**UNIVERSITY OF BUEA**

**FACULTY OF ENGINEERING AND TECHNOLOGY**

****

**DEPARTMENT OF COMPUTER ENGINEERING (FET)**

**CEF 440: INTERNET PROGRAMMING (J2EE) AND MOBILE PROGRAMMING**

**2022-2023**

**WORD DOCUMENT REPORT**

**NAMES OF GROUP MEMBERS**

|  |  |
| --- | --- |
| 1. AMBE MBONG-NWI NCHANG | FE20A007 |
| 1. ASONGNA FRANK TONGWA | FE20A012 |
| 1. BESINGI NAURA MABOLA | FE20A023 |
| 1. FONGE BERTIN AMIN-SHU | FE20A040 |
| 1. NKWETAKEM TABO BRUNO | FE20A091 |

**Course Instructor**

DR. VALERY NKEMENI

**OUTLINE:**

**Questions:**

1. What are the major types of mobile applications that exist and what are their differences?

* Answer to question

1. Give a thorough review of programming languages used for mobile programming?

* Answer to question

1. What is a mobile-app development framework? Review these frameworks considering the features: Language, performance, cost, user experience, community support, complexity.

* Answer to question

1. How do you Collect and Analyze the Requirements of a Mobile App to be developed?

* Answer to question

1. How do you estimate the mobile-app development cost:

* Answer to question

**Week One Questions and Answers**

**Question 1:** What are the major types of mobile applications that exist and what are their differences? **Answers:**

Mobile apps can generally be classified into four types based on their functionality and purpose below is listed the 4 main types of mobile apps.

1. **Native Apps**: These are mobile applications that are specifically designed and developed for a particular mobile platform, such as iOS or Android, and can be downloaded from the platform's app store. Native apps are optimized for the platform they are built on and can access device features, such as GPS, camera, and contacts. They typically offer the best performance and user experience.
2. **Web Apps**: These are mobile applications that are accessed through a mobile browser and are built using web technologies like HTML, CSS, and JavaScript. Web apps are not installed on the device, and users can access them using a URL. They can work across different platforms and do not require any installation, but may have limited functionality compared to native apps.
3. **Hybrid Apps**: These are mobile applications that combine the features of native and web apps. They are built using web technologies but are packaged as native apps and can be downloaded from app stores. Hybrid apps can access device features and offer better performance than web apps, but may not be as optimized as native apps.
4. **Progressive Web Apps (PWA)**: These are web applications that use modern web technologies to provide an app-like experience. PWAs can be accessed through a mobile browser and can be installed on the user's home screen. They offer features like push notifications, offline mode, and access to device features, making them a good compromise between web and native apps.

Each type of mobile app has its own strengths and weaknesses, and choosing the right type of app depends on the specific requirements and goals of the project.

Some apps, like Instagram, have versions that are available as both native apps and hybrid apps. The specific type of app that an organization chooses to build depends on their goals, target audience, and technical requirements.

Below is a table of differences between the different mobile app types:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Native apps** | **Web apps** | **Hybrid apps** | **Progressive web apps** |
| **Installed** | Yes | No | Yes | Optional |
| **Platform** | Specific | Cross-platform | Cross-platform | Cross-platform |
| **Access to device features** | Yes | Limited | Yes | Yes |
| **Development** | Requires platform-specific tools and languages | Uses web technologies like HTML, CSS and Javascript | Uses web technologies but packaged as native apps | Uses web technologies but optimized for app like experience |
| **App store Presence** | Available in app stores | Not available in app stores | Available in app stores | Optional to publish in app store |
| **Offline Access** | Possible, but requires local storage | Limited | Possible, but requires local storage | Yes |
| **Updates** | Must be downloaded and installed | Automatically updated on the server | Can be downloaded and installed | Automatically updated on the server |
| **Examples** | Instagram (iOS and Android)  Google Maps (iOS and Android)  Snapchat (iOS and Android)  Spotify (iOS and Android)  Twitter (iOS and Android) | Google Docs (accessed through a mobile browser)  Trello (accessed through a mobile browser)  Pinterest (accessed through a mobile browser)  Slack, Airtable (accessed through a mobile browser) | Uber (iOS and Android)  Yelp (iOS and Android)  Basecamp (iOS and Android)  Instagram (their Android app is a hybrid app) | Starbucks (can be added to home screen on iOS and Android)  Twitter Lite (can be added to home screen on iOS and Android)  Trivago (can be added to home screen on iOS and Android)  Pinterest, Flipkart (can be added to home screen on iOS and Android) |

**Question 2:** Give a thorough review of programming languages used for mobile programming?

1. **Java**

Since 2008, Java has been the choice of choice for creating applications on the Android mobile platform because it can run on a Java Virtual Machine (JVM). Most Android secondary languages are considered a version of Java or a different presentation of Java, this does not mean that this language is limited only to this platform, in fact Java is considered a multipurpose language.

