**CEF 440: INTERNET PROGRAMMING AND MOBILE PROGRAMMING**

**REPORT**

1. **Types of mobile applications and their differences**

A mobile application is a software developed specifically for use on small wireless computing devices, such as smartphones and tablets rather than desktop or laptop.

There are three basic types of mobile apps namely *Native, Web and Hybrid.*

* **Native app:** A native mobile application is a mobile application which is built for a specific operating system(OS). Thus, you can have a native Android mobile app,a native IOS mobile app and not to mention other platforms.

***Pros:*** 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. Native apps utilize the native device UI, giving users a more [optimized customer experience](https://clevertap.com/blog/customer-experience-optimization/).

***Cons:*** The problem with native apps is the fact that if you start developing them, you have to duplicate efforts for each of the different platforms. The code you create for one platform cannot be reused on another. This drives up costs. Not to mention the effort needed to maintain and update the codebase for each version*.*

* **Web app:** It behaves similarly to native apps but is accessed via a web browser on your mobile device.They are not standalone apps in the sense of having to download and install code into your device. They’re actually responsive websites that adapt its user interface to the device the user is on.

Web apps are designed using HTML5, CSS, JavaScript, Ruby, and similar programming languages used for web work.

***Pros:*** There is no need to customize a platform or OS. This cuts down on development costs since it’s web-based. Also, it doesn't take up space on your mobile since it is accessed directly from the web.

***Cons:*** It is also pertinent. Web apps are entirely dependent on the browser used on the device. There will be functionalities available within one browser and not available on another, possibly giving users varying experiences.

* **Hybrid app:** These are web apps that look and feel like native apps. They might have a home screen app icon, responsive design, fast performance and even be able to function offline, but they’re really web apps made to look native. Hybrid apps use a mixture of web technologies and native APIs. They’re developed using languages such as Ionic, Objective C, Swift, HTML5, and others.

***Pros:*** Building a hybrid app is much quicker and more economical than a native app. As such, a hybrid app can be the minimum viable product – a way to prove the viability of building a native app. They also load rapidly, are ideal for usage in countries with slower internet connections, and give users a consistent user experience.

***Con:*** Hybrid apps might lack in power and speed, which are hallmarks of native apps.

**Difference between Native, Web and Hybrid apps**

1. *Difference between Web and Native Apps*

*Functionality:*  Web Apps give users access only to functionalities supported by the browser. Despite it’s rich design it can’t access device features unlike Native Apps that lets the user interact with the devices internal hardware and operating systems and even grant access to native features such as Users contact list.

*User Experience:* Web Apps lack consistency when it comes to user experience. This is because of their heavy dependency on browsers. On browsers certain features or images may look different. Buttons and menu bar features may be challenging to access from mobile browsers. Browser window resizing may impact the look, feel, and functionality of the web application.

1. *Difference between Native and Hybrid apps*

*App development:* In a native app, developers have to rewrite and redesign all the app functionalities in the native development language.Whereas a hybrid app lets you write the app functionality in a single codebase. You can then wrap your code in a lightweight native app shell or container. The container enables you to take advantage of native features in your mobile devices, like hardware, calendars, and notifications.

*Cost Efficiency:* Hybrid apps achieve the same performance and user experience as native apps at a lower cost. Your developers can build them using commonly used app development languages and technologies like JavaScript, CSS, and HTML5. They can then integrate them with hybrid app development frameworks like Ionic, Cordova, or React Native. Both time and cost of development are lower, but you can still upload them to an app store to enjoy the same reach and discoverability.

1. *Difference between Web and Hybrid Apps*

*Origin reach:* You can deliver both web apps and hybrid apps from app stores. However, Web apps rank higher in search engine results with no additional effort. In addition, you get better search results than hybrid apps with the same keyword targeting.

*Performance:* In most cases, web apps tend to be lighter in size than hybrid apps. They utilize less mobile storage and memory. However, the underlying technologies are non-native. Browser dependency could result in increased mobile battery consumption for users. .

1. **Programming Languages used for Mobile programming**

Javascript

Javascript is a popularly known programming language. It is a high-level interpreted programming language that can support object-oriented and functional programming. It is a paradigm language run by a browser and it is also used to develop web pages. It is well-versed for mobile app development that can be used in any Android, iOS or Window platform.

Scala

Scala is a new programming language. It is a modern, general-purpose programming language that has a reliable static type system. Its strongest asset is its multi-paradigm language and it has also emerged as a language that can take the place of Java as it was originally designed to solve the problems of Java. It is interoperable with Java as it runs a Java virtual machine(JVM) making it one of Scala's notable features.

Java

Java is one of the most preferred programming languages by mobile app developers. It is an official Android development tool that can run in two ways; in browser window and virtual machine. Java is a general-purpose programming language that is class-based, object-oriented and designed to have as few implementation dependencies as possible.

Though Java isn’t the best choice when working on iOS apps, it provides a well when working on cross-platforms.

PHP

PHP is an open-source programming language which is easy to learn, making it easy for beginners with exceptionally smooth integrations. Developers recommend PHP when it comes to designing mobile apps with database access. It is favored for content heavy apps and can load quickly despite slow connection.

