* 1. **INTRODUCTION**

**What is Android?**

**Android** is an operating system that powers mobile devices and is one of the most popular mobile platforms today. The android platform runs on hundreds of millions of mobile devices throughout the world. It's the largest installed operating system of any mobile operating system and growing rapidly day by day.

You can develop apps and games using Android and it gives you an open marketplace for distributing your apps and games instantly.

**Android** is the operating system for powering screens of all sizes. The Android version is named after a dessert. The latest version of android is Android 9.0 – Pie. The following table shows how the android platform evolves.

**Android** is an open source and Linux-based Operating System for mobile devices such as smartphones and tablet computers. Android was developed by the Open Handset Alliance, led by Google, and other companies.

**Android** offers a unified approach to application development for mobile devices which means developers need to develop only for Android, and their applications should be able to run on different devices powered by Android.

The first beta version of the Android Software Development Kit (SDK) was released by Google in 2007, whereas the first commercial version, Android 1.0, was released in September 2008.

On June 27, 2012, at the Google I/O conference, Google announced the next Android version, 4.1 Jelly Bean. Jelly Bean is an incremental update, with the primary aim of improving the user interface, both in terms of functionality and performance.

The source code for Android is available under free and open-source software licenses. Google publishes most of the code under the Apache License version 2.0 and the rest, Linux kernel changes, under the GNU General Public License version 2.

**Features of Android**

Android is a powerful operating system competing with Apple 4GS and supports great features. A few of them are listed below:

|  |  |
| --- | --- |
| Feature | Description |
| Beautiful UI | Android OS basic screen provides a beautiful and intuitive user interface. |
| Connectivity | GSM/EDGE, IDEN, CDMA, EV-DO, UMTS, Bluetooth, Wi-Fi, LTE, NFC, and WiMAX. |
| Storage | SQLite, a lightweight relational database, is used for data storage purposes. |
| Media support | H.263, H.264, MPEG-4 SP, AMR, AMR-WB, AAC, HE-AAC, AAC 5.1, MP3, MIDI, Ogg Vorbis, WAV, JPEG, PNG, GIF, and BMP |
| Messaging | SMS and MMS |
| Web browser | Based on the open-source WebKit layout engine, coupled with Chrome's V8 JavaScript engine supporting HTML5 and CSS3. |
| Multi-touch  Android has native support for multi-touch which was initially made available in handsets such as the HTC Hero. |  |
| Multi-tasking | Users can jump from one task to another and same time various applications can run simultaneously. |
| Resizable widgets | Widgets are resizable, so users can expand them to show more content or shrink them to save space |
| Multi-Language | Support single direction and bi-directional text. |
| GCM | Google Cloud Messaging (GCM) is a service that lets developers send short message data to their users on Android devices, without needing a proprietary sync solution. |
| Wi-Fi Direct | A technology that let apps discover and pair directly, over a highbandwidth peer-to-peer connection. |
| Android Beam | A popular NFC-based technology that let users instantly share, just by touching two NFC-enabled phones together. |

**Android Applications**

Android applications are usually developed in the Java language using the Android Software Development Kit.

Once developed, Android applications can be packaged easily and sold out either through a store such as Google Play or the Amazon Appstore.

Android powers hundreds of millions of mobile devices in more than 190 countries around the world. It's the largest installed base of any mobile platform and is growing fast. Every day more than 1 million new Android devices are activated worldwide.

This tutorial has been written to teach you how to develop and package Android applications. We will start with environment setup for Android application programming and then drill down to look into various aspects of Android applications

**HISTORY OF MOBILE APPLICATION DEVELOPMENT**

To understand what makes Android so convincing, you must study how mobile development has evolved and how Android differs from other mobile platforms.

The Motorola DynaTAC 8000X was the first commercially available cell phone and it is of brick size. The first-generation mobile was expensive, not particularly full-featured, and has Proprietary software.

As mobile phone prices dropped, batteries improved, and reception areas grew, more and more people began carrying these handy devices. Customers began pushing for more features and more games. They needed some way to provide a portal for entertainment and information services without allowing direct access to the handset.

Early phones have postage stamp-sized low-resolution screens and limited storage and processing power, these phones couldn’t handle the data-intensive operations required by traditional web browsers. The bandwidth requirements for data transmission were also costly to the user.

