CEF440:

INTERNET AND MOBILE PROGRAMMING

Presented by:

HEUYA NDJANSEB KEVIN KENNEDY   **FE20A049**

INSTRUCTOR: Dr. NKEMENI VALERY

March 2023

CONTENT:

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) Review mobile application development frameworks by comparing them using key features

5) Collect and analyze the requirement for a mobile app

6) Estimate the mobile app development cost

1. Different type of mobile app that exist

Mobile applications, or "apps," have become an integral part of our daily lives. From social media and gaming to productivity and education, there is an app for almost everything. These apps can be divided into several major types based on their functionality and purpose. In this article, we will explore the major types of mobile applications and their differences in detail.

1. Native Apps

Native apps are mobile applications designed for a specific platform, such as iOS or Android. They are built using the platform's native programming language and are optimized for that particular operating system. Native apps can access and use the device's hardware and software resources, such as camera, GPS, contacts, and notification system.

Native apps are highly responsive and offer a seamless user experience. They can take advantage of the latest features of the operating system and offer faster performance compared to other types of apps. Native apps are also more secure as they are downloaded from official app stores and go through a rigorous approval process.

However, developing native apps can be time-consuming and expensive. Developers need to create separate versions of the app for each platform, which

requires different coding languages and tools. This can make maintenance and updates more challenging, especially if the app needs to be updated for multiple platforms.

2. Web Apps

Web apps are mobile applications that run on a mobile web browser. They are essentially websites optimized for mobile devices and can work on any platform with a web browser. Web apps are built using web technologies such as HTML, CSS, and JavaScript, and are not installed on the device itself.

Web apps are easy to develop and maintain as they use standard web technologies. They can be accessed from any device with an internet connection and do not require users to download or install anything. Web apps are also platform-independent, which means they can run on any device with a web browser.

However, web apps do not offer the same level of performance and functionality as native apps. They cannot access all of the device's hardware and software resources, and they may not work offline. Web apps also lack the responsiveness and interactivity of native apps, which can lead to a less engaging user experience.

3. Hybrid Apps

Hybrid apps are mobile applications that combine elements of both native and web apps. They are built using web technologies but are packaged and installed on the device like native apps. Hybrid apps can access some of the device's hardware and software resources, but not all of them.

Hybrid apps offer the best of both worlds, combining the portability and flexibility of web apps with the performance and responsiveness of native apps. They can access the device's camera, GPS, and other hardware resources, and can work offline. Hybrid apps can also be developed more quickly and at a lower cost compared to native apps.

However, hybrid apps may not offer the same level of performance and functionality as native apps. They may also require more maintenance and updates, as they need to be updated for both the web and native components.

In summary, native apps are designed for a specific platform, web apps run on a mobile web browser, hybrid apps combine web and native technologies, and progressive web apps provide an app-like experience on the web. Each type has its own advantages and disadvantages, and the choice of which type to use depends on the specific needs and requirements of the application. Developers should carefully consider the trade-offs between performance, functionality, and portability when choosing the type of mobile app to develop.

2. Review of programming languages used for mobile app

Sure, here is a thorough review of five popular mobile application programming languages:

1. Java

Java is one of the most widely used programming languages for developing mobile applications. It is an object-oriented language that can run on any platform with a Java Virtual Machine (JVM). Java is used for developing Android applications and offers many features such as memory management, platform independence, and object-oriented programming.

Java is a versatile and flexible language that allows developers to create robust and scalable applications. It offers a wide range of libraries and tools to help developers build complex applications more efficiently. Java is also highly secure and has a large community of developers.

However, Java cText Bodyan be complex and requires a steep learning curve for beginners. Additionally, Java applications may not perform as well as native applications due to the overhead of the JVM.

PROS

→ Java is Simple. Any language can be considered as simple if it is easy to learn and understand. The syntax of Java is straightforward, easy to write, learn, maintain, and understand, the code is easy to debug.

→ Java is an Object-Oriented Programming language. Java is an object-oriented language that helps us to enhance the flexibility and re-usability of the code. Using the OOPs concept, we can easily reuse the object in other programs.

