RESEARCH AND ANALYSIS OF THE MOBILE DEVELOPMENT ECOSYSTEM IN THE CONTEXT OF THE COURSE INTERNET AND MOBILE PROGRAMMING, CEF440

NAME	MATRICULE NO
YIMNAI NERUS ZAUMU	FE20A123
Tabot Charles Bessong	FE20A106
Bebongnchu Yannick Nkwetta	FE20A022
Balemba Junior Balemba	FE20A021
Tandongfor Shalom Changeh	FE20A111

1. List the major types of mobile applications and their differences.

There are three basic types of mobile apps if we categorise them by the technology used to build them.
Native apps are created for one specific platform or operating system. Web apps are responsive versions of websites that can work on any mobile device or OS because they're delivered using a web browser which is today used by android,iOS,Mac,Windows and Linux systems. Hybrid apps are a combination of native and web apps. Hybrid applications are built by combining both native and web development technologies. The core application code is written using programming languages such as JavaScript, Dart. The code is then wrapped within a native application using open-source frameworks such as Ionic or React Native, Flutter.

I. **Native apps** are developed for specific platforms or operating systems like Android, iOS, or Windows Phone. They are written in languages that are native to the platform, such as Java or Kotlin for Android and Objective-C or Swift for iOS. Native apps are installed directly onto the device and can access all of the device's hardware features, such as the camera, microphone, and GPS since there is no layer of abstraction between the OS and the programming language used to develop the application. This ability to communicate directly with the OS gives them high performance advantages,however can make the process of developing the skills to build such applications hard for inexperienced engineers. To use the application, you will need to download it on platforms such as apple store for iOS and play store for Android.

II. **Web apps** are mobile-optimised web pages that look like an app. They are delivered using a mobile browser and can run on various internet browsers like Chrome or Safari. Web apps are written using Programming languages like HTML,CSS,Javascript,PHP.. The most popular language used for web apps is Javascript which has frameworks like React JS, Svelte , Vue , Angular just to name a few. Most web applications today require a backend which are built using frameworks like programming languages like JavaScript,PHP,Go and Python. Databases such as MySQL,MongoDB and firebase are used for storing users' information on the web app. Generally a web app can have a frontend and a backend. The frontend is the user interface which help the user interact with the web app. The backend is use to communicate the user's request from the frontend to the server. For a web app to operate, it needs a web server, application server and database. Web servers manage the requests that come from a client, while the application server(backend) completes the requested task. A database stores any necessary information

III. **Hybrid apps** are a combination of native and web apps. They are built using web technologies like HTML, CSS, and JavaScript and then wrapped in a native container that can access native platform features. Hybrid apps can be developed across all platforms[1]. They can work on multiple platforms like on Android and iOS. Examples of technologies used for this are React Native, Flutter,iOnic.

Native apps tend to be faster and more reliable when it comes to user experience because they are developed specifically for a mobile system. Web apps have a lower barrier to entry compared to native apps due to their comparative simplicity and thus tend to be cheaper and easier to develop. Hybrid apps use app-embedded browsers to render HTML web pages, which makes them slower than native apps but faster than web apps.

Parameters	**Native**	**Web**	**Hybrid**
Performance	✦✦✦✦✦ Most robust performance.	✦✦✦ Runs code on webview, which executes slower due to the abstraction.	✦✦✦✦ Performance comparable to Native, due to native execution of code.