**Wireless Application Protocol**

The Wireless Application Protocol (WAP) standard emerged to address the above concerns. WAP was a stripped-down version of HTTP. WAP browsers were designed to run within the memory and bandwidth constraints of the phone. Third-party WAP sites served up pages written in a mark-up language called Wireless Markup Language (WML). The WAP solution was great for handset manufacturers and mobile operators. Phone users can access the news, stock market quotes, and sports scores on their phones.

WAP fell short of commercial expectations due to the following reasons and Critics began to call WAP “Wait and Pay.”

• Handset screens were too small for surfing.

• WAP browsers, especially in the early days, were slow and frustrating.

• Reading a sentence fragment at a time, and then waiting seconds for the next segment to download, ruined the user experience, especially because every second of downloading was often charged to the user.

• Mobile operators who provided the WAP portal often restricted which WAP sites were accessible.

**Proprietary Mobile Platforms**

Writing robust applications with WAP, such as graphic-intensive video games, was nearly impossible. Memory was getting cheaper, batteries were getting better, and PDAs and other embedded devices were beginning to run compact versions of common operating systems such as Linux and Windows. A variety of different proprietary platforms emerged and developers are still actively creating applications for them.

Some of the examples of proprietary mobile platforms are:

• Palm OS (now Garnet OS)

• RIM BlackBerry OS

• Java Micro Edition [Java ME] 5

• Binary Runtime Environment for Wireless (BREW)

• Symbian OS

• OS X iPhone

Each platform has benefits and drawbacks.

**INTRODUCTION TO MOBILE APPLICATION DEVELOPMENT**

**Mobile application development** is the process by which application software is developed for small low-power handheld devices such as personal digital assistants, enterprise digital assistants, or mobile phones. These applications are either pre-installed on phones during manufacture or downloaded by customers from various mobile software distribution platforms.

Mobile software is developed by using different platforms and programming languages based on the target mobile device. There are many different hardware components found in mobile devices so their applications are developed using different software architectures.

Most of the methodologies in use are based on the model-driven approach which has three different views of the application development process:

(1) the application itself and its structure,

(2) the business logic and

(3) the graphical user interface of the application.

Weapons of Mobile Development

Hardware

Computers

Mobile Phones

Framework

Programming Language

Integrated Development Environment

Compiler

Simulator

Monitors

Platforms Available

1. J2ME

Programming Language: Java

Integrated Development Environment: Netbeans/Eclipse

Framework: KVM

Hardware Deployment: Multiple

Installer Packaging Options: Jad/Jar

1. iOS (Apple)

Programming Language: Objective C

Integrated Development Environment: Xcode

Framework: Cocoa Touch

Hardware Deployment: iPhone, iPad, iPod

Installer Packaging Options: .app/.ipa

Development Tool Cost

1. Android

Programming Language: Java

Integrated Development Environment: Eclipse/Netbeans

Framework: Dalvik VM

Hardware Deployment: Android Only

Installer Packaging Options: .apk

Development Tool Cost

1. BlackBerry

Programming Language: Java

Integrated Development Environment: JDE - BlackBerry Java Development Environment

Framework: Dalvik VM

Hardware Deployment: Android Only

Installer Packaging Options: .alx, .cod

1. Windows Mobile

Programming Language: C#, VB.NET, Basic4ppc

Integrated Development Environment: Visual Studio 2008, 2005, 2003, Basic4ppc IDE

Framework: .Net Compact Framework

Hardware Deployment: Windows Mobiles

Installer Packaging Options: .ota/.cab

**Characteristics of Mobile Applications Development**

1. **Mobile devices are deeply personal.**

The phone is something that belongs to just one individual. It’s private, personal, and likely to be carried with you everywhere you go. We like to customize our phones and they have become a symbol of social status. The phone I carry, and the content I have, become part of my brand.

1. **Mobile devices are hyper-social.**

The phone holds all of your contacts and allows you to do much more with them. Basic communications include voice calling, text messaging (sending a message of 140 characters or less. SMS or Short Message Service was the first Twitter), multimedia messaging (sending videos or photos along with text messages), and e-mail. Beyond the basics, you may want to consider online presence, physical proximity, or relationship strength, to allow for more “contextual communications.”

1. **Mobile devices are location-aware.**

Even if you don’t know exactly where you are, your phone does! It can also help you discover what is located nearby, get directions, or even interpret what is right in front of you. If your friends are opt-in, you can display their present location. This is one of the most powerful features of mobile devices and one that, until recently, was reserved for native applications. However, the most recent versions of Fireworks, Chrome, and the Android browser, now have location APIs as well.

