



CHANGE THIS TO THE TITLE OF YOUR
PROJECT
AND CHECK THE SPACING

**Final Year Project
B.Sc.(Hons) in Software Development**

BY
DARREN REGAN

APRIL 24, 2020

Advised by Martin Hynes
DEPARTMENT OF COMPUTER SCIENCE AND APPLIED PHYSICS
GALWAY-MAYO INSTITUTE OF TECHNOLOGY (GMIT)

Contents

1	Introduction	3
1.0.1	Objectives for project	4
1.0.2	Sections	5
2	Methodology	7
2.1	Research Methodology	8
2.2	Software Development Methodology	8
2.3	Project management	8
2.4	Development tools	8
2.5	Source Control	8
3	Technology Review	9
3.1	Overview	9
3.2	Main Technologies	10
3.2.1	Kotlin	10
3.2.2	Kotlin vs Java	12
3.2.3	Firebase	13
3.2.4	Picasso	13
3.2.5	Native Applications	13
3.2.6	Hybrid Applications	14
3.2.7	Hybrid vs Native Applications	14
4	System Design	17
5	System Evaluation	19
6	Conclusion	21

CONTENTS

List of Figures

1.1 The image caption should be succinct but descriptive. 6

3.1 Hybrid Application 15

Chapter 1

Introduction

My project is a Native Android E-Commerce Application built in Android Studio using Kotlin/Java programming languages. The general concept of my application is to create a e-commerce app that mirrors Amazon or Alibaba.

A specific goal of this project was to learn how native development differs from hybrid development(Xamarin, Ionic, React Native etc). Hybrid development was a topic that had multiple course projects, but i never got to experience building a project based on native development in Android or iOS.

A focus i made on my application is integrating different technologies such as Firebase, Picasso and Retrofit. Using these technologies made certain aspects of my application easier to implement. For example Picasso which is a image downloading and caching library can manage many aspects of image processing in an android environment such as handling ImageView recycling, complex image transformations with minimal memory use and automatic memory and disk caching which all have a heavy amount of coding involved to be implemented manually.

Firebase is another technology i use which has a great number of features and products such as Cloud Firestore which stores and syncs data between users and devices - at global scale - using a cloud-hosted, NoSQL database.

Authentication to manage users in a secure way by offering authentication through email, password and third-party providers like GitHub, Google, Facebook and Twitter.

Realtime Database which is an efficient, low-latency solution for mobile apps that require synced states across clients in realtime.

1.0.1 Objectives for project

Use a new programming language and framework

One of the objectives i had set for this project is to use a new language and framework, i made the choice of android and kotlin as it met all my objectives for my project. The main goal of my project was to build a native mobile application and compare it to hybrid applications that i have built throughout my course.

Create a Native Mobile Application

Use new and useful technologies ss

Include social Media Integration ss

Create a Native Mobile Application

1.0.2 Sections

Methodology After setting objectives for my project, i set out to make a plan for development using Kanban boards. I used an agile approach for my development setting specific goals for each week. Validation and Testing was done using Junit.

Technology Review In this section i explain the Technologies used in my project, technologies such as Firebase, Kotlin, Android Studio are explained in great detail. I will also review programming in android with kotlin compared to my experience through out my four years in college. Comparisons such as Native development vs Hybrid development, use cases for different programming languages, language features such as more use of lambda expressions and more Kotlin features that make it a great alternative to java for android development.

System Design In this section i provided an explanation of the overall system architecture using UML class, sequence and interaction diagrams as well as screenshots of UI components such as how each view is formed with ImageView, TextBox etc.

System Evaluation In this section i evaluate my project against the objectives i set out.