Hardware features access	✦✦✦✦✦ Fastest as there is no abstraction.	✦✦✦ Slower and less efficient access due to layers of abstraction.	✦✦✦✦ All APIs available to Native will be available, so fast aswell.
Debugging and Profiling Tools	✦✦✦✦✦ Excellent tools for debugging platform and logic issues.	✦✦ Mostly web-development based tools are used to debug and profile; which might not bring out platform related issues easily.	✦✦✦✦ Solid tools will be available in the common language framework for regular debugging.
Dev Support Ecosystem	✦✦✦✦✦ Excellent Support.	✦✦✦ Less size of community around the platform.	✦✦✦✦ Solid support from the framework maintainers and the community.
Development Cost	✦✦✦ More Expensive	✦✦✦✦ Mostly reusable code hence less expensive.	✦✦✦ Slightly more expensive than native.
Development Time	✦✦✦✦ Will have to write separate code for each platform, with less frameworks for development.	✦✦✦✦ Most code works accross platforms.	✦✦✦✦ Most code works across platforms.
Ease of finding Devs	✦✦✦✦✦ Developers are most widely available.	✦✦✦ Technology(HTML,CSS,JS) is easy to find but actual responsive web developers are slightly hard to find.	✦✦✦ Devs are usually hard to find.
Ease to deploy to multiple platform	✦✦ Will need separate devs and written from scratch per OS.	✦✦✦✦ Most functionality works on other platforms.	✦✦✦✦ Most functionality works on other platforms.
Build Size	✦✦✦✦ Usually the leanest possible	✦✦✦ Builds are slightly bloated with multiple	✦✦✦ Usually bloated compared to a native build.

	code and resources	files. Usually files to support other platforms get included in the final build.	
Programming Languages	Java and Kotlin for Android, Swift and Objective-C for iOS.	HTML, CSS, Javascript , PHP aand many other web development languages are used.	Mostly frameworks like Flutter, React Native, Ionic, Xamarin.

Other than these three basic types of mobile apps, a few more need to be mentioned.

• Progressive Web Apps

A Progressive Web App (PWA) avoids the need for download, so offers a flexible alternative to standard Web Apps. It is a lightweight app that runs on the URL of a device's web browser. It looks and feels like a mobile app, but it's not delivered natively on the device. Instead, it runs inside an app browser. Doing so enables PWAs to match or mimic the user experience of native apps by being smaller, more agile, and faster-running web apps. Common technologies used to run PWAs include HTML, CSS, JavaScript, and WebAssembly.

These are extensions of websites that you can save on your devices that work like apps. PWAs use web browsing APIs and have the functionalities of native apps. These web pages are added to your devices to mimic the web application. They also run faster irrespective of the device type or Operating System used.

Pros:
Progressive Web Apps use very little data and are automatically updated when you use them. There is no requirement for installation as PWAs are web pages. Another added advantage is that they can be easily shared by sending the URL.

Cons:

The main disadvantage or con of using PWAs is that they are limited to the operating system in which they are used. Also, these apps can have integration problems. Also, PWAs are limited regarding key re-engagement features like adding to the home screen, updating notifications, etc.

A solid example of a PWA is MS Teams. It's developed as a PWA on an Electron framework from GitHub, which combines the Chromium rendering engine and Node.js JavaScript platform. And it benefits from being a PWA through its dynamic approach to video conferencing, messaging, and integration with Office 365 products such as Word, Excel, and Powerpoint. So, MS Teams as a progressive web app enables complete integration between desktop and app but enables a native experience for the user in all instances.

• Cross-Platform Apps

These apps were created to be compatible with multiple operating systems and run on desktops, tablets, mobile phones, smartwatches, and even smart TVs.

The single biggest challenge of the native approach is unreusable code. Developers need to create another set of code to fit another operating system, to replicate the app. But there is a way around this, and it supports different types of mobile apps, in the form of cross-platform development. And one of the more popular cross-platform-based frameworks is React Native, which is an open-source UI framework that offers a high level of flexibility. React Native uses a Javascript library. And, it sits alongside other cross-platform frameworks such as Flutter, Ionic, and Xamarin as a native app-building alternative that enables cross-platform development. In practice, code used to develop an app for a specific OS, for example, Android, becomes reusable for iOS and Windows phones.

Pros:
These applications have over 90% reusable codes that are easy to maintain and update. Also, they have a much broader reach than other applications. Since cross-platform apps run on any operating system, they are exposed to many users.

Cons:

Due to the abstract nature of these apps, the codes are hard to write. Developers who understand this cross-platform functionality and tools can only write these codes.