1. **Mobile devices promote quick focused usage.**

Assuming that one is on the move, I will stop to navigate my phone when I need a vital bit of information like my shopping list, current traffic, flight arrival times, stock prices, or recent messages. The exception to this is entertainment. More and more, mobiles fill those empty spaces when we’re standing in line, waiting, or sitting for long idle periods — we reach for our phones to watch our favorite podcasts, scan tweets, listen to music, or immerse ourselves in a good film.

1. **Mobile devices are sometimes connected.**

When planning your mobile offering, it’s important to develop for both online and offline scenarios. Sometimes your users will be riding the subway, on a plane, traveling abroad, or otherwise unable to connect to the Internet, their home network, or both. Although unlimited data plans do exist, they’re not as prevalent as you may think. So if you want to avoid costly phone bills you need to consider how users will engage with your content when they’re offline.

1. **Mobile devices support a spontaneous lifestyle.**

Carrying all of your PIM (personal information management) data means you can respond to events as they unfold and share new data in real-time. For example, if traffic is blocked, I can meet my party at a different restaurant. If I’m standing in line at the cinema and tickets sell out, I can quickly locate an alternative and have my friends meet me there.

1. **Connectivity.**

Apps are always online as the device is constantly logged in to the mobile network. This allows user-specific information or notifications to be pushed to the App as they are available. In combination with the ubiquity of mobile applications, this is the most important characteristic. A service, that must be called actively to get a reaction is useless in a mobile environment. And with the growing number of Apps on each smartphone, this push-functionality becomes critical to keep an App in the users' minds.

1. **Convenience**

An emotional design and a simple (one-handed) handling guarantee high acceptance. A good App can do its job in different contexts and fast varying situations (changing environmental light and noise, unsteady movement of the device, etc.). So the information architecture and the overall usability must be planned with care to create a fitting and joyful interaction flow. Of course, good content also counts for convenience. Analyzing the users’ needs and creating a useful idea out of it is still essential.

1. **Localization**

Localization and the possibility to offer location-based information is a key feature that makes mobility vivid and practical. It separates the wheat from the chaff as it embeds the App in the users’ context. Sure, this feature might not make sense for every App but localization must not always be thought of as the big thing. Just think of automatically associating the location with a note or photo or by just limiting possible options or sorting places. It can be useful in little, just creating a good experience for the user.

1. **Reachability**

Reachability covers a more social attribute given by the nature of mobile applications themselves. A good App can be used – and more important makes sense – anywhere at any time. The core of mobile devices is to be used anywhere at any time. The same is true for Apps where reachability has become available. Not in sense of usage, but in sense of updated information and perpetual usefulness.

1. **Security**

Security has several facets. The data transferred over the network must be encrypted through the carrier network. As some Apps sync data with online, web-based applications, the storage of this data on the server must also be secured. Another aspect concerns the data on the device itself. I don’t want anybody playing around with my mobile phone and getting access to my bank account data. Mobility is delicate, and so is the date aggregated and generated in this context.

1. **Personalization**

Creating personalized content based on individual usage or context is another characteristic. It builds on all previous characteristics as it is a kind of melting down of all of them. I want my App fitting my needs and I want my App behaving like I want it to do. This need covers not only personalized content but also control over data stored, shared, or used for further actions. The option of turning localization on or off is true personalization. An individual background or personal categories are a convenience.

**Benefits**

* Most of these devices have easy functionality and use
* Mobile Application development is a key part of the evolution of mobile phones
* How does this application help me?
* Does it make my work easier?
* Does it enable me to easily use it?
* Very popular mobile applications
* Game applications
* Weather information
* Sports scores and updates applications
* The Bible, Koran
* The Dictionary
* Fitness and health applications etc…
* There are also free trial applications that are in the development stages
* Select an application that helps you have functionality and style as well, like
* The blackberry has a set of sophisticated and fabulous features
* The stock viewer enables you to keep track of your stocks and business listings
* iPhone’s breathtaking application is the text n drive
* The mobile application development industry truly is bringing innovative solutions to the technological realm.

**MOBILE APPLICATION PLATFORMS, TOOLS, AND TECHNOLOGIES.**

**1 INTRODUCTION**

“Mobile is the future” - Eric Schmidt (Ex-CEO, Google)

Mobiles are rapidly becoming an integral part of people’s lives. Gartner expects the mobile applications industry to increase to a massive 21.6 billion sales figure by 2013 generating revenue of 29.5 billion dollars. Therefore, as developers, we can look at this phenomenon as an opportunity to develop software that has the potential of reaching a vast audience.

