This link has a complete checklist for each technology including many not listed below. Additional info is contained in the table based on personal research and usage of the software.

<http://www.markus-falk.com/mobile-frameworks-comparison-chart/>

|  |  |  |
| --- | --- | --- |
|  | Pro | Cons |
| Native Apps  **Note: Native app does not mean everything is written as native. It means that only the platform specific functionality and the web page viewer is written in native code. The main app would still be written as a normal web page, using the web development languages we know.** | - Top performance  - Less bandwidth consumption (certain things can be stored locally)  - App store  - Can use device-specific features to greatly improve our product (ex: sending notifications to the user when there’s a problem with the order). | - Possible costs (Only when/if we decide to release into App Stores. Developing does not cost anything)  - Lots of time and resources  - Don’t have the man power  - No experience using the native languages (could also be a good thing, depending how we look at it). |
| The Web | - Supports all platforms  - Simplest solution | - No access to phone software/hardware  - No app store  - Performance depends on network/browser/OS |
| Client-side Web | - Supports all platforms  - Access to GPS & Gyro  - Good for prototyping  - Looks like an app | - No access to most of phone’s software/hardware  - Performance depends on network/browser/OS  - No app store |
| Hybrid Apps | | |
| PhoneGap | - Access to phone software/hardware  - App store  - Can make native extensions  - Lowest-common denominator  - Strong community | - Performance depends on network/browser  - Just like a web app  - Cannot use native user UI interfaces  - Lowest-common denominator  - Setup is not user friendly |
| **Titanium**  **\*for data-driven applications**  **\*WP support is on the way**  **\*Not an attempt at write once, run everywhere** | - Close to native performance  - Simple and powerful API  - WebView is only used for custom logic.  - Core of portable APIs and platform-specific APis  - Extendable  - Familiar feeling like Eclipse  - Highly supported  - Very Popular | - No image processing.  - App logic/handling events is javascript  - No Windows Phone (right now)  - Some UI components are not as good as native counterparts  - No NFC Support currently |
| **Rho Mobile** | - Critically Acclaimed  - Well Supported (Motorola)  - Familiar Look and Feel  - Model Creator, a cool feature  - Built in emulator (x times faster)  - Has NFC support | - Based on Ruby |
| Nomad | - Used with Visual Studio | - Early phase of development  - Not very popular  - It doesn’t support deployment to Windows Phone |
| **MoSync** | - Most easy to use so far  - Familiar Eclipse feeling  - Supports Windows Phone  - Very Simple to use  - Can use C++ | - Big lack of support  - Not as popular as Titanium or Phone Gap |
| **Corona** | - Professionally used  - Results are impressive | - Seems to be made for games |

**What we need**

* Support for Android, iOS, WP
* Fast and responsive
* Access to phone’s hardware/software