2. Do a thorough review of the programming languages used for mobile programming.

As seen above, there are three types of mobile applications and the different types have different programming languages and different technologies used for their development.

Native Applications:

Android apps for example use Java and Kotlin and iOS apps are built with Swift and Objective-C

The main programming languages used for native app development are Objective-C and Swift for iOS, Java for Android, and C# for Windows Phone. Objective-C is an object-oriented programming language that is used to write software for Apple's iOS and macOS operating systems. Swift is a newer programming language that was developed by Apple to replace Objective-C. It is designed to work with Apple's Cocoa and Cocoa Touch frameworks. Cocoa Touch framework is the application development environment for iOS. It includes the Foundation and UIKit frameworks. UIKit includes classes for event handling, drawing, image-handling, text processing, typography, and interapplication data transfer. It also includes user-interface elements such as table views, sliders, buttons, text fields, and alert dialogs

Java is an object-oriented programming language that is used to write software for Android devices. Java is today owned by Oracle after they purchased the app from a startup. Kotlin is another programming language that can be used to develop Android apps. It was developed by JetBrains and is designed to be more concise, expressive, and safe than Java.Kotlin has several features that make it more precise and safe than Java. For example, Kotlin has proper function types and use-site variance without wildcards. Additionally, Kotlin does not have checked exceptions.

C# is a modern, object-oriented programming language that was developed by Microsoft. It is used to write software for Windows Phone devices. Xamarin is a cross-platform software development tool that can be used to develop native apps on iOS and Android using C#[2].

Web apps:

They are built with programming languages like **HTML CSS and Javascript** for the frontend and languages like **Javascript, php, python and Go** for the backend of the web application. Javascript and python for example have libraries and frameworks built on top of the language to ease the step of development with Javascript having frameworks like (**React,Angular,Node JS**) for frontend and backend and python having (**Flask and Django** for the backend)
Web applications are built using a combination of programming languages and technologies. The most commonly used programming languages for web development are **JavaScript, Python, Java, C++, C#, PHP, and Perl.** HTML (the markup language), CSS (the styling tool), and JavaScript (programming language) are the technologies used in front-end development[3]. HTML makes the skeleton of your website, CSS is used to style your website, and JavaScript is used to add interactivity to your website. Python is a general-purpose language that is mostly used for creating web applications. Ruby is another general-purpose language that is mostly used for creating web applications, to the best of my knowledge I now ruby on the rails is use mainly for the backend. There are several frameworks that can be used to build web applications, including Angular, React, Vue.js, Ruby on Rails, Django, Flask, and Express.js[3].

Hybrid applications:

They are built using web technologies such as **HTML, CSS, and JavaScript**, and then encapsulated into a container called a webview. There are several frameworks that can be used to build hybrid applications, including React Native, Flutter, Ionic, jQuery Mobile, and Appcelerator Titanium[1]. These frameworks allow developers to build hybrid mobile and desktop applications using a combination of native and web technologies[3]. React Native is the most popular framework for hybrid application development[1]. It allows developers to compile JavaScript into machine code to achieve native performance[2]. Flutter is another popular framework that is easy to use and implement[1]. Ionic is best used for mobile app development[1]. jQuery Mobile is a JavaScript framework that is fully dependent on the plugins available in JavaScript like Content Slider, Image Slider, Pop-up Boxes, etc.. Appcelerator Titanium is an open-source framework that allows developers to build native apps on iOS and Android using JavaScript[1].

3. Review the mobile applications frameworks and compare them .

1. Flutter

Flutter is Google's open-source software development framework for making native Android and iOS apps for mobile, web and desktop from a single codebase. It's free and open source.

It's renowned for enabling mobile app developers to create beautiful apps quickly. It's a complete framework that includes widgets, a rendering engine, and testing and integrating APIs to help developers build aesthetically pleasing mobile apps.

