**Spot Finder Android app for Outdoor sports such as BMX and skateboarding**

**SEAN MOYLAN & SHANE MORAN**

BSc. (Honours) in Software Development

**Final Year Project**

Friday April 24, 2020

Advised by: Martin Kenirons

Department of Computer Science and Applied Physics

Galway-Mayo Institute of Technology



**Contents**

**Table of Contents**

1. **Introduction**
   1. **Context**
   2. **Objectives**
   3. **Overview**
2. **Background**
   1. **Fdfg**
   2. **fgffg**
3. **Methodology**
   1. Principle of Android Apps
   2. Spot Finder Android Apps
   3. Solutions of Spot Finder Android Apps
   4. Encryption by public-key cryptography
   5. Homomorphic encryption
   6. Vote Mixing Systems
   7. Examples of Spot Finder Apps
   8. Conclusions
4. **Technology Review** 
   1. Bitcoin: P2P decentralized currency
   2. Users - Addresses
   3. Transactions
   4. The chain of blocks (Blockchain)
   5. Proof of work
   6. Consensus
   7. Other characteristics of blockchain implementation
      1. UTXO (Unspent Transaction Output)
      2. Merkle Chain
      3. Rewards: mining and commissions
      4. Disadvantages, weaknesses and attacks on Bitcoin
      5. High latency
      6. Bitcoin software
      7. Double expenditure attacks
      8. Scalability
   8. Conclusions of the blockchain technology analysis
5. **Conclusions**
6. **Appendices** 
   1. Github 61

# About this project

**Abstract** Skaters, for the sake of innovation, transparency and citizen participation are interested in pilot voting projects based on blockchain technology. This innovative technology allows the decentralization of the government of operations, specifically the voting of citizens in an immutable, auditable, safe and reliable distributed registry. Throughout this study we will see the problems detected during its deployment phase, the solutions adopted and the conclusions for its use in institutions. The result will be achieved with the elaboration of a proof of concept prototype.

**Authors:** Shane Moran & Sean Moylan

## Introduction

The idea of the project is based on the problem that BMX riders, skaters and other outdoor sports do not always have available locations for recreational use at convenient times and infrequently summer.

## The purpose of this Final Year Project is to build an application using a new technology that the developers have never used before. Therefore, it allows for research to be done, learn up and coming up with technologies and discover new ways of developing software. The front-end technology being used to create the mobile application is Android Studio. Android Studio was chosen because of its new and gives the opportunity to learn a new Mobile Application framework. It will be connected to a back-end server running on a MongoDB Flask Database. This application will be useful and easy to use for users who will beneﬁt from this. This way, the developers will be able to show they have worked and have a fully functioning app that will satisfy what was required from them.

## The technologies used will be explained and why they were used throughout. This document will contain steps taken in creating the project and they will be documented clearly. The set research and development will all be detailed and allow others to understand why speciﬁc software was chosen. It will in detail compare similar technologies that could have been used. These sections will demonstrate the similarities between speciﬁc frameworks helping to explain what technologies were needed for the development application. This project is implemented using technologies that have not been used together too often which may lead to some issues. This means any problems faced or certain aspects found to be new will be documented along with the development.

## Context

The context of this project revolves around having fun with action sports anywhere, wishing to ﬁnd out more spots to BMX or skate. Opening the app on the user’s phone, they can view the various spots via Google Maps. This application prevents the user getting lost and gives them accurate information about each location. The information on the application will be provided by the users, to provide variety. Reading the data and retrieving images from the database needs to be very fast, as any delayed hang in performance could lead to a bad user experience.

The project will be developed as an android application which provides users with images, information and a map of the various spots. Users of the application can find their nearest spots easier in relation to their current location at any given time. When the user decides they wish to learn more information about a speciﬁc location, they can retrieve images and text about the spot on the application. Along with the mobile application a web application will be developed for use, to update, delete or add information to the spots database.