Also, for individual mobile developers, this industry could offer very lucrative benefits. Even today, the possibility of making profits is huge. Success stories of developers, both novice and expert, becoming instant millionaires after creating the hit application(s) are plenty. The job opportunities for mobile developers are very good too. According to ACM career news, there is a high demand for mobile application developers but not enough skilled people are available(ACM CareerNews 2010). Hence, if, as a developer, your resume boasts mobile development knowledge, you have an advantage over those who don’t. This knowledge will be useful even if you do not want to focus your career just on mobile application development. More and more businesses are entering the mobile application industry and introducing mobile applications for their users (at&t Media Newsroom 2011); hence you can use this knowledge in almost any field you enter.

Thus, this chapter aims to be a starting point for the readers into the world of mobile application development by giving an introduction and basic overview of different aspects of mobile platforms and applications.

The chapter has been organized as follows: Firstly, we will cover some basic concepts important to mobile development. Then, we will give a step-by-step guide to developing applications, from designing to publishing. Simultaneously, we will introduce and explain the related concepts. Later, we cover the differences between desktop development and mobile development - mainly the constraints and opportunities mobile development has. To conclude, we look at what the future of mobile holds and then give a summary of the concepts covered in the chapter.

1. **CORE CONCEPTS**

Before we can delve into mobile application development, let’s look at some of the core concepts.

* 1. **Platforms and operating systems**

To be able to develop mobile applications, we need to first understand the underlying principles of mobile platforms and how they are related to mobile applications. Mobile platforms are those that allow software and services to be run on devices (Fling 2009). Examples of mobile platforms include Palm, BlackBerry, iPhone, Android, and Windows Mobile.

Mobile operating systems provide tools that allow the application to share data and services. Examples of the mobile OS include Palm OS, Symbian, Windows Mobile, Mac OS X, and Android.

(Note: For simplicity, ‘mobile platforms’ and ‘mobile operating system’ will be used interchangeably in the rest of this chapter).

For this chapter, we will be focusing on three platforms: Software Development Tools and Technologies 6

Android,

iOS and

Windows Phone 7.

* 1. **Application frameworks and applications**

Mobile application frameworks are a set of class libraries that provide developers with an interface to build applications. Examples of application frameworks include Java ME, Cocoa Touch, Android SDK, and WebKit.

Applications are built using application frameworks on the given platform. The next section covers steps the developers need to take while developing mobile applications.

Mobile applications are developed targeting a specific OS which can be done by either using native or cross-platform development. This thesis is reviewing different mobile development technologies, mobile operating system characteristics, mobile native development, mobile cross-platform development, and previous literature with an existing decisional framework for selecting a native or cross-platform approach.

**Mobile Development Technologies**

A developer requires extensive knowledge of Java or Kotlin for developing applications on Android and Objective-C or Swift for iOS. Deploying applications on both platforms, the native application development process would require a big investment in time, money, and effort and there is a need to re-write the code to be run in different runtime environments. Therefore, many companies are using a cross-platform approach.

Cross-platform frameworks are offering developers a tool for developing mobile applications without the need for extensive knowledge in either Java or Swift. Developers can use C# with Xamarin, JavaScript with React Native, Dart with Flutter, or even web technologies with Adobe PhoneGap. These are competencies companies might already have in-house and therefore a cross-platform approach might be a suitable selection. This will be detailed further in section 2.1.3.

Cross-platform approaches do not give the same user experience as a native approach. A native touch and feel can only be achieved using truly native components inside the app. Deep integration with the device and improved performance are important parameters to take into consideration when developing an app.

According to Shah et al. (2019, p 6), native applications are preferred when it comes to performance-related parameters like access to native device APIs, ease of update, rendering UI, and providing a better user experience. Native applications require platform-specific knowledge which is often cost intensive. Shah et al. (2019, p 7) conclude that despite the disadvantages of a cross-platform approach, they often are the preferable approach except when developing high-performance applications.

**1.1 Mobile Operating System Characteristics**

Today the market is dominated by two contenders – Android and iOS which have different characteristics and is of interest to know before developing an app.

**1.1.1 Android OS**

Android’s primary purpose is to create an open software platform available for carriers, OEMs, and developers to introduce a product that improves the mobile experience for users. Android is designed so no industry player controls or restricts the innovation of another resulting in a product with source code open for customization and porting. To prevent uncontrolled customization that can lead to incompatible implementations, the Android Open Source Project, led by Google, maintains the Android Compatibility Program that spells out what it means to be Android compatible and what’s required of device builders to achieve that (Android Source, 2019).

