studio	C#	 Windows Phone Store	Non

Comparison of mobile platforms technologies

6.2. Cross-platform approach

Cross-platform apps:



Définition

- Cross-platform development is an approach to develop apps that are compatible with multiple mobile OS (apps can be used regardless of the OS).
- From one code, it's possible to generate multiple apps for different OS.

Characteristics :



No need of specialized developers.



Faster and lower cost for development, maintenance and update (one code for multiple platforms).



Closer performance to native apps.



Access to device features through plugins or native code.



Wider reach (audience).



Reusable code.



UI design quality: can't take advantage of and use all the build-in features provided by mobile devices.



Update lag: app can't take advantage of new OS features immediately.



The need to add additional platform-specific work in order to handle a particular functionality.

Technology examples :

- React Native.
- Flutter.
- Ionic.
- Kotlin Multiplatform.

6.3. Web approach

Web apps:



Définition

- Web apps are websites that look and feel like apps on phones or computers. They can be accessed by the web browser of the mobile device.
- There is no need for a user to download the application from an app store.

Characteristics :

Cross-platform (run in web browsers).



No installation needed => require minimum device memory.



Easy to maintain.



Loading depends on internet speed.



No access to native functionalities (GPS,...).



Internet is essential.

Technology examples :

- HTML.
- CSS.
- JavaScript.

6.4. Progressive web approach**Progressive web apps (PWAs):****Définition**

- Progressive web apps (PWAs) *are web apps that aim to provide a user experience similar to that of native mobile apps. They leverage modern web technologies to deliver the same features as traditional mobile apps (exp: icon on home screen).

Characteristics :

Run on any platform through browser (mobile and desktop devices).



Act like a mobile app with offline availability, push notifications, access to device hardware (GPS,...).



Browser compatibility issues: not all PWAs features are supported by every browser.



Limited access to device hardware.



Performance limitation: they may struggle with handling large datasets or complex computation due to browser memory and processing limitations.



Managing offline states and synchronization can be complex.

Technology examples :

- Service Workers,
- Web App Manifests,
- IndexedDB,
- Frameworks (Angular, Ionic, React), ...

6.5. Hybrid approach

Hybrid apps:



Définition

- Hybrid apps mix web apps and native mobile apps. They are made with web technology but are put into a mobile app shell (i.e. they can be download from app stores).
- They can do some things that mobile apps do, like using camera or notifications.
- They are a middle ground, not as perfect as mobile apps but better than just web apps.

Characteristics :



Quick and simple development.



Single codebase for all platforms => Reduces maintenance costs and simplifies updates .



Access to phone's features (GPS,...).



Run slower and may have performance issues.



Appearance might vary across different devices.

Technology examples :

- Cordova.
- Ionic.
- React Native.
- Flutter.

7. Development Frameworks & Tools



Définition

- A **framework** is a structured set of components and tools that provide a foundation for developing software applications.
- It facilitates the development process by offering pre-built functionalities and development tools, allowing developers to focus more on the specific features of their application rather than on basic underlying infrastructure.



Flutter

Flutter: multi-platforms, a software development kit for mobile apps. Created by Google in 2013 and published in 2017. It's based on DART programming language.

Example :



ionic: multi-platforms, open source created in 2013. A technology based on AngularJS, Cordova et Node JS.

Example :



React Native: open source, multi-platforms for JavaScript. Created by Facebook in 2015. It's based on ReactJS.

Example :



Xamarin : multi-platforms. It's acquired by Microsoft and based on C# language.

Example :



8. Conclusion

- Together, the **frontend** and **backend** work seamlessly via APIs to provide a functional and responsive mobile app, ensuring a smooth user experience while efficiently managing data and resources on the server side.
- There are many mobile app development approaches (**native, hybride, cross-plateform, ...**), each one offers advantages and disadvantages.
- The choice of a mobile app development approach depends on some constraints such as the target audience, plateform, required fonctionnalities, ...