Key features:

- Heavily optimized, mobile-first 2D rendering engine with excellent support for text
- Modern react-style framework
- Rich set of widgets implementing Material Design and iOS-style
- APIs for unit and integration tests
- Interop and plugin APIs to connect to the system and 3rd-party SDKs
- Headless test runner for running tests on Windows, Linux, and Mac

- Dart DevTools for testing, debugging, and profiling your app
- Command-line tools for creating, building, testing, and compiling your apps
- Appealing visuals and a high-end user experience and interface
- Native-level performance

2. React Native

React Native is an open-source UI development framework created by Meta (formerly Facebook). It combines the best parts of native development with React, a best-in-class JavaScript library for building user interfaces. The libraries offer fast and consistent development experiences for both Android and iOS apps. Prominent businesses like Airbnb, Skype, and Amazon Prime are all built using the React Native framework.

The major appeal of React Native is quick development and implementation time.

Key features:

- Reusable elements
- Compatible with third-party plugins
- Single code base across all platforms
- Component-based GUI creation for front-end apps
- Declarative API for predictive UI
- NPM for installation
- Live reload

3. Xamarin

Xamarin is an alternative open-source platform for building mobile apps for iOS, Android and Windows that is Microsoft-owned. It’s developed using the C# language, which requires fewer lines of code, and helps developers build products faster.

Xamarin enables developers to share an average of 90% of their applications across platforms. It means developers can write all their business logic in a single language but achieve a native look, feel, and performance on each platform.

Key features:

- Complete binding for the underlying SDKs
- Objective-C, Java, C and C++ Interop
- Modern language constructs
- Robust Base Class Library (BCL)
- Modern Integrated Development Environment (IDE)
- Mobile cross-platform support
- Rapid advancement
- Native look and feel
- Compatible with lots of devices

4. Apache Cordova

Apache Cordova - previously PhoneGap - is a developer-friendly mobile app development framework. It allows you to use HTML5, CSS3, and JavaScript for cross-platform app development.

Apps execute within wrappers targeted to each platform, and rely on standards-compliant API bindings to access each device's sensors, data and network status.

The Cordova plugins enable developers to use smartphone hardware features like GPS and cameras to give users a native-like experience.

Key features:

- Quick prototype deployment
- Combines native app components with WebView
- Access device-level APIs
- Single source of code
- Command Line Interface (CLI)
- Plugins with smartphone hardware features
- Third-party application management tool

5. Ionic

Ionic is a complete open-source framework for building modern, high-quality cross-platform mobile apps from a single code base. Developers can create apps for iOS, Windows and Android.

Ionic allows developers to use several user interface elements such as filters, forms, views, and navigation menus in their app designs. It can also be used to create Progressive Web Apps (PWAs) and hybrid apps.

Key features:

- World-class documentation
- 100% open-source mobile SDK
- A CLI for building and testing
- Cross-platform native bridge
- React, Angular and Vue integrations
- Intuitive UI components

6. NativeScript

NativeScript is another open-source framework. This framework is best suited for developing with Typescript, Angular, JavaScript, CSS, and Vue.js. It helps speed up app load times and cuts down development time.

Businesses like to use NativeScripts because it offers comprehensive backend and business support as well as cross-platform development. NativeScript framework's APIs are similar to Xcode and Android Studio. It also allows you to develop Android and iOS apps from a single source code.

Key features:

- Native user interface without WebViews
- Full access to Android and iOS APIs
- Reusable elements
- Backend support
- Cross-platform application

7. JQuery Mobile

JQuery Mobile is an HTML-5-based UI framework designed to make cross-platform and responsive websites and apps. It’s built on the jQuery and jQuery UI foundation and offers Ajax navigation with page transitions, touch events, and various widgets. Its lightweight code is built with progressive enhancement and has a flexible, easily themeable design.

JQuery Mobile was one of the most popular mobile app development frameworks with customizable templates. Most jQuery-based plugins and scripts work on pages without requiring any changes to the HTML code.