Android is an open-source Linux-based software stack. Using a Linux kernel allows manufacturers to develop hardware drivers for a well-known kernel. A hardware abstraction layer provides interfaces to device hardware capabilities with the Java API Framework (Android Developer Platform, 2019).

Android applications cost $25 to host on Google Play and developing an application for Android is easier compared to iOS since the guidelines are not as strict as iOS (Pratap, 2018).

**1.2 Mobile Application Native Development**

When developing native mobile applications, developers target one specific platform using its SDK and frameworks. The application is tied to that specific environment (Heitkötter et al. 2013, p 3). The native application development approach for the two mobile OS dominating the market today, Android and iOS (Statcounter, 2020), is hereby presented.

**1.2.1 Android Native Development**

Applications for Android can be written using Koitlin, Java, or C++ programming languages. The Android SDK tools compile the source code, data, and resource files into an APK that contains all the content of an Android application and is the file that Android-powered devices use to install the application (Android Developer 13 Fundamentals, 2019). Each application has only access to the components required to do its work and cannot access parts of the system which is not given permission (ibid, 2019). According to Android Developer Fundamentals (2019), each application lives in its security sandbox protected by the following Android security features:

• Android operating system is a multi-user Linux system where each application is a different user

• The system assigns each application a unique Linux user ID and only the user ID assigned to that application can access the files that the system has set permission for.

• Each process has its virtual machine, so an app’s code runs in isolation from other applications

• Every application runs in its own Linux process. The Android system starts the process when any of the app’s components need to be executed and shuts down the process when it is no longer needed, or the system must recover memory for other applications.

An Android application must declare all components it is using in a manifest file, AndroidManifest.xml including the permissions the application requires, minimum API level, declare hardware or software features such as camera, and declare the API libraries the application needs to be linked with (ibid, 2019).

The entire feature-set of the Android OS is available using Java APIs enabling building UI, access to non-code resources, notifications, and activity manager to provide a common navigation back stack and content providers to access data from other applications. With this framework, developers have full access to the same framework APIs that Android system applications such as email, SMS, calendars, internet browsing, etc are using (Android Developer Platform, 2019).

**Mobile Application Cross-Platform Development**

Developing native applications for different platforms means different programming languages for each platform, different APIs, and different IDEs. Cross-platform development tools have been developed to write an application source code once and run it on different OS (Palmieri et al. 2012, p 180). Shah et al. (2019, p 2) presents the following advantages and disadvantages of developing using a cross-platform approach: Advantages

• WORA, which is the primary advantage. The application is implemented using a single code base but can be deployed to multiple platforms.

• Cross-platform tools usually use well-known programming languages and syntaxes

• Major tools have a large community-driven open source platform. New features and modules are continually added, updated, and revised.

• Cost-cutting can be done effectively since development is simpler compared to multiple platforms

• Updates and bug fixes can be rolled out for every platform affecting all platforms at once. Disadvantages

• Native applications are designed to work flawlessly for the specified platform with better performance compared to cross-platform counterparts.

• Native applications have a deeper integration with device API for all features.

• Native applications provide a rich UX. Rendering of high-end graphics is only effectively possible with native application development.

• Deployment to the application store of the respective platform is usually lowered compared to cross-platform applications. Some applications which simply inject HTML and JavaScript code inside a container are rejected from getting published.

• Complex applications increase the risk of the application freezing and crashing because the support of all devices is practically infeasible with cross-platform solutions. 15

The market today offers several different cross-platform frameworks. In the following sections, the main characteristics of Xamarin, ReactNative, Adobe PhoneGap, and Flutter are presented.

**2.1 Xamarin**

Xamarin is an open source platform from Microsoft for developing mobile applications targeting iOS and Android using C# and .NET. enabling developers to use base libraries for working with strings, dates, and files/IO. Xamarin has a big community with more than 60 000 contributors from more than 3 700 companies. (What is Xamarin, n.d.).

Developers must use Xamarin.Forms for building applications targeting both Android and iOS with a single shared codebase, where they can design and build mobile applications from a single API. Xamarin.Forms is an application framework for building mobile applications including cross-platform navigation, animation APIs, dependency service, messaging center, etc. With Xamarin.Forms the developers can access native user interface features for each platform such as iOS Safe Area and Android elevation. The Visual API gives the application a consistent Material Design look and feel across iOS and Android applications (Xamarin. Forms, n.d.).