9. Appendix

Pros	Cons	
<ol style="list-style-type: none"> 1. Low cost of backend development/integration. 2. No need to have a dedicated backend development team. 3. Shorter time-to-market. Can be deployed in a few hours. 	<ol style="list-style-type: none"> 1. Lack of flexibility in building and launching custom features. 2. Monthly limits/restrictions depending on the plan. 3. Limited scalability potential as the user base grows. 	MBaaS backend
Pros	Cons	
<ol style="list-style-type: none"> 1. More flexible and customizable than SaaS backend solutions. 2. Tiered cost structure based on usage. Pay only for the resources you utilize. 3. Ready-to-use components and pre-programmed libraries for embedding features and capabilities within the app. 	<ol style="list-style-type: none"> 1. Backend service outage can affect your app users. 2. Costs more in case of highly-scalable apps that attract a huge amount of traffic in the long run. 	MBaaS backend
Pros	Cons	
<ol style="list-style-type: none"> 1. Highly flexible and adjustable as per the needs of the project. 2. Unlimited scalability and reliability. 3. Ultimate security and data privacy – no dependency on third-party integrations. 	<ol style="list-style-type: none"> 1. Costly. 2. Takes a lot of time and specialized talent to develop and manage. 	Custom backend

10. References

- To learn more about Prograssive Web Apps, please feel free to visit : *
- To learn more about Mobile App Approaches, please feel free to visit : * and *
- To learn more about Mobile App Bachend , please feel free to visit : *

11. Exercise : Evaluation test

Q1

Which programming language can be used to develop Android mobile app ?

- Java/kotlin
- Swift
- C#
- Dart

Q2

List two constraints related to the development of Mobile Apps

Q3

In Native development the application is dedicated to a specific platform.

- True
- False

Q4

What is Flutter ?

- A data base management system.
- A framework to develop only web applications.
- A software development kit (SDK) to create mobile, desktop, and web apps.
- A programming language to develop mobile apps.

Q5

What is the main advantage of using Flutter for app development?

- Uses multiple programming languages.
- Creates applications for iOS.
- Provides a native experience on different platforms with a single codebase.
- Is free and open source

Glossaire



Android

the mobile OS developed by Google, designed for touchscreen mobile devices such as smartphones and tablets.

API (Application Programming Interface)

is a set of rules and protocols that allows different software applications to communicate and interact with each other.

iOS

the mobile OS developed by Apple exclusively for its hardware including iPhone, iPad, and iPod Touch.

Abréviations



API : Application Programming Interface

Apps : Applications

MBaaS : Management of Business Application As a Service

OS : Operating System

PWAs : Prograssive Web Apps

SaaS : Software As A Service

UI : User Interface

Bibliographie



Mike Katz et al. Flutter Apprentice Learn to Build Cross-Platform Apps, 2nd Edition, 2021.

Thomas Bailey , Alessandro Biessek, et al. Flutter for Beginners: Cross-platform mobile development from Hello, World! to app release with Flutter 3.10+ and Dart 3.x, Kindle, 2023.

Sanjib Sinha. Beginning Flutter with Dart: A Beginner to Pro. Learn how to build Advanced Flutter Apps (Flutter, Dart and Algorithm Book 1), Kindle Edition, 2021.

Mark Clow. Learn Google Flutter Fast: 65 Example Apps, Paperback, 2019.

Dieter Meiller. Modern App Development with Dart and Flutter 2. 2021

Webographie



<https://gs.statcounter.com/os-market-share/mobile/worldwide>

<https://gs.statcounter.com/os-market-share/mobile/algeria>

<https://flutter.dev/>

<https://www.statista.com/statistics/271644/worldwide-free-and-paid-mobile-app-store-downloads/>

<https://www.similarweb.com/platforms/algeria/>

<https://web.dev/learn/pwa>

https://www.thedroidsonroids.com/blog/what-is-a-mobile-app#What_is_a_mobile_application

<https://www.spaceotechnologies.com/glossary/mobile-app/what-is-mobile-app/>

<https://www.nomtek.com/blog/mobile-app-backend>