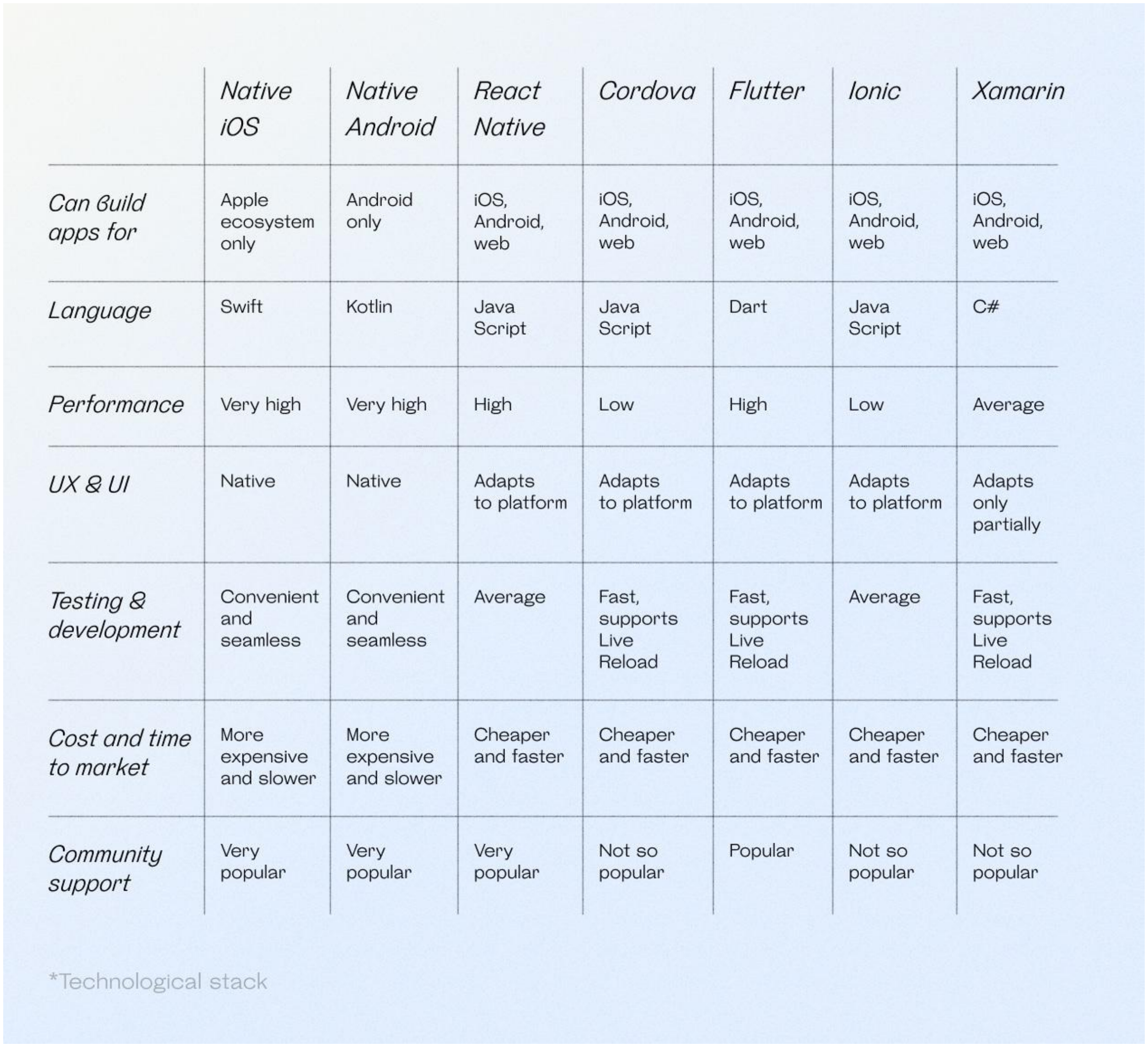
Be careful: Since the end of 2021, JQuery Mobile is no more supported. Read here for more information.