Conclusion

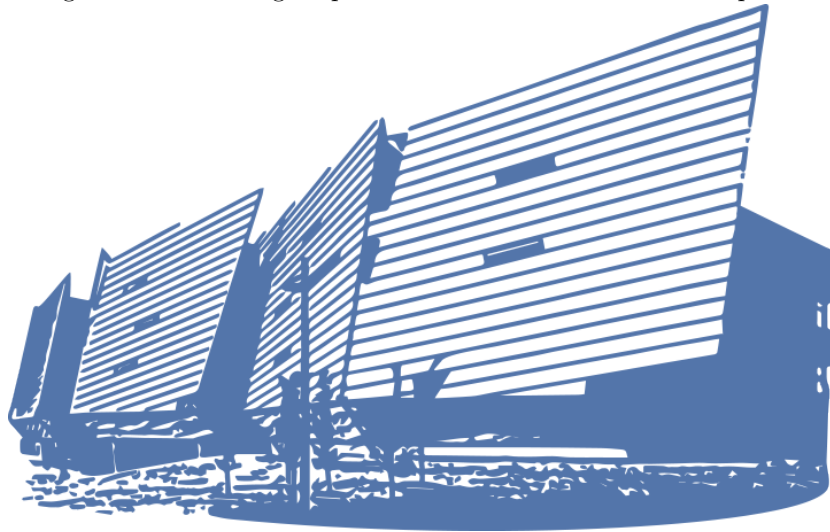
In my conclusion i evaluate my overall goals for my project and list outcomes of the project. I go over discoveries made, what I've learned and what i can improve on in the future

Github Repository <https://github.com/DarrenRegan/Final-Year-Project>

My GitHub Repository contains my Dissertation, APK file which are both available for download with a click of a button at the top of the README.

- The README contains a quick explanation of the project as well as an installation guide, Devices used in testing as well as resources used along with links to research material.
- The code for my project is located at Final-Year-Project/app/src/main/ Java folder contains the code for activities and models
- Res folder contains the code for UI elements of the project, main/res/layout contains XML for all activities

Figure 1.1: The image caption should be succinct but descriptive.



Chapter 2

Methodology

This chapter covers the various methodologies that were implemented in this project, this includes Research methodologies, Software development methodologies, project management, supervisor meetings, developments tools, testing and source control.

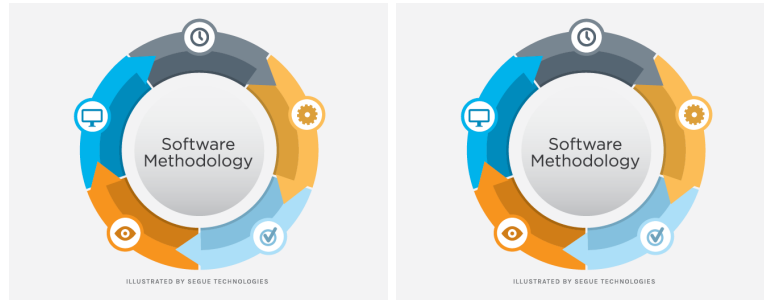
An overview of methodologies used in this project; **Research methodologies** which includes Mixed Methods, Quantitative Research and Qualitative Research. **Software development methodologies** which includes Agile Development, Continuous Delivery, Test Driven Development, Feature Driven Development, Extreme Programming etc. **Project management** which includes GitHub Kanban board and supervisor meetings. **Development tools** Android studio, Firebase. **Source Control** GitHub, Overleaf, Badge/Shield

2.1 Research Methodology

The research methodology that was used in this project was a Mixed Methods Research methodology, using a mix of both Qualitative Research and Quantitative Research. Qualitative research approaches are employed across many academic disciplines and is useful at an individual level. Qualitative data collection methods vary using unstructured or semi-structured techniques.

Various data collection tools were used for gathering data such as

2.2 Software Development Methodology



Explain Agile Development, Continuous Delivery, Test Driven Development, Feature Driven Development, Extreme Programming etc. How i implemented them in my project

2.3 Project management

Explain GitHub Kanban board and supervisor meetings. How i used both to improve my project

2.4 Development tools

Explain Android Studio, Firebase. Compare Android studio to IntelliJ

2.5 Source Control

Explain GitHub, Overleaf, Badge/Shield.

Chapter 3

Technology Review

This chapter discusses the different technologies used in throughout the project. It discusses the the advantages and disadvantages of each technology and why certain technologies were used over others. It also discusses hybrid applications compared to native applications, advantages, disadvantages, uses at different business structures and other topics.

3.1 Overview

This project is a Native android app built with Kotlin in Android Studio, Fire-base is used for the database, statistics, verification etc.