## Objectives

The project will require a number of objectives to be accomplished in order to provide a solution that works and is suitable for use by outdoor sports enthusiasts. [<https://github.com/seanmoylan/Applied_Project/issues>]

* A Mongo database will be used to store the text which appears in the application. This database will need to be setup and hosted in such a way that it can be accessed from clients through the android application.
* Pull information from the Mongo DB using ﬂask.
* A client mobile application will be the main product / asset for the project. This application will be able to locate the user via the GPS on their mobile device. The app will then show the user the locations of each spot. This will provide the user an idea of how far they are from the chosen spot and will allow them to see when they are near their destination. Each location is speciﬁed on the application as a list on the home page. Then chosen images and information will appear for each location.
* An additional requirement is to develop a web application for the use of the company to allow them to modify the information shown to the user. The web application is a simple website that can access the database to create, update and delete information from the database.
* Using the Google Maps application programming interface (API) within the mobile application to show the user their location along with the speciﬁc spots. Each spot is indicated by a marker in the map. When the user clicks on a speciﬁc marker, the name of the location appears as a label.

## Overview

An app developed for BMX, Skateboard and any other sports that takes advantage of the outdoors and streets to have fun or help them progress in their chosen sport. The app will allow the user to locate popular locations for use in these sports that other users have uploaded. This will allow people to travel all around the world and be able to quickly find the best spots to visit to BMX, skate ... etc [<https://github.com/seanmoylan/Applied_Project/blob/master/README.md>]

# Background

## Collaboration

The idea came through Sean Moylan for making a spot finder app. This application will enable the easy accessibility of information for tourists. All images and information included in this application was provided by users. We began the planning stage of the project with an initial meeting with our mentor (Martin Kenirons). The initial meeting took place in GMIT in October 2020.

During the meeting, we discussed exactly what was wanted as an outcome of this project by Martin as set out in the objectives. Meetings took place once every two weeks from this point to mid-February in order to iron out all the details in relation to design, requirements and any problems regarding the project. From January the development of the project, during these meetings the developers produced a working version of the application. This allowed Martin to give immediate feedback and change any of the requirements. Martin began to give the developers the information needed for the application in December and January in order to begin population of the database.

## Technology

During the initial project planning, such frameworks that were considered included Android Studio, Ionic, React native and Flutter. But we chose android studio. Android studio provides a flexible Gradle-based build system. It builds variants and multiple APK generation. It has expanded template support for Google Services and various device types. It provides a rich layout editor with support for theme editing. Line tools to catch performance, usability, version compatibility, and other problems. It has ProGuard and app-signing capabilities. Its built-in support for Google Cloud Platform, making it easy to integrate Google Cloud Messaging and App Engine. [<https://www.quora.com/What-are-the-pros-and-cons-of-Android-Studio>]

The back-end services researched include Docker, MySQL, MongoDB, Cassandra, Flask, NodeJS and Django. After much research and reviewing of examples Flask and MongoDB were chosen.

### Android vs Ionic

Android Studio and Ionic are similar in what they oﬀer with regards to prebuilt components in Android Studio and a comprehensive suite of built in widgets in Ionic [<https://developer.android.com/guide/topics/ui>]. With Ionic, you create a real native app but you do this by creating a web app (with Hyper Text Mark-up Language, JavaScript and Cascading Style Sheets) which will be wrapped by a real native app that hosts a web view while Android studio you tend to write Java (or Kotlin) code which can be compiled on the target device. The main reasoning for using Android studio over Ionic is that it is reliable and resourceful. It is powered by Android and therefore is well documented and although it’s still relatively old there are loads of interactive talks online about its upcoming, current and new feature. Ionic has numerous third-party built-in tools. It has thousands of threads on Stack overflow and packages on NPM (node.js package manager) to help developers along the way.

### Android vs React Native

### React Native is similar to Android development and compiles to native application by default. A basic React Native application is given a basic set of components. The developer must style most of them separately for each platform. This creates more work for the developer and increases time of development and cost of development for a company. React Native is developed by Facebook, while Android was various developers, but most notably Google. Each company is well developed and looked upon favourably by smaller businesses. React Native applications are developed using JavaScript and React libraries to build user interfaces. React previously existed to create web applications. Android is developed using various programming languages such as Java, C, C++, Kotlin and is solely used for mobile application development. Android is a very popular platform with over 980,000 results on GitHub in comparison to 145K for Flutter. This is because Android is older and developed by Google, therefore it is becoming more and more easier to develop in each day.