Haberl (2015, p 49) concludes that since Xamarin implements a layer over the native environment, there is no real control of what is generated as the final code to be run on the device. This can lead to a lot of workarounds within the codebase.

**Pros:**

• Share more than 75% of the code across platforms (Manchada, 2019)

• Strong community with over 60 000 contributors and more than 3 700 companies (Manchada, 2019)

• A single tech stack for faster development using Visual Studio and C# (Leuschenko, 2018)

• Applications for all platforms where developers can create applications for mobile and desktop experiences simultaneously (Leuschenko, 2018)

**Cons:**

• License is free for individuals and startups but is expensive for enterprises with a $250 - $5 999 annual fee (Manchada, 2019)

• Not recommended for applications with heavy graphics (Manchada, 2019)

• Limited access to certain important libraries (Manchada, 2019)

• Slow debugging process, especially on Android (Manchada, 2019) 16

• Does not support all available 3rd-party libraries for Android and iOS without a specific wrapper (Leuschenko, 2018)

• Larger application size adding 2-5 Megabytes for the release and around 20 Megabytes for debug builds (Leushenko, 2018)

**2.2 React Native**

React Native is an open source framework released by Facebook in 2015, in 2018 it had the 2nd highest number of contributors for any repository on GitHub. The programming language is Java Script and renders the UI into native components giving applications developed in React Native a native look and feel. The UI is built with platform-agnostic components like View, Text, and Image that map directly to the platform’s native UI building blocks (React Native, n.d.).

**Pros:**

• Share more than 80% of the code across platforms (Manchada, 2019)

• Preview results right away and ready-to-apply elements (Manchada, 2019)

• Hot reloading features allow developers to see changes made within seconds (Manchada, 2019)

• UI rendering gives a highly responsive interface (Manchada, 2019)

• Reliable and stable applications (Ganguly, 2018)

• Ready-made components achieve simple functionalities making the development process faster (Ganguly, 2018)

•The growing community where developers can get help (Ganguly, 2018)

**Cons:**

• Not a fully cross-platform framework. To use native components there must be separate code for iOS and Android (Manchada, 2019)

• Since the framework is not built in conjunction with iOS or Android, it sometimes lags behind the native platforms (Manchada, 2019)

• Lacks consistency when it comes to releasing updates (Manchada, 2019)

• Slow debugging process, especially on Android (Manchada, 2019)

• Ready-made components collection is quite small limiting programmers and the component quality can lack in reliability when they are not official releases (Ganguly, 2018)

• Application performance lacks compared to the native application (Ganguly, 2018)

• JavaScript programming language making applications with low security and poor memory management (Ganguly, 2018) 17

**2.3 Adobe PhoneGap**

Adobe PhoneGap is an open-source framework built on top of Apache Cordova. Developers are using web technologies like HTML, CSS, and JavaScript to build mobile applications but still have access to native device APIs like cameras, GPS, and accelerometers (Adobe PhoneGap, n.d.).

**Pros:**

•It is possible to share the application with the team to get their feedback (Manchada, 2019)

• Cloud solution to create an application directly (Manchada, 2019)

• Third-party tools, a large community, and a large number of plugins (Manchada, 2019)

• Uses an intuitive desktop for mobile application development and serves the application created on the desktop to mobile devices connected to it (Manchada, 2019)

**Cons:**

• No recommended for high-performance applications and hardware-intensive applications due to their poor performance and lack of UI widgets (Manchada, 2019)

• Dependent on iOS SDKs to build an application that requires a Mac (Manchada, 2019)

•Low-performance applications compared to native applications (Manchada, 2019)

**2.4 Flutter**

Flutter (first released Dec 4th, 2018) is an open-source framework by Google for creating native compiled mobile applications that can be run on both iOS and Android. Flutter is using the object-oriented programming language Dart with widgets optimized for 2D graphics. Developers can use the same design on both iOS and Android. Flutter provides access to some native device APIs, like camera, geolocation, network, and storage but Flutter does not intend to build support for all native services and APIs (Flutter FAQ, n.d.). 18

**Pros:**

• Hot reloading features allow developers to see changes made within seconds (Manchada, 2019)

• Can build applications quickly that look native on both Android and iOS (Manchada, 2019)

• Based on Dart programming language that developers have found easy to acquire skills for (Manchada, 2019)

• Widgets as Google Material Design and Apple style with Cupertino pack (Manchada, 2019)

•It is high-speed animation and is simple to create and integrate into the application (Fitzgerald, 2019)

• Single language for layout and backend (Fitzgerald, 2019)

**Cons:**

• Flutter does not offer support for Android TV and Apple TV (Manchada, 2019)

• Lacks concerning native development (Manchada, 2019)

• Since Flutter applications is using built-in widgets instead of platform widgets the size of the application is usually bigger (Manchada, 2019)

• Limited libraries and often developers must build functionality themselves (Fitzgerald, 2019)

• Not an extensive or easy support for continuous integration (Fitzgerald, 2019)

**Existing decisional frameworks**

A **mobile development framework** is a software framework that is designed to support mobile app development. It is a software library that provides a fundamental structure to support the development of **applications** for a specific environment.