Topics:

- Kotlin/Java comparison
- Fire-base
- Picasso
- Android Studio/IntelliJ
- Native Applications
- Hybrid Applications
- Hybrid vs Native comparison

3.2 Main Technologies

This section will discuss the main technologies currently in use in the android application.

3.2.1 Kotlin



Kotlin is a cross-platform, statically typed, general-purpose programming language with type inference. Kotlin is designed to inter-operate fully with Java, and the JVM version of its standard library depends on the Java Class Library, but type inference allows its syntax to be more concise. Kotlin mainly targets the JVM, but also compiles to JavaScript or native code (via LLVM). Language development costs are borne by JetBrains, while the Kotlin Foundation protects the Kotlin trademark.

Kotlin is the preferred language for Android app developers as of May 2019, since the release of Android Studio 3.0 in October 2017, Kotlin has been included as an alternative to the standard Java compiler. The Android Kotlin compiler targets Java 6 by default, and lets programmers choose between Java 8 to 14 for optimization purposes.

Kotlin originated at JetBrains, which is the company behind IntelliJ IDEA. Kotlin has been open source since 2012 and has a large team of full-time developers working on it, there is also the Kotlin project of GitHub which has more than 370 contributors.

Advantages

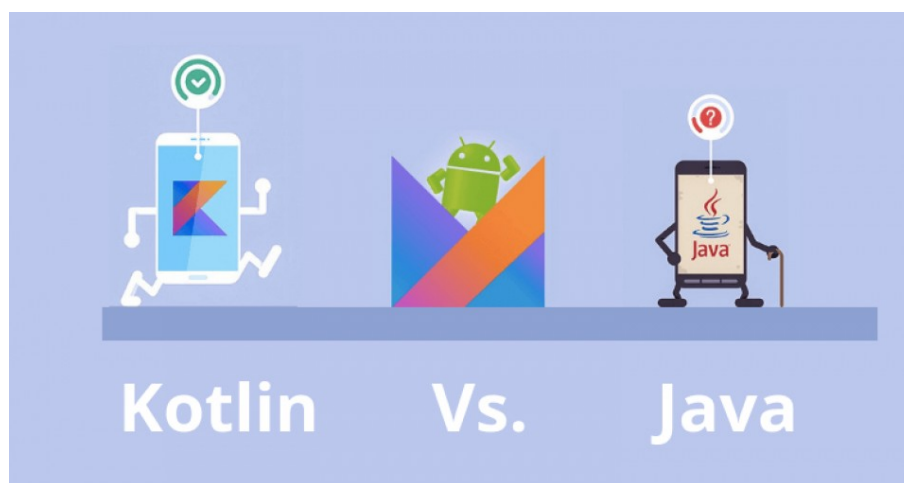
Kotlin has many advantages, many are quite serious improvements in readability and workflow which was noticeable when creating my project

- **Less code combined with greater readability** - Spend less time writing code and working to understand the code of others.
- **Mature language and environment** - Kotlin has developed continuously over the years not only as a language but as a whole ecosystem with very robust tooling. Its seamless integration with Android Studio, makes it actively used by companies to develop Android applications.
- **Kotlin support of Android Jetpack and other libraries** - KTX extensions adds kotlin language features, such as coroutines, extension functions, lambdas, and named parameters, to existing Android libraries.
- **Interoperability with Java** - You can use Kotlin along with the Java programming language in your applications without needing to migrate all your code to Kotlin.
- **Support for multi-platform development** - You can use Kotlin for developing not only Android but also iOS, back-end, and web applications by sharing the common code among the platforms.
- **Code safety** - Less code and better readability lead to fewer errors. The Kotlin compiler detects the remaining errors, making the code safe.
- **Easy to Learn** - Kotlin is very easy to learn, especially for any Java experienced developers.
- **Large community** - Kotlin a great support and many contributions from the community, which is growing all over the world. According to Google, over 60% of the top 100 apps on the Google Play Store use Kotlin. Many startups and Fortune 500 companies have already developed Android applications using Kotlin and more and more companies are prioritizing Kotlin Native application development over other options due to the robust toolkit and optimizations that make your applications the best that they can be.