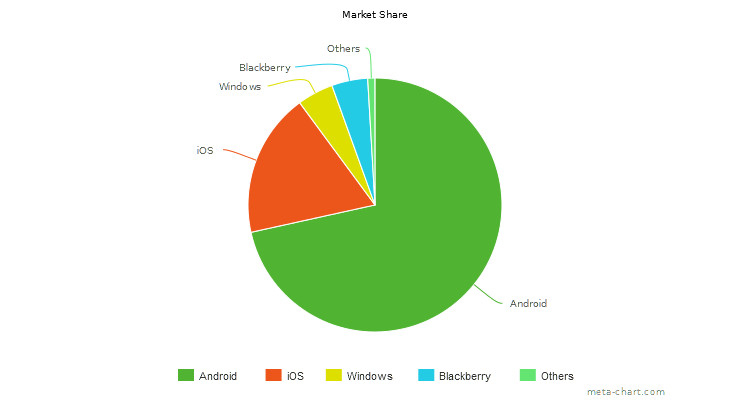
### Android vs Flutter

The original reason for using Android was the fact that Flutter is so new and the idea and challenge of learning a new cross platform mobile application platform didn’t sound appealing. Also, learning a programming language (Dart) with less resources to that of Android, would not be ideal. Android provides you with the API libraries and developer tools necessary to build, test, and debug apps for Android. It also provides a rich application framework that allows you to build innovative apps and games for mobile devices in a Java language environment. Flutter is Cross-platform mobile framework from Google as well and is a mobile app SDK to help developers and designers build modern mobile apps for iOS and Android. Android SDK belongs to "Frameworks (Full Stack)" category of the tech stack, while Flutter can be primarily classified under "Cross-Platform Mobile Development". "Android development" is the top reason why over 280 developers like Android SDK, while over 13 developers mention "Hot Reload" as the leading cause for choosing Flutter. Flutter is an open source tool with 69.5K GitHub stars and 8.11K GitHub forks. Here's a link to Flutter's open source repository on GitHub. [<https://stackshare.io/stackups/android-vs-flutter>]

### Why Android?

We opted to develop an Android app for many reasons, other than the fact that it has the most resources. After a good deal of researching other options of Mobile application development, Android came out on top by far. Firstly, the main reason is the use of java as a programming language makes it easy to port the app to multiple operating systems like Symbian and Ubuntu. Thus, businesses can target multiple platforms with Android app development. Android apps have a rapid development cycle lasting a few hours. It offers a competitive edge to companies who wish to have a quicker go-to-market for their new idea. Reduced Time to Market (TTM) is, thus, one of the best benefits of Android development.

Another one is its availability of the Android SDK. The development teams can use the material design from these SDKs to build interactive apps. However, developers are required to pay a one-time registration fee for application distribution. After that, they can leverage any computer device to build and test the product for their smartphones, ensuring low investment and increased user engagement. In turn, the end users, are benefited by an interactive app, and the enterprise gains a higher return on investment. Android P introduced several additional and in-built security features. It will help with the protection against malware and viruses. Thus, safety and reliability are exceptional benefits of android application development. To conclude, with over 75% Android device users today, developing an app on this platform is a value proposition for organizations globally. It helps them address a higher range of audience and gain control over the market. [<https://www.rishabhsoft.com/blog/5-advantages-of-android-app-development-for-your-business>]



### **Server**

### **Flask is a micro web framework written in Python. This means flask provides you with tools, libraries and technologies that allow you to build a web application. This web application can be some web pages, a blog, a wiki or go as big as a web-based calendar application or a commercial website. Micro-framework is normally framework with little to no dependencies to external libraries. This has pros and cons.**

### **Pros would be that the framework is light, there are little dependency to update and watch for security bugs,**

### **Cons is that some time you will have to do more work by yourself or increase yourself the list of dependencies by adding plugins. [**<https://pymbook.readthedocs.io/en/latest/flask.html>]

### **It is designed to make getting started quick and easy, with the ability to scale up to complex applications. It began as a simple wrapper around Werkzeug and Jinja and has become one of the most popular Python web application frameworks. Flask offers suggestions but doesn't enforce any dependencies or project layout. It is up to the developer to choose the tools and libraries they want to use. There are many extensions provided by the community that make adding new functionality easy. [**<https://palletsprojects.com/p/flask/>]