### **Advantages**

* + Java is a high-level programming language that is easy to learn and use.
  + Java is platform-independent, meaning that it can run on any operating system.
  + Java is object-oriented, which makes it easier to write complex programs.
  + Java has automatic garbage collection, which means that it frees up memory automatically when it is no longer needed.
  + Java has built-in support for multithreading, which allows multiple threads to run simultaneously.
  + Extensive library available online.

### **Disadvantages**

* Java can be slow compared to other programming languages like C and C++.
* Java requires more memory than other programming languages.
* Java can be complex to write and debug.
* Java has poor support for creating graphical user interfaces (GUIs).
* Java requires a paid commercial license for use in some cases.

1. **Swift**

If you want to create applications for the iOS platform, a programming language created by Apple, it is the most suitable. Swift is a multipurpose and highly readable language on which you can program without major difficulties, the opposite of its predecessor Objective C, which was replaced by Swift as the official language for developing applications for Apple devices.

**Advantages**

* Code is easy to read.
* Interactive nature that saves time in development and compilation.
* Its Xcode IDE is self-contained and if a library is included it simplifies the use of workspaces.
* With Swift you can also develop server-side.
* Swift is a clean language with simple syntax that can be read and written easily.
* Swift is faster than Objective-C and Python.
* Swift provides safeguards to prevent errors and improve readability.
* Swift has better optimization and error handling mechanisms than Objective-C.

**Disadvantages**

* The learning curve in Swift is higher with respect to other languages for iOS.
* Its type checking is very strict and type errors will not allow compilations to be made.
* It is relatively young and therefore not best suited for very robust applications.
* It is somewhat unstable and requires periodic updates.
* Swift has a smaller community than other programming languages like Java and C++.
* Swift has less third-party support than other programming languages like Java and C++.

1. **Kotlin**

Kotlin is a relatively new programming language that has managed in a short time to become the preferred language for application development on the Android platform. This has been created by Google and is a great alternative to Java. It is now like an official language for Android applications.

### **Advantages**

* It delivers native level support while running on various platforms.
* It allows to execute functional programming through lambdas.
* Kotlin allows faster compilation and lightweight app development.
* Kotlin code is much smaller in size compared to Java.
* Kotlin allows compiling to bytecode for executing in the JVM. This ensures easy interoperability with Java code.
* Kotlin has an intuitive and concise syntax that makes it a more efficient, compact, and clear language.

### **Disadvantages**

* It has been said that Kotlin will become the successor of Java but for now it is far behind because its compilation times are two to three times slower than Java.
* Its support and community have not been fully developed so the resources to solve problems are quite limited.
* Kotlin is a relatively new language compared to other programming languages like Java and C++.

1. **Objective-C**

Generally, Objective-C is a general-purpose, object-oriented programming language that adds Smalltalk-style messaging to the C programming language. This language features dynamic typing, partial typing, named parameters for methods and function calls, and numerous data types including objects, structs, arrays, strings and others.

**Advantages**

* Objective-C is a powerful programming language that was developed to work on both Apple and Android devices.
* Unlike Swift, Objective-C does not have a runtime library, meaning that the software has to be compiled before it can run. Because of this, Objective-C is considered a lower-level programming language and thus has faster compilation time than Swift.
* Objective-C is object-oriented and supports procedural programming as well as functional programming.
* It also allows for inheritance from parent classes, which makes it easier to build upon other people’s work without having to start from scratch.
* Objective-C has a large and well-established library of frameworks and tools.
* Objective-C is stable and reliable, and does not change as often as Swift.

**Disadvantages**

* Objective-C is a difficult language to learn. In order to become an effective programmer in the language, you have to know a lot of different concepts and commands.
* You can’t do as much with Objective-C as you can with other languages. There are not a lot of options for developers so if you want your app idea to be successful, it has to be simple enough.
* Some say that it’s not as efficient as other languages. For example, Java is more efficient because of how it handles memory allocation.

1. **C#**

It’s one of the most common programming languages in the world, and it’s used by Microsoft. You can use C# to write programs for all Windows devices as well as for apps on Windows Phone and Xbox One. C# is a general-purpose object-oriented programming language created by Microsoft designed for building a variety of different apps. It was released in 2000 as a successor to another Microsoft language called C++. It’s easy to learn, has an efficient syntax, and is open source then the C# programming language might be a good choice.

