

LAB ASSIGNMENT#01

MAD

|  |  |
| --- | --- |
| **Student Name** | **Syed Muneeb Ahmed** |
| **Registration Number** | **SP20-BSE-019** |
| **Program** | **BS-(SE)** |
| **Semester** | **VI** |

Submitted to: Sir KAMRAN

* **COMPARISON OF NATIVE AND CROSS PLATFOM MAD:**
* **NATIVE PLATFORM:**

According to statistics, this is the most common type of mobile application. Native apps are developed for a specific platform: iOS or Android. The default browser, calendar, mail on your smartphone are all native apps, created taking into account the features of the operating system. [Nexar](https://ideasoft.io/project-nexar.html" \t "_blank) is an example of such an app. By focusing on a specific platform, native applications are fast and efficient, they can work without an Internet connection, and are intuitive for the user. However, if you want to create an app that runs on both iOS and Android, you will have to develop two separate apps, which increases the development cost.

* **CROSS-PLATFORM:**

The main advantage of this type of application is in the name. Cross-platform apps run simultaneously on multiple operating systems, which means developers can use the same code base that connects to native components via the so-called bridges. Cross-platform is achieved by compiling source code for execution on each platform. Each separate compilation will result in a separate executable file. This type of mobile development is cheaper than the native one. Yes, it sounds great, but as with each option, there are some drawbacks here. Cross-platform apps are less flexible, they are harder to maintain, and they have a lower user experience because this kind of app doesn’t take into account the uniqueness of each platform.

* **DIFFERENT SCENERIOS OF PLATFORM:**
* **NATIVE PLATFORM:**

1. When a project requires extensive data processing and, therefore, intensive use of the device’s memory, the most obvious option is for the members of the team to choose a 100% native development. The higher performance of native applications is a significant aspect. A design and development for Android, another one for iOS and another for Windows Phone, if the idea is finally to have an app in this operating system.
2. If the choice is a native application, this will have implications in the design and development, which will take the project to specific usability standards for each platform. The native elements of Android and iOS are completely different and this conditions the entire process.
3. Native applications usually have greater visibility at app stores and often get better user ratings and recommendations. This is essential for [a good ASO positioning](https://bbvaopen4u.com/es/actualidad/algunos-consejos-seo-de-ultima-generacion-para-principiantes).

* **CROSS-PLATFORM:**

1. Compared with the obvious benefits offered by a native application, web project encapsulating solutions or cross-platform applications considerably shorten development and deployment times at app stores. There are therefore fewer costs and the risk of opting for projects that will fail is lower.
2. Moreover, we should not forget that cross-platform applications not only cut costs, but they also maximize the profit, either through exposure to a larger number of users (the company will have an application in most environments) or on the revenue side.
3. It is not necessary to incorporate new profiles to the team other than those that can already be serving a desktop version: having a team knowledgeable in HTML, CSS and JavaScript would be enough.

* **LIST OF FRAMEWORK:**
* **NATIVE PALTFORM:**

1. Xamarin
2. RhoMobile
3. Native Script
4. Kony Mobile Platform
5. Monocross
6. Sencha touch
7. Appcelerator
8. Phone Gap
9. CodeName One