Python

Python is another widely used programming language for different purposes. Developers use it in creating cross-platforms for web, desktop and mobile apps. It can also produce android and desktop apps from scratch. It is popularly taught in school as it is simple, readable, a bit complex but not too complicated. Python has been used in creating popular applications such as Instagram, Youtube etc.

1. **Mobile App development framework**

A mobile app development framework is a software framework that is designed to support mobile app development. They are templates and fundamental structures that facilitate the work of developers as they build and optimize apps.

1. **Review of mobile app development frameworks**

Examples of frameworks include;

* **React Native framework**

It allows developers to create modern and fast apps for both iOS and android. It is based on the JavaScript programming language.

It has a high performance, a large number of free to use third party libraries and plugins.

* **Flutter framework**

Apps built with flutter can be shared across android and iOS platforms. It offers the best performance among cross-platform solutions. It has a feature called hot Reload which allows developers to implement code changes almost instantly without refreshing the whole app hence, speeds up the delivery process. It also mimics the platform’s visual style, so the appearance of the elements and animations are similar to native.

* **Native iOS (Swift)** **framework**

It is a successor to Objective-C that was previously used for creation of iOS apps. Apps built with Swift can be shared across Apple platforms such as macOS, iOS and watchOS. Some of its features are short build time, small size of apps and super-fast performance.

* **Native Android (Kotlin) framework**

It allows developers to write lower-level code which makes it easier to control the environment, build complex features and debug apps. Apps built with this framework do not require an “interlayer” of code to access devices hardware such as camera, microphone or GPS.

* **Ionic framework**

It is based on JavaScript, HTML and CSS and supports easy integration with numerous third-party libraries and other open-source frameworks. Ionic offers a great selection of pre-built UI components and their appearance updates automatically to match platforms’ visual styles. However, the performance of apps developed with Ionic lags slightly behind React Native and Flutter.

When building a mobile application, certain factors have to be checked or reviewed in order to make the right decision and build correctly. The table below shows how various frameworks can differ in terms of some factors;

|  | **React Native** | **Flutter** | **Native iOS** | **Native Android** | **Ionic** |
| --- | --- | --- | --- | --- | --- |
| **Language** | Java Script | Dart | Swift | Kotlin | Java Script |
| **Performance** | High | High | Very high | Very high | Low |
| **Cost and time to market** | Cheaper and faster | Cheaper and faster | More expensive and slower | More expensive and slower | Cheaper and faster |
| **Community support** | Very popular | Popular | Very popular | Very popular | Not so popular |
| **Complexity** | Provide great optimization | Provide great optimization |  |  | Suitable for simple apps |

1. **How to collect and analyze requirements**

The analysis requirements phase is the second phase of the software development life cycle.The requirements in this phase can be categorized into the functional and the non-functional requirements. In order to collect and analyze these requirements for a mobile application the following steps can be taken;

1. **Identify the end users and developers of the Mobile application**

The first step of the requirements analysis process is to identify Developers who are the main sponsors of the project. They will have the final say on what should be included in the scope of the project and hence will be able to ensure that mobile applications remain functional.The developers will determine the non-functional requirements of the mobile application.

Next, identify the end-users of the mobile application. Since the product is intended to satisfy their needs, their inputs are equally important.They will determine the functionalities of the mobile application.

1. **Capture the requirements of the mobile application**

Each of the Developers and end-users are to be asked their requirements for the desired mobile application. Here are some requirements analysis techniques that will used to capture requirements

#### 1. Holding One-On-One Interviews with developer and end-users to get specific requirements

2. Conducting group interviews to understand the flow of information between developers and end-users, this will ensure that they will be no conflict of ideas later development of the mobile application

3. Utilize use cases to provide a walkthrough of the mobile application through the eyes of the end users to help visualize how the mobile application will actually work.

4. Build Prototypes to provide users a sample look on how the final product will be to help address feasibility issues and identify problems ahead of time.

5 Evaluate existing prototypes and compare and identify knowledge gapsy

1. **Categorize requirements of the mobile application**

Functional Requirements: They define what a product must do, what its features and functions are.

Non-functional Requirements: They describe the general properties of a system. They are also known as *quality attributes.* Examples include security, availability, performance and usability.

1. **Interpret and record requirements of the mobile application**

Once the requirements are categorized, determine which requirements are actually achievable and document each one of them.

1. **Sign off**

Once a final decision is made on the requirements, we ensure that you get a signed agreement from the Developers. This is done to ensure that there are no changes or uncontrolled growth in the scope of the mobile application development.

**6. How to estimate the mobile app development cost**

This depends on a variety of factors such as, the app’s functionality, design complexity,platforms supported and the level of expertise required to develop it. We can estimate the cost of developing a mobile app using the following steps;

* **Define our app’s requirements:**

We can start by defining our app’s features, functionalities and design.

* **Choose a platform:**

We decide which platforms we want our app to support i.e if it supports iOS or android or both.

* **Choose a development approach:**

Here we can develop the mobile application choosing what model methodology of the Software Development Life Cycle each of which have a different cost structure

* **Consider outgoing costs:**

Consider outgoing costs such as updates, maintenance, marketing.

* **Research app development cost:**

It is important to research the cost of app development based on the platform you choose.