**Advantages**

* Powerful object-oriented programming capabilities
* Support for multiple languages
* Support for libraries of code written in other languages, such as C++ and Visual Basic .NET
* Support for multitasking and multithreading
* C# becomes a flexible language that can be used on cross-platform thanks to the wide features of .NET.
* C# has an inbuilt garbage collector. It is a memory manager that keeps track of unused items and automatically releases memory.

**Disadvantages**

* It lacks a large community of developers and programmers who can help enhance the code or provide support.
* C# is not as popular as other programming languages like Python and JavaScript.

1. **JavaScript**

Qualified as the most popular programming language today, JavaScript has demonstrated great versatility by allowing the development of web, mobile and cross-platform applications thanks to frameworks such as React. It is now possible to run it outside of browsers, as well as compile from other languages, making it a true full-stack development language.

### **Advantages**

* Friendly and pleasant syntax for development.
* Easy to control.
* Versatile and flexible thanks to Node.js.
* Fast and lightweight.
* JavaScript is executed on the client-side that’s why it is very fast.
* JavaScript is easy to learn. Anyone who has basic knowledge of programming can easily learn JavaScript.

### **Disadvantages**

* Produces quite a lot of JS code fragments.
* Can be easily broken.
* Decreases execution performance of other software.
* Its semantics can sometimes be inconsistent.
* JavaScript can be slow when it requires outside resources or calls to a backend server.
* JavaScript can be vulnerable to security threats like cross-site scripting (XSS) attacks.

1. **Python**

Like others, Python is a great language for app development because it’s both easy to read and learn. It is also very popular among developers, which means there are plenty of resources available to help you out as you develop your skills.

**Advantages**

* Python is beginner-friendly. It has a clear and easy-to-use syntax that makes it easy for beginners to learn.
* The language is interpreted rather than compiled, which means it executes one line at a time rather than all at once.
* Python has a large community of developers who come together on the same platform to help one another and bring the language forward.
* Python is versatile. It can be used in a wide range of applications.

**Disadvantages**

* Python can be slow when it requires outside resources or calls to a backend server.
* Python is not as fast as other programming languages like C++ or Java.

1. **Dart**

Dart is an open-source, object-oriented, general-purpose programming language developed by Google in 2011. Dart uses a ‘C’ style syntax and optionally trans compiles it into JavaScript. It is used for both client-side and server-side web development. Dart is also being used for Native and Cross-platform mobile development.

**Advantages**

* Easy to learn when used to C-style syntax.
* Code can be shared across web and mobile apps.
* Super friendly and helpful community.
* Dart has high performance
* The first advantage is that it is easy to learn.
* The second thing is about its availability of documentation.
* Programs written in Dart tend to run faster than programs created in JavaScript.
* Dart is very stable and it can be used to build production quality real-time applications

**Disadvantages**

* Backend (API/Server) focus is currently very low.
* Dart has very limited resources online and it’s hard to find solutions to problems.
* Dart is fairly new to the programmers and rarely used in the market

**Question 3:** What is a mobile-app development framework? Review these frameworks considering the features: Language, performance, cost, user experience, community support, complexity.

**Answers:**

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

**Types of Mobile Applications:**

1. **Native Applications:** Applications designed to run on a particular platform like android, windows or iOS. The aim here is best performance accomplished by incorporating best UI modules leading to nice User Interface. They can be used offline. Example: facebook, whatsapp.
2. **Web-based Applications:** These applications are written in CSS, HTML and Javascript. They require less memory usage because data is stored on internet servers. Here, you are required to use internet connectivity. Example: Google map.
3. **Hybrid Applications:** This is a combination of Native and Web-based applications. Example: Twitter, Instagram.

**Top Ten Mobile-Application Development Frameworks:**

1. **React Native (Hybrid):** Best open-source **javascript** (base language) library to build native applications for all devices and platforms. It was introduced by facebook in 2018 and also used in instagram.

**Features**:

* Write once and use everywhere (code written can be used on any platform to build applications)
* Uses Javascript which has a wide-spread usage and is also very popular
* Easy to use if you know javascript.
* Live reload (modify code and see modifications simultaneously)
* Strong community support
* Compatible plugins third party packages
* Cost efficient testing (provides few unit testing functionalities via javascript frameworks and other third- party testing tools).
* Supports IOS and android
* It relies heavily on third party applications and plugins

1. **Flutter (Hybrid):** it’s a **UI toolkit by Google**. Helps in building applications for web, mobile and desktop for iOS and Android. Project built here has a **single codebase** due to the fact that flutter was built upon the **DOS language** which uses object-oriented programming.