→ Java is a secure language. Java reduces security threats and risks by avoiding the use of explicit pointers. A pointer stores the memory address of another value that can cause unauthorized access to memory.

<u>CONS</u>

→ Java provides not so attractive look and feels of the GUI

Though there are many GUI builders in Java for creating the graphical interface still they are not suitable for creating complicated UI. There are many inconsistencies while using them.

→ Java requires significant memory space. Java requires a significant or major amount of memory space as compared to other languages like C and C++. During the execution of garbage collection, the memory efficiency and the performance of the system may be adversely affected.

2. Swift

Swift is a programming language developed by Apple for developing iOS, macOS, watchOS, and tvOS applications. It is a modern programming language that offers a simplified syntax and advanced features such as automatic reference counting and generics.

Swift is easy to learn and has a simple syntax that makes it ideal for beginners. It is also highly efficient and offers better performance compared to Objective-C, the previous programming language used for developing iOS applications. Swift is also open-source and has a large community of developers contributing to its development.

However, Swift is only used for developing Apple products and is not compatible with other platforms. Additionally, the language is still relatively new, and some developers may prefer Objective-C for more complex applications.

<u>PROS</u>

→ Rapid Development Process. Swift is really easy to Read and Write because it is a clean and expressive language that has simplified Syntax and Grammar. It requires a lot less code performing the same takes when compared to Objective-C.

→ Improved Performance and Safety. As the name suggests, this programming language is designed to be swift. The focus here is on speed, performance and the ability to outperform the predecessor languages. Another biggest strength of Swift is its safety, Strong typing and error handling functions prevent code crashes and errors.

→ Easy to Integrate with Objective-C. According to developers, Swift and Objective-C should learn to Coexist. You can add new features using swift to your existing

Objective-C project or you can incorporate objective-C elements in your new swift project. Swift language is perfectly compatible with Objective-C.

CONS

→ Talent Pool is Limited. Swift community is growing fast but it is still much smaller as compared to other open source languages. According to a recent StackOverflow developer survey, only 8.1 percent of the total 78,000 respondents use Swift.

3. Kotlin

Kotlin is a programming language developed by JetBrains for developing Android applications. It is a modern language that is fully interoperable with Java and offers many features such as null safety, extension functions, and coroutines.

Kotlin is designed to be concise and expressive, making it easier for developers to write high-quality code quickly. It is also highly efficient and offers better performance compared to Java. Kotlin is also open-source and has a growing community of developers.

However, Kotlin is still relatively new and may not have as many resources and libraries available compared to Java. Additionally, it may require more time and effort for developers to learn compared to Java.

PROS

→ Fewer bugs. If you code in Kotlin, the chances of making bugs are extremely less compared to other complex programming languages. The reason is very simple and straight forward. Because it is easy to learn and only involves less code to accomplish the same thing, the number of bugs will be also less.

→ Smaller learning curve.Compared to its predecessor Java, Kotlin's learning curve is much smaller. Anyone who has a basic understanding of programming can start developing in Kotlin without any previous Android development experience.

→ Productivity improvement. Because it is easy to learn and maintain, there is a huge productivity boost compared to Java and other Android development programming languages, it may not be as fast as coding in Flutter. However, given the functionality and the well-thought features.

CONS

→ Not as mature as Java. One of the biggest drawbacks of starting with Kotlin is that it is not a matured programming language as Java. Which means that, there could be a lot of bugs and huge changes coming up with every update.

→ Learning Kotlin can be hard. When we say learning Kotlin can be hard, we are not referring to the programming language itself, but finding proper learning materials.

4.C#

C# is a programming language developed by Microsoft for developing Windows, iOS, and Android applications. It is an object-oriented language that is similar to Java and offers many features such as garbage collection, type safety, and automatic memory management.

C# is easy to learn and has a simple syntax that makes it ideal for beginners. It is also highly efficient and offers better performance compared to Java. C# is also open-source and has a large community of developers contributing to its development.

However, C# is not as widely used as Java or Swift and may not have as many resources and libraries available. Additionally, while it can be used for developing iOS and Android applications, it is primarily used for developing Windows applications.