Python Flask vs Node.js

Choosing a language back end is an important step to the development planning phase of any project. While ﬂask is written is Python, node.js is JavaScript which allows JavaScript run on the server side and not in the browser because it is an environment. Flask is a micro framework for Python based on Werkzeug, Jinja2andgoodintentions. The comparison of Flask and node.js breaks down to a comparison between Python and JavaScript. The performance at the back end of any application is important for the speed and response time of the application. Both ﬂask and node.js have good performance however node.js has better performance this is because it is based on chrome V8, a fast and powerful engine. Flask is suﬃcient once application is not memory intensive. Scalability of the application when running on node.js is optimum as node.js supports an asynchronous application. This ensures for smooth, fast scaling. Python’s Flask does not support traditional asynchronous programming, but this can be mimicked with coroutines, making asynchronous processing achievable.

### Python Flask vs Python Django

Django is a full-stack web framework for Python, whereas ﬂask is a lightweight and extensible Python web framework. Django is focused on the ﬁnal product alone, it is an all-inclusive experience. Flask focuses on the experience and the learning opportunities. It provides simplicity, ﬂexibility and ﬁne-grained control. Although Django comes with everything built in, it is diﬃcult to change predeﬁned platforms, libraries etc. Although Django was released ﬁrst in 2005 in comparison to Flask in 2010 the latter has outgrown Django and is a more popular framework today. Although both are easy on the eye, some would say that Django has underpowered templating and Object-Relational Mapping.

### Python Flask vs PHP

Another alternative was the popular general-purpose scripting language that is especially suited to web development, known as PHP. Its Fast, flexible and pragmatic, PHP powers everything from your blog to the most popular websites in the world. But, its inconsistent API, fragmented community and lack of a routing system lead to us choosing Flask. [<https://stackshare.io/stackups/flask-vs-php>]

### **Database**

#### **MongoDB** is one of many non-relational databases. It is a NoSQL database that is used across many areas in the technology and business sector. MongoDB is a free open source database management system. As it has evolved over the years it is now one of the most popular document-oriented databases. It is run on a mongo server and can be created and controlled from your command prompt. Data in MongoDB is stored in JSON like documents. A format called BSON which is JSON in binary style issued for document storage in the database. This allows for an easy to read format of the data. Dissimilar to relational databases for example MySQL which uses rows and tables as a database structure MongoDB is formed of collections and documents. Each database in mongo can have multiple collections. ”A collection is a group of documents”. These documents can have many ﬁelds. The Documents in the database are a set of key value pairs. An id will be assigned to the documents making your database easy to edit. As MongoDB is a NoSQL non-relational database it diﬀers in many ways from a relational database management system. A relational database is row and column based as opposed to document and ﬁeld based. MongoDB is considerably easier to set up and is not vulnerable to SQL injection. A RDMS can be more challenging to understand and lets down the database in terms of its hierarchical storage not being as good and how it is open to SQL injection. One of the more popular relational databases is MySQL. Compared to MongoDB, it is quite inﬂexible in terms of the database schema. MongoDB is known for its ability to ”handle large amounts of unstructured data” which MySQL lacks in its technical features. Improving databases in terms of storage and their schemas can help to build more eﬃcient and easier to understand technologies. MongoDB runs better than many other database systems used today. It is a high-quality technology which has features that are desirable in having a secure database. Known for its scalability, memory processing and concurrency MongoDB makes for a desired database. All these features make it easy to develop a fast, reliable database which makes it work alongside evolving applications and is sought after by businesses. It is ﬂexible and can be adapted to work with many platforms and can be integrated into software much easier than other database management systems. The most common frameworks that are used in conjunction with MongoDB are NodeJS and a newer one Mongo Flask. It is also supported by cloud services such as Amazon Web services and Google cloud platform.

### MongoDB vs MySQL

There are ultimately two types of databases. These include SQL database and NoSQL databases. MySQL is an example of a SQL database while MongoDB is an example if a NoSQL database. MongoDB is stored in a document like json. It is ﬂexible and dynamic because the structure can be changed to meet customers’ needs and can be scaled horizontally. SQL is written in SQL query language and can be scaled vertically. MongoDB is a much newer server and was released in 2009 vs SQL server which has been published since 1989. MySQL had document orientated structure model while SQL is a relational database management system