Key features:

- Easy and familiar jQuery syntax
- Built-in *FastClick* library
- Framework agnostic - can be used with React, Angular and others
- Rich ecosystem of templates, icons and plugins
- Out-of-the-box UI elements
- Provides support for Vue.js, React, and Svelte

8. Appcelerator Titanium:

Appcelerator Titanium is a great choice for professional developers because it uses JavaScript, which is one of the most popular programming languages in the world. It also has good performance and security. However, its community and support are not as large as some of the other frameworks on this list.



	Native iOS	Native Android	React Native	Cordova	Flutter	Ionic	Xamarin
Can Build apps for	Apple ecosystem only	Android only	iOS, Android, web	iOS, Android, web	iOS, Android, web	iOS, Android, web	iOS, Android, web
Language	Swift	Kotlin	Java Script	Java Script	Dart	Java Script	C#
Performance	Very high	Very high	High	Low	High	Low	Average
UX & UI	Native	Native	Adapts to platform	Adapts to platform	Adapts to platform	Adapts to platform	Adapts only partially
Testing & development	Convenient and seamless	Convenient and seamless	Average	Fast, supports Live Reload	Fast, supports Live Reload	Average	Fast, supports Live Reload
Cost and time to market	More expensive and slower	More expensive and slower	Cheaper and faster	Cheaper and faster	Cheaper and faster	Cheaper and faster	Cheaper and faster
Community support	Very popular	Very popular	Very popular	Not so popular	Popular	Not so popular	Not so popular

*Technological stack

4. How do you collect and analyse requirements for mobile applications you want to build(functional and non-functional requirements)?

The following considerations to include when mapping out business requirements:

- What is the purpose of the app or product? What are you trying to accomplish?
- What is the current problem(s) it will solve?

- How will it improve the current process? Will it facilitate a new process?
- What is the product vision statement?
- Will the app need to be started from scratch, or can you leverage existing assets?
- What should the app be able to do? What is the product’s core functionality?
- What features will it need?
- What is the monetization or business model?
- Are there branding and design guidelines to follow?
- Is the task feasible?

Collections

1. Identify the stakeholders: Identify all the stakeholders who will be involved in the development process, including the customer, end-users, developers, and project managers.
2. Gather information: Gather as much information as possible about the app you want to develop. This includes information about the customer's business, their target audience, and their goals for the app.
3. Define the scope: Define the scope of the project by identifying what features and functionality will be included in the app.
4. Prioritise requirements: Prioritise the requirements based on their importance to the customer and end-users.
5. Create use cases: Create use cases that describe how users will interact with the app.
6. Define non-functional requirements: Define non-functional requirements such as performance, security, and scalability.
7. Review and refine: Review and refine the requirements with the customer and other stakeholders to ensure that they are complete and accurate.
8. Document the requirements: Document the requirements in a clear and concise manner so that they can be easily understood by all stakeholders.

Analysis
Once you have collected the functional and nonfunctional requirements for an app you want to develop for a customer, you can begin the process of analysing them. Here are some steps you can follow:

1. Review the requirements: Review the requirements to ensure that they are complete and accurate.
2. Identify dependencies: Identify any dependencies between requirements and ensure that they are properly accounted for.
3. Identify conflicts: Identify any conflicts between requirements and resolve them.
4. Prioritise requirements: Prioritise the requirements based on their importance to the customer and end-users.
5. Create a traceability matrix: Create a traceability matrix that maps each requirement to a specific feature or functionality in the app.
6. Create use cases: Create use cases that describe how users will interact with the app.

7. Define non-functional requirements: Define non-functional requirements such as performance, security, and scalability.
8. Review and refine: Review and refine the requirements with the customer and other stakeholders to ensure that they are complete and accurate.

5. How do you estimate the cost of a mobile app?

The cost of developing a mobile app depends on various factors like the complexity of the app, the features and functionality required, the platform (iOS or Android) you want to target, the experience and location of the development team, and much more.

According to a survey by Clutch, the average cost of developing a mobile app is around $171,450, but this figure can range from $10,000 for a basic app to more than $500,000 for a more complex app with advanced features.