PROS

 → C # is purely object-oriented, but C ++ is a combination of object-oriented and   procedure oriented.

→ C # is type-safe

→ A programmer does not need to pay much concentration to problems such as memory loss, which is a worrying problem for the C ++ programmer.

→ Cross-platform. An application will run fine only if the machine installed the .NET Framework.

CONS

→ A programmer cannot do low-level things like interacting directly with hardware through drivers and firmware.

→ It does not come with an independent compiler that can straightforwardly interpret the maximum levels of language to the underlying hardware architecture of a pure assembler.

5. JavaScript

JavaScript is a programming language used for developing web and mobile applications. It is a versatile language that can be used for both front-end and back-end development. It is used for developing hybrid and web applications for both iOS and Android platforms.

JavaScript is easy to learn and has a simple syntax that makes it ideal for beginners. It is also highly efficient and offers better performance compared to other web-based languages. JavaScript is also open-source and has a large community of developers contributing to its development.

However, JavaScript is primarily used for web development and may not have as many resources and libraries available for mobile application development compared to other languages. Additionally, JavaScript applications may not perform as well as native applications due to the overhead of the web browser.

In conclusion, each mobile application programming language has its own strengths and weaknesses. Java, Swift, Kotlin, C#, and JavaScript are all popular languages used for developing mobile applications, and the choice of language depends on the specific needs and requirements of the application. Developers should carefully consider the trade-offs between performance, functionality, and ease of use when choosing a programming language for mobile application development.

PROS

→ Simplicity. JavaScript is designed from the ground up to be easy to understand and learn. This simplicity pays dividends in reducing the time required to onboard and familiarise new developers with the code—even for large and complex projects.

→ Reduced Server Load. Javascript is capable of running code on the client-side, i.e. in the web browser itself. This makes tasks such as data validation or calculation possible without sending data back to the server. Client-side operations save valuable server resources and make services inherently more scalable.

→ Rich Interfaces. JavaScript gives developers the tools required to create engaging interfaces to interact with users and provide added value.

CONS

→ The main problem or disadvantage in JavaScript is that the code is always visible to everyone anyone can view JavaScript code.

→ If the error occurs in JavaScript, it can stop rendering the whole website. Browsers are extremely tolerant of JavaScript errors.

3. What's mobile app frame work, list and compare them

A mobile application development framework is a set of tools, libraries, and guidelines that developers use to build mobile apps more efficiently. It provides a structure for developing mobile apps, including pre-written code, user interface components, and other features that can be easily integrated into the app. Frameworks can help developers save time and effort by providing a foundation for building apps, allowing them to focus on customizing the app for their specific needs. Some popular mobile application development frameworks include React Native, Ionic, and Xamarin.

4. Review mobile application development frameworks by comparing them using key features

React Native:

React Native is the best JavaScript library to build native applications for all devices and platforms. With React Native, you can develop rich applications for both iOS and Android. It also allows creating platform-specific versions of various components allowing easy using of single codebase across various multiple platforms. This community-driven JS library was introduced by Facebook in 2018.

Some of the React Native features are:

- Low-code
- Compatible third-party plugins
- Declarative API for predictive UI
- Supports iOS and Android

Xamarin:

Xamarin is one of the top mobile app development frameworks and it is .Net based. This native platform is introduced by Microsoft. Being a cross-platform and open source app building platform, [Xamarin](#) offers a development ecosystem with backend, API, components, etc. It is a .Net developer platform supported by various tools, libraries, and programming languages. With [Xamarin developers](#), you can build native applications with the support of an active community for Android, tvOS, watchOS, iOS, macOS, and Windows.

Some of the Xamarin features are:

- A strong community of 60,000 contributors
- Versatile backend infrastructure
- Diagnostic tools
- Application loader
- Android SDK manager
- Storyboard files
- Google emulator manager

Ionic:

helps you to build interactive hybrid and Progressive Web Applications along with cross-platform applications. This open source framework offers premium services for creating applications. Ionic covers building applications for the web, Android, and iOS. Besides, while working in Ionic, you can constantly create applications and ship them to the deployable locations. Ionic Studio is the lightning version of Ionic and it is quite powerful. You can install it locally to have an easy visual development environment.