Modern-day mobile app development frameworks provide several built-in benefits, like speed, effectiveness, and a bug-free atmosphere. The framework uses pre-built elements, including compilers, debugging tools, and a toolkit, to display the application on the target device using the company’s code.

Using the expertise of a reputable mobile app development company may help the firm create mobile applications which are quick, reactive, and provide excellent user interactions. The use of mobile app development platforms speeds up and simplifies the app creation procedure.

Frameworks can be in three categories:

native frameworks for platform-specific development,

mobile web app frameworks, and

hybrid apps,

which combine the features of both native and mobile web app frameworks.

**Types of Mobile App Frameworks**

Android, iOS, as well as Windows, are just a few of the digital devices available. Mobile applications come in a variety of shapes and diverse sizes. We’re not speaking about specific app areas like purchasing or games. However, in terms of how applications work on a smartphone, mobile apps may be divided into three categories.

1. **NativeApps**  
   They are designed for specific operating systems such as Android, iOS, and Windows. Apps make use of device features such as RAM, camera, GPS, and so on.
2. **WebApps**  
   A Web app is a software that is kept in a distant location and distributed via the Web using a browser interface. Emails, online shopping sales, auction sites, weblogs, instant messaging apps, and other web apps are popular.
3. **HybridApps**  
   Hybrid apps are applications that are launched on a smartphone in the same way that every other application is. What sets them apart is that hybrids combine features from native apps and components from web apps.

The mentioned frameworks are commonly used to create these apps.

* + Native app framework
  + Hybrid app framework

Native apps are the highest-quality applications in respect of both aesthetics and functionality.

However, native application development is more expensive. Furthermore, achieving the highest customer engagement might be difficult due to the variety of platforms and devices available.

When financial constraints and a large number of customers are the primary goals, an application development firm may choose hybrid or cross-platform application development technologies.

**Mobile application design considerations.**

Today around 2 billion people are using smartphones. It is expected to reach around 4 billion by 2020. The number of mobile devices is more than PCs. Around 70% of the people spend time on apps and 80% of the people use mobile applications for purchasing, payment, and booking. Irrespective of any industry, mobile applications are proving to be an essential tool for business growth and reach. The opportunity is huge and the market potential is large. So, while developing an app for your business you should keep the above things in your mind.

This blog is more focused on key factors to consider before developing a mobile application for your business. **Mobile app** development is thriving like never before. With the invention and innovation of new mobile devices and OS’, it has now become imperative for every business to create mobile apps presenting their products and services. Though making a business app has become an easy task, if you are looking for your app as a tool for your business growth, then you need adequate strategies, effort, cost, and the right sort of marketing campaigns.

The operating context and physical characteristics of mobile devices demand careful coding and design. For example, streamlining code so that it executes as fast as possible is crucial. Code optimization can only go so far, of course; intelligent design that works within the device limitations can also help prevent your visual presentation from overtaxing the rendering system.

**Code**

While making your code run faster is always beneficial, the slower processor speed of most mobile devices increases the rewards of the time spent writing lean code. In addition, mobile devices almost always run on battery power. Achieving the same result with less work uses less battery power.

**Design**

Factors like the small screen size, the touch-screen interaction mode, and even the constantly changing environment of a mobile user must be considered when designing the user experience of your application.

**Code and design together**

If your application uses animation, then rendering optimization is very important. However, code optimization alone is often not enough. You must design the visual aspects of the application such that the code can render them efficiently.

*1. Research*

If you have a great idea to develop a world-class mobile application for your business, the first thing you need to understand is the market, customer demand, and trends. Even before giving a technological touch to your business app, you need to do market research. A market analysis report can give you very useful insights. You can understand the popularity of similar apps available in the market and your competitor strategies. This will allow you to optimize your app from the very beginning. Apogaeis believes in “Learning from others' mistakes than Learning after making mistakes.”