**Features:**

* Excellent Documentation (hence easy to use as all steps are present for installation and learning)
* Uses Dart Framework (has most components inbuilt hence needs no bridge to communicate with native modules. The C++ engine here provides everything you need)
* Rich flutter API
* Facilitates testing as it comes with testing modules which is also explained in the documentation.
* Strong Command Line Application.
* Built-in material design
* Supports both IOS and android
* Strong widget support

1. **Xamarin (Hybrid):** It is a cross-platform and open-source app building platform built on **.NET** and **C#** and was introduced by Microsoft. It can be used to create applications for iOS, Android and Windows.

**Features:**

* Allows invoking libraries from various languages like java, C++, objective C (flexible)
* Modern language constructs (C# is a modern language which has dynamic language features and generics)
* Diagnostic tools and Modern IDE like Visual Studio that provides autocompletion, libraries etc
* Mobile cross-platform support (Android, iOS and Windows)
* A strong community of 60,000 contributors and ability to share large amounts of code.

1. **Iconic (Hybrid):** It helps you build hybrid, desktop and interactive web apps along with cross-platform apps. It covers building apps for the web, android and iOS. It supports **Angular,** **React** and **Vue JS.**

**Features:**

* Complete app development kit which has rich libraries (good frontend building blocks)
* Community support of about five million developers.
* Iconic CLI. This command line interface has commands that greatly help developers.
* Web components (set of web platform application program interfaces APIs). They create customized and reusable HTML tags leading to reduce start-up and loading time.
* Faster development time
* Open-source

1. **PhoneGap (Hybrid):** It is owned by Adobe systems**.** It allows you build mobile applications using **HTML**, **CSS3** and **Javascript.**

**Features:**

* Free and open-source
* Wonderful documentation and tutorial on usage
* Easy to use if you know HTML, CSS and Javascript
* Robust backend hence developers have little work to do
* Taps on the device’s hardware. (that is use camera, location)

1. **Corona (Hybrid):** It is also called **Solar2D** or **2D game engine**. It’s a cross-platform and open-source framework good for building games and apps. It supports amazon, iphone, windows, mac, android, apple TV etc. It provides **C++,** **objective C** and **Java extension**.

**Features:**

* Build once and use everywhere because it can be used to create mobile and desktop applications
* Use of Lua (open-source scripting language which is fast and lightweight).
* Multiple plugins for all needs
* Strong API support to integrate with any native library
* Faster development process
* Free and Open-source.
* 500,000 plus developers.

1. **JQuery Mobile (Hybrid)**: It is **touch-optimised** and used to build highly responsive apps for smartphones, tablets, and desktops. User interface system is based on **HTML5**

It enables u create highly branded websites.

**Features:**

* Compatible with all modern browsers and platforms like Android, iOS.
* Lightweight size hence optimizing speed
* Touch-optimized widgets
* Built on jquery core so easy to learn for those that know jquery
* Automatic initialization
* Simple API
* Powerful theming framework.

1. **NativeScript (Native):** It’s an open-source framework to create native mobile apps empowered with: **Angular**, **Javascript**, **CSS**, **Typescript,** and **Vue.js.**

Advantage: it reduces the code and time of the app loads on the system. It has a strong web-empowerment platform.

**Features:**

* Can use any view as the root of the application
* Allows more than one frame as it important for a good mobile GUI.
* Templates can be updated to be compatible with Angular, Typescript
* Provides LiveSync an Webpack which can be used simultaneously.

1. **Appcelerator Titanium (Hybrid):** Makes use of **Angular** and **Bootstrap** to build **HTML5** hybrid mobile applications. It is open-source. It makes use of a single Javascript codebase.

**Features:**

* Code reusability (reuse about 60-90% of code across different platforms).
* Bi-directional updates resulting in seamless movement between design and code
* Combines and normalizes data from multiple sources, optimize payload size and convert to mobile form.
* Seamless integration
* Drag-and-drop palette

1. **Mobile Angular UI(Hybrid)**: Fusion of **HTML5, Bootstrap and Angular JS,** to build hybrid mobile applications.

**Features:**

* Excellent Documentation which has directives on how to easily use the application.
* No JQuery dependencies because you can use Angular
* Build fully responsive interfaces with a super-small CSS file due to Bootstrap
* Provides mobile components not present in bootstrap like switches, scrollbars.
* Easy learning curve if you already know bootstrap and angular JS.

**Question 4**: How do you Collect and Analyze the Requirements of a Mobile App to be developed?

**Answers:**

To be able to create a good mobile app based on a business idea, a document called a Mobile App Requirement Document is prepare which clearly states the features and main components of the app to be developed, its limitations, who is to use the app, the goals and functionalities of the app. This document helps in clarification and avoiding confusion during development.