With its out of box features, ionic is always seen as ideal platform over the other mobile app development frameworks.

Some of the Ionic features are:

- Intuitive UI components
- Faster development time
- Powerful & stable development platform
- Evergreen community of 5 Million developers
- Complete control over app building

Flutter:

Flutter is a UI toolkit by Google help in building native applications for the web, mobile, and desktop. This UI toolkit is featured with fully customized widgets, which support to create native applications in a short period. Besides, the Flutter's layered architecture ensures a faster rendering of components. Some of the striking Flutter features are:

- Built-in material design
- Built-in Cupertino (iOS-flavor) widgets
- Rich motion APIs
- Supports both iOS & Android
- Strong widget support
- High-performance application

Swiftic:

Swiftic is one of the best mobile app development frameworks available in the iOS platform. It is featured with an easily navigable interface. Using Swiftic is a convenient option as you get 30 days money-back guarantee option for all its plans. Besides, it has another 6-months of success guarantee scheme. If you don't get any result from it, 6-month service is free.

Some of the significant features are:

- Interesting push notification
- Become a loyal shopper with a loyalty card
- Build your mobile store
- In-app coupons
- Use scratch cards to win prizes
- Easy Communication
- Menu & Ordering
- Customer Engagement
- App Promotion
- Social & Media Feeds
- App Publication Assistance
- Advanced Analytics
- Third-party integration

Onsen UI:

Onsen UI is recognized as the most powerful tool when it comes to crafting complex mobile applications. It builds brilliant HTML apps with JavaScript, HTML, CSS. It also supports the libraries like Angular JS1, Angular 2+, React, and Vue. One thing that is applaud worthy of this platform is its competency to the app's appearance depending on the platforms. For instance, if we wish to have a native look for iOS and

Android, it utilizes the automatic styling on the UI components. Its rich set of UI components includes tabs, stack navigation, lists, forms, etc.

Some of its significant features are,

- Zero time set up
- Simple to use and easy to learn
- Time-effective and cost-effective development

5. How to collect and analyse the requirements for a mobile application to be developed

1. Conduct user research: Gather information about the target audience, their needs, and preferences. This can be done through surveys, interviews, or focus groups.

2. Define the app's purpose: Determine the primary goal of the app and what problem it will solve for the users.

3. Identify features and functionalities: Make a list of all the features and functionalities the app should have to achieve its purpose. Prioritize them based on their importance.

4. Consider technical requirements: Determine the technical requirements of the app, such as platform compatibility, data storage, security, and performance.

5. Define user experience: Determine how users will interact with the app and what the user interface should look like. This can be done through wireframing or prototyping.

6. Create a project plan: Based on the requirements gathered, create a project plan that outlines the timeline, budget, and resources needed to develop the app.

7. Validate requirements: Validate the requirements with stakeholders and potential users to ensure that they align with their expectations and needs.

8. Continuously iterate: Requirements may change during the development process, so it's important to continuously iterate and refine them as needed.

6. How to estimate mobile application development cost

Estimating the cost of mobile application development can be challenging, as it depends on various factors such as the complexity of the app, the features and functionalities, the platform, and the development team's experience. Here are some steps to estimate the cost:

1. Define the scope: Determine the scope of the project, including the features and functionalities, the target platform, and the timeline.

2. Identify the development team: Determine whether you will hire an in-house team or outsource to a development agency. The cost will vary depending on the location and experience of the team.

3. Calculate development hours: Break down the project into smaller tasks and estimate the number of hours required for each task. Multiply the hours by the hourly rate of the development team.

4. Include additional costs: Consider additional costs such as project management, testing, design, and marketing.

5. Add contingency: Add a contingency budget to cover unexpected expenses or changes in requirements.

6. Compare quotes: Get quotes from multiple development teams to compare prices and services.

7. Review and adjust: Review the estimate regularly and adjust it based on changes in requirements or scope.

It's important to note that cost should not be the only factor in choosing a development team. Consider their experience, portfolio, and communication skills as well.