However, it's important to keep in mind that these are just averages and the actual cost of developing a mobile app for your business or project can vary depending on your specific needs and requirements. It's best to consult with a professional mobile app development company to get a more accurate estimate.

Some of the common factors that affect the cost of mobile app development:

- **Platform**: The cost of developing an app can vary depending on whether you want to target iOS, Android, or both. Developing an app for both platforms will generally be more expensive than developing one for just one platform. iOS developers tend to charge higher than Android app developers. On the other hand, it costs less to develop a cross-platform mobile app.
- **Complexity**: The complexity of the app will also impact the cost. Apps that require complex features such as machine learning, AI, or augmented reality will generally cost more to develop.
- **Cost breakdown**: The cost of developing an app will differ depending on the type of app. A simple app will cost the least an average app costs more while a complex app is the most expensive.
- **Skill, rate & experience**: The mobile app developer and development firm you hire play a vital role in the overall cost of the app.
- **Type of App:** There is the option of developing a simple web app that is a responsive app developed mainly using web technologies and costs the least. It is ideal for small businesses. A native mobile app takes advantage of the core features of the mobile device and OS. This type of app tends to be highly

feature-rich and also costs more. A hybrid app is one that is the best of both worlds, native and web apps. This is a good way to keep your costs low and still develop a good app. A cross-platform app is usually developed using a framework and since it uses a single codebase for both iOS and Android, it will also cost less.

- **Development team**: The location and experience of the development team will also play a role in the cost of app development. Developers in certain regions or countries may charge more or less than others.
- **Features**: Every mobile app is different with a different set of features like user authentication, GPS location, etc. The more features and customization the more the cost. The cost will rise as you integrate your app to use more hardware technologies like camera, Geofencing, Bluetooth, accelerometer, pedometer, altitude, etc.
- **Integration points** – will an app be integrated with third-party apps that will be the source of its content.
- **Use of visual objects** – complexity of visual objects inside of an app will significantly influence the cost.
- **Maintenance plan** – once an app development project is over, certainly it will require technical support from its developer to provide updates that patch bugs or introduce new features.

Primarily apps can be broken down into four major groups based on the amount of work involved in developing them. These choices of functionalities are essentially what determines the custom app development cost. Generally, gaming apps and enterprise apps tend to have more costs to get an app developed than simple and API apps.