Disadvantages

- **Shift from Java to Kotlin** - Kotlin is an amazing programming language and there is a reason why leading lead companies have started using kotlin, but at their core their two different languages. Developers won't be able to quickly shift from one to another without taking time to learn Kotlin. Therefore company's have to consider different approaches to Android app development as additional expenses are required on training a team of developers.
- **Hard to find experienced developers** - There is a high demand for specialists in Kotlin as Google made it the preferred language for Android development in 2019, but there is still a very large amount of Java programmers on the market compared to Kotlin developers. This means on average the Kotlin developers may be younger meaning less senior developers available for hire. This is quite a large disadvantage, but will quickly fade away as many leading tech companies have switched which creates a ripple effect down the chain of companies.
- **Limited learning resources** - Although the number of Android app developers who use Kotlin instead of Java increase everyday, there is still a limited number of resources in the market compared to Java. Many College courses will teach Java over Kotlin as both are so similar, meaning most Kotlin developers come from a background in Java and learn to code in Kotlin themselves.

3.2.2 Kotlin vs Java



3.2.3 Firebase



3.2.4 Picasso



3.2.5 Native Applications



3.2.6 Hybrid Applications

Hybrid App



3.2.7 Hybrid vs Native Applications

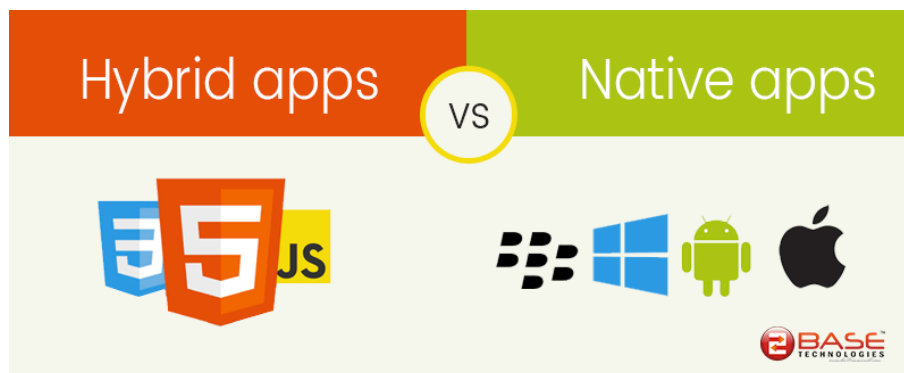
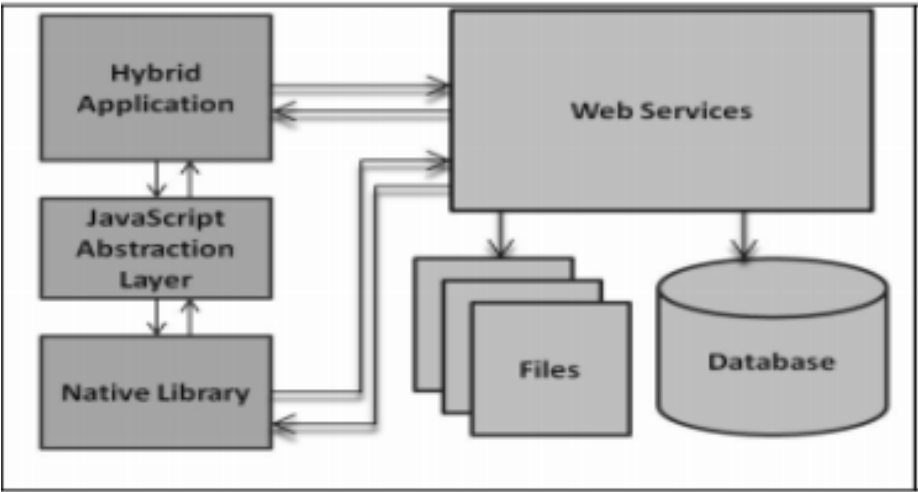


Figure 3.1: Hybrid Application



Chapter 4

System Design

Blah.....

Chapter 5

System Evaluation

Blah.....

Chapter 6

Conclusion

Blah.....

Bibliography