**AIM:**

1. Comparative Analysis of Various Mobile OS.
2. Comparative Analysis of Various Mobile Application Languages.
3. Comparative Analysis on Various Mobile Application Development Environments.
4. Comparative Analysis of Security or Malware Protection in Various Mobile OS.

**THEORY:**

1. Comparative Analysis of Various Mobile OS:

* Here, for an in-depth analysis we have chosen these three OS as the majority of users and devices are currently running on one of these three OS.

|  |  |  |  |
| --- | --- | --- | --- |
| **Criteria** | **Android** | **Apple iOS** | **Windows Mobile** |
| **Language** | Java (Also C and C++) | Objective-C | C#, Visual Basic, C or C++ |
| **Application** | apk | ipa | cab/XAP/APPX |
| **Development Tool** | Eclipse or IntelliJ IDEA | Xcode (only on MAC) | Visual Studio 2012+ (only on Windows) |
| **Development Cost** | Free | Tools are free but need Publisher Account | Visual Studio Cost |
| **Need for Publisher Account** | No | Yes | No |
| **OS Family** | Linux | Darwin | Microsoft Windows |
| **Supported CPU Architecture** | ARM, x86, x86-64 | 64-bit ARM V&A | 32-bit ARMv7-A and 32-bit ARMv8-A |
| **License** | Free and Open-Source, but usually bundled with proprietary apps drivers. | Proprietary except for open source components. | Proprietary, Commercial Software |
| **Garbage Collector** | Yes | No | Yes |
| **Memory Usage** | High | Low | High |
| **Rooting** | Allowed | Allowed | Allowed |
| **Interface** | User-friendly | User-friendly | Not user-friendly |

1. Comparative Analysis of Various Mobile Application Languages:

Here, for an in-depth analysis, we have chosen these Mobile Application Languages as these are currently the most widely used mobile application development languages.

**Flutter**

It is a Reactive cross-platform mobile development framework that helps build native apps for both iOS and Android with a single code base.

* Language Stack: Dart has borrowed aplenty from Java and C/C++. So, using Dart as a programming language, developers can create mobile applications for Android, iOS, and Windows.
* Performance: As Flutter eliminates the JavaScript bridging for initiating interactions with native elements, thereby greatly enhancing the app speed.
* User Interface: Thanks to an easy-to-use interface, Flutter is fun to use. Developers get to use several widgets, which shorten the development time.
* Pricing: An open-source platform, Flutter is free of cost.
* Specific Feature: Hot-reload feature, which allows developers to make changes at the backend and see the implementation simultaneously. It allows them to immediately assess if the changes are in sync with other components of the app.
* When to Use? If UI is a key priority of your app.

**Java/Kotlin**

Kotlin is often termed as “The Java Alternative”. Kotlin is a general-purpose, cross-platform, statically typed programming language with type inference and solely used for Android App Development.

* Language Stack: Resembling Java, Kotlin aims at JVM (Java Virtual Machine), JavaScript, and Native.
* Performance: Though the bytecode of Kotlin is similar to Java, the former is considered a better alternative to the latter. Java/Kotlin is a vibrant and extensible programming language. A well-designed languages with an intuitive set of APIs, it brings in quick results.
* User Interface: Kotlin stands out when it comes to features and user experience as it is interoperable in Java. Developers use it along with SDKs to create remarkable and stand-out user experiences.
* Pricing: Kotlin is free of cost, and its source code is up there on GitHub.
* Specific Feature: As they are general-purpose programming languages and not a software development kit, there is no need for developers to port the whole application to it.
* When to use? It is the perfect choice for Android app development.

## **React Native**

A JavaScript framework built upon the React library, the React Native is the creation of Facebook. Like Flutter, it allows the development of iOS and Android apps using a single code base.

* Language Stack: React Native supports JavaScript and React.JS. It even enables building a few elements in Java, Swift, or Objective-C, making it the foremost mobile development framework when it comes to the support for programming languages.
* Performance: The framework provides the code elements specifically to native APIs, generating performance that resembles that of the native applications.
* User Interface: Innumerable ReactJS library UI elements give developers a lot of power to create an impressive graphical interface.
* Pricing: React Native is open source, so developers can use it free of cost.
* Specific Feature: The framework offers high code reusability across platforms. The developer community has created a lot of libraries that can be used as building blocks to accelerate development.
* When to Use? If you want universal code for mobile, web, and desktop.Comparative Analysis on Various Mobile Application Development Environments:
* Here, for an in-depth analysis, we have chosen these Mobile Application Development Environments as these are currently the most widely used mobile application development environments.

|  |  |  |  |
| --- | --- | --- | --- |
| **Criteria** | **Android Studio** | **Xcode** | **Visual Studio** |
| **Supported Mobile Platforms** | Android | iOS | Windows |
| **Supported computer OS** | Windows, Linux, Mac OS X | Mac OS X | Windows |
| **Programming Languages** | Java and Kotlin | Swift, Objective-C | C#, C++ |
| **Speed and Complexity of installation**  **(1-5)** | 5 | 4 | 5 |
| **Official Documentation and Community**  **(1-5)** | 4 | 5 | 1 |
| **Complexity of Development**  **(1-5)** | 4 | 4 | 3 |

1. Comparative Analysis of Security or Malware Protection in Various Mobile OS:

* Android and iOS are two of the most used mobile operating systems currently in the market. Therefore, here we have chosen these two for our comparative study.

|  |  |  |
| --- | --- | --- |
| **Criteria** | **iOS** | **Android** |
| **Code Signing** | Yes | Yes |
| **Application Permissions** | No | Yes |
| **Sandboxing** | More Effective | Less Effective |
| **Data Encryption** | Yes | No |
| **Memory Protection** | Yes  ASLR | Yes  MMU + ASLR  (Android v4) |
| **Code Protection** | Yes  DEP | Yes  Java Type Safety + DEP  (Android v2.3) |
| **Availability of Antivirus tools** | Yes  Few | Yes  Many |
| **Component Protection** | Not a component-based | Yes |
| **Development tools availability** | Partial | Yes |
| **Development friendliness** | No | Yes |
| **Installation Vectors** | Restricted | Multiple |
| **Application Portability** | Yes | Yes |
| **Unofficial Repositories** | No | Yes |
| **Application Testing** | Yes | No |
| **Application Remote Removal** | Yes | Yes |