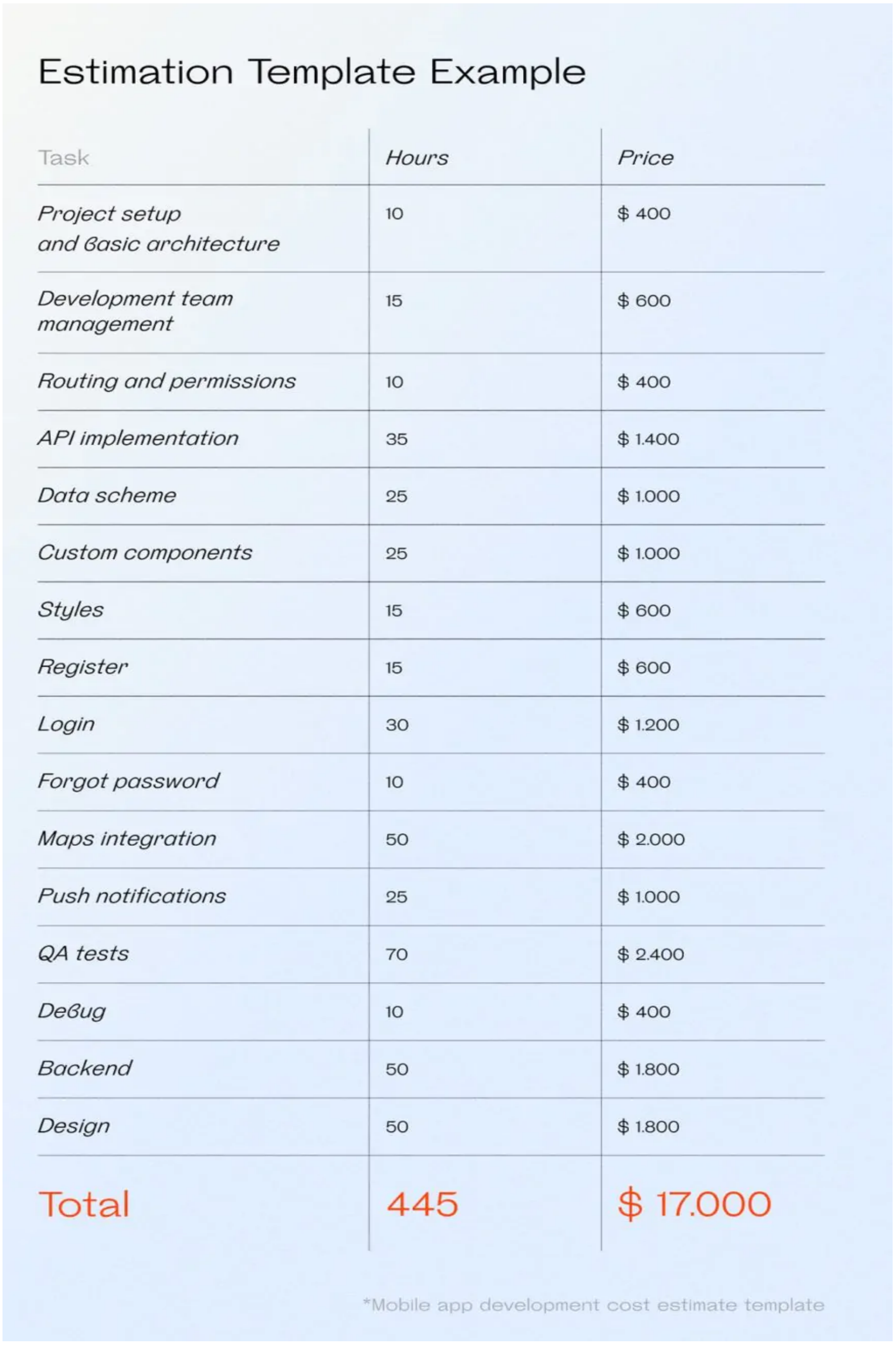
This is because enterprise and gaming mobile applications include many features, increasing the complexities, cost to build an application, and the time required for building the apps. Here’s a quick estimate of what is the average cost of developing an app.

- Simple Apps – $10,000 to $20,000
- API Apps – $35,000 to $70,000
- Enterprise Apps – $60,000- $1,00,000
- Gaming Apps – $1,00,000- $2,50,000

The figures provided are an elementary cost estimate for mobile application development. The final cost of mobile app development services may vary according to the features that the application requires.

They are always liable to change depending upon the scalability of the project. These are of course price ranges associated with getting an app built from a dev shop.

The cost estimate may vary if you opt to get your mobile app developed by a freelancer, or rather an offshore managed resource. However, make it a point to never compensate for the app quality.



Estimation Template Example

Task	*Hours*	*Price*
Project setup and basic architecture	10	$ 400
Development team management	15	$ 600
Routing and permissions	10	$ 400
API implementation	35	$ 1.400
Data scheme	25	$ 1.000
Custom components	25	$ 1.000
Styles	15	$ 600
Register	15	$ 600
Login	30	$ 1.200
Forgot password	10	$ 400
Maps integration	50	$ 2.000
Push notifications	25	$ 1.000
QA tests	70	$ 2.400
Debug	10	$ 400
Backend	50	$ 1.800
Design	50	$ 1.800
Total	445	$ 17.000

*Mobile app development cost estimate template

To conclude, mobile app estimation is a process that takes into account all aspects of the future project and is based on a detailed technical specification. The process is conducted in several stages, from the project’s breakdown by screens or by features, to the estimation by developers and then by a project manager. Although the development team may provide to the client a precise cost estimate for app development and work on the fixed price model, a more rough and top-level estimation together with the time & material price model bears fewer risks for both parties and is more flexible to implement necessary changes when the development has already started.