The main components of this document are:

1. Business goals and App Objectives (describing app idea)

This part elaborates on:

* The main problem the app will be solving or the main purpose of the app
* The unique selling point of the app that makes it different from its counterparts
* The benefits the app will bring

1. Sequence of navigation or user flow

It describes the navigation pattern of the app. That is it describes the sequence of activities as the user explores the app.

It tells how the user moves from maybe the on-boarding screen to the registration screen, to the home screen etc.

1. Referring existing applications

Going through already existing applications so as to know exactly what is required for an app like the one you are to build and also to direct the user on what their expectations will look like in the end

1. App features (functional and non-functional requirements)

It states the various features of the app and the clear description of their functionality. It also states the non-functional requirements such as the performance, security, compatibility

1. App UX

Bringing out a general overview of the user experience will be so as to provide bases for future designs.

1. App Technology and infrastructure requirements

Knowing the exact environment where the app will be hosted be it on Android or iOS. it also precises if the app requires any device hardware such as the microphone, camera.

1. Creating wire-frames

Bringing out a demo design of how the app will look like

1. Choosing a requirement format

Select a suitable format for your applications. The various formats are listed as follows.

* Functional specification document (FSD) format

It covers all requirements of the application where the app’s features are detailly explained

* User stories

They are almost like the FSD but much more precised. They enumerate the things the user can do in the app. They help to identify which feature are critical and which can be dropped out during the release of a later version

* Sketches and wire-frames

They are in the forms of visual representations of the requirements and give clarity on the scope of wok

* Mixed format

It contains the both aforementioned formats and is the most recommended.

**Question 5:** How do you estimate the mobile-app development cost:

**Answer:**

The final cost of an application depends on dozens of factors, but if we try to summarize, we may picture the cost based on certain criteria according to GolodForm’s extensive survey. This cost is calculated as

cost = Total development time x Hourly rate.

However, due to the nature of the apps to be built this price ranges and time ranges might differ due to the various factors that constitute the nature of the app.

The estimated time to build a simple app with features such as user interface, user registration and login is between 4-11 weeks with a cost of 500,000frs to 2,500,000frs.

Example of simple apps are calculator App, flashlight etc

The estimated time to build a complex app with features such as third-party integration, real time updates, push notifications and advanced user interface is between 7-16 weeks with a cost of 2,500,000 to 5,000,000frs. Example of complex apps are social media Apps, E-commerce apps, gaming apps, etc

The estimated time to build an advanced app with features such as artificial intelligence, virtual reality, machine learning is between 3-6 months with a cost of 5,000,000 to 15,000,000frs. Example of advanced apps are health care apps, financial apps, enterprise level etc

**Factors influencing the cost to develop and app**

**In-house vs outsourcing app development:**

in-house refers to hiring a team while outsourcing refers to giving the project to an app development agency.

**Pros and cons of in-house app development**

**pros**

* High level of control
* Face to face meetings

**cons**

* Lack of talent pool
* Dealing with hiring and bureaucracy0

**Pros and cons of outsourcing app development**

**pros**

* High availability of talent pool
* Lower rates
* Flexible

**cons**

* Time zone differences
* lack of personal control

Overall, in most cases, outsourcing is cheaper in cost than in-house building

**Native vs Hybrid**

in terms of cost, native app development is more costly than hybrid app development, since it requires to be built on different platforms

**Android or Ios**

Though they are relatively same in cost and time, iOS is slightly costly than android

**Visual design customization**

This refers to the design of the various screens which can vary in cost depending the number of screens, the complexity of the designs etc

**App maintenance cost**

it costs approximately 200,000frs on average to maintain an application annually

**Features and functionalities**

The various features and functionalities of an app influences the cos of an app. Some of these features could be

**Hardware features**

certain native features of an application are to be taken into consideration during development based on the nature of the application being developed. Such features could be internet access, Bluetooth, and has a time cost of approximately 30-100hours to be implemented

**user login**

it has atime cost of approximately 70hours to be implemented

**user engagement**

Features such as social media sharing, push notifications etc can have atime cost of approximately 25-45hours to be implemented

**Geo-localizations**

features requiring maps, locations are common in apps and has time cost of approximately 10-50hours to be implemented

**Payments**

based on the different payment services used and their customization,it has time cost of approximately 20-200hours to be implemented

**Cross device synchronization**

This refers to synchronizing data between different devices such as phone and tablets .it has time cost of approximately 20-200hours to be implemented

**Data encryption**

it has time cost of approximately 10-80hours to be implemented