Customer reviews can provide you with a glimpse of customer likes and dislikes, choices, preferences, and future demands. You can take note of the pain areas and try to resolve those in your mobile app. Research can allow you to plan better and allow you to prepare a robust application from the very beginning.

*2. Identify Target Audience*

This comes under the research stage and is extremely important. Identifying the right set of audiences for your application is very important. The entire application future depends on these target sets of users, as they play a very significant role in application development, as well as app feature expansion and growth. Questions like who is going to use my application and how it can add value to their lives should be asked before starting development. If you meet the expectation of your users, definitely your app is going to get popular and it will help you to generate more revenue.

*3. Right Platform Selection*

One of the most important questions you need to ask yourself is, on which platform you want to deploy your business app. It is advisable to start with one platform. Be a master of a single platform, later you can move on to other platforms. iOS, Android, and Windows are the most popular mobile app platforms.

To decide the best-suited platform for your mobile application, you need to keep certain things in your mind like app brand, target audience, app features, and most importantly pricing strategy. After that, you need to choose a development methodology for your business app; Native, Mobile Web, or Hybrid. Native applications have many advantages, but they may be slightly expensive.

*4. Set Plan of Action*

Before starting to build your business mobile application, make sure you understand the entire process. A robust business app takes a decent amount of time and effort. It also passes through various stages. The business owner should understand the importance of all the stages, before going ahead. Project Management, App Design, App Architecture, App Development Methodology, App Testing, Enhancement, and finally App Deployment- These are the basic stages of any mobile application. Set a plan of action for your app development process. Monitoring and controlling is very much essential. Release a beta version of your app first, and do rigorous testing and review before releasing the full version to the end users.

*5. Know your Budget*

Developing a business mobile app requires investment along with knowledge and strategy. Knowing your budget and allocating it properly during each stage of app development is essential. There are various aspects of an app that requires money to be invested in; app development, maintenance, updating, marketing, etc. Your app budget depends on your niche. The type of app you want to create and the type of content you wish to add to your app.

*6. Think Out of the Box*

Always keep one thing in mind. People want something different. Why did they choose one brand ahead of another similar brand? It is only because they feel the chosen brand has something unique to offer. Mobile applications are the best way to engage your customers and probably the easiest way to convert a lead into a potential buyer. Human beings tend to get bored of one thing very quickly. As there are millions of smartphone applications available in the market, app users want to be served something new. Hence, it is recommended to think out of the box strategies or features and don’t let your users choose someone else.

*7. Smooth and Efficient*

No excuse, your business mobile app must be smooth and efficient. If your app is taking a lot of time to load, it may prove to be fatal. Either user uninstalls the app after first use or rarely opens it. It adds to create a negative impression in the user’s mind. Make sure your app does not consume a lot of memory space and processing power on a mobile device.

Efficiency is something users expect from all mobile applications. It includes user-friendliness, data efficiency, battery usage, security, and the like. If your app is using a lot of 3G or 4G data, it might not work for you in the longer run. Users may download and forget. So, don’t fall into that category and develop your app in such a manner that it won’t consume more data from your users and does not drain the battery.

*8. User Experience*

User Experience is the backbone of any application. If you failed to provide your users with an excellent experience, then there is no reason for them to come back to you and use your app. User experience is an increasingly crucial feature when it comes to the digital landscape. An app defines how a user feels and thinks about your business and services. It is about making something valuable, easy to use, and effective for your target audience. Don’t make your app confusing or complicated.



A well thought mobile application offering a seamless user experience can translate into several key benefits. The functionalities and content that are delivered to your app must be in line with your target audience. The end product must provide superior quality customer service. If you are unable to provide a quality mobile application, it may harm your brand image.

*9. Focus on Marketing Strategy*

There is no meaning in building an app for your business if you are not focusing on, how to market it and make it available to potential users. Creating a buzz, before the launch can promote your app in such a way that your app can get a lot of reach from the word go. Most business owners fail in executing their marketing strategies for their mobile applications.

Different Mobile Applications have different targeted users. Similarly, you have to analyze your industry and your potential user base. After analyzing that only, you can make strategies on how to promote it. Online campaigns are one of the preferable options for many business owners. But your marketing strategy depends largely on your industry, offerings, and your potential users.

*10. Testing*

It is important to test your mobile application before you launch. Before users get an experience, you should make sure it can fulfill their requirements. Smooth, efficient, and high-performance are a few of the things you should keep in mind while testing. Testing will allow you to dust out all the nooks and crannies and make your app a strong business tool that can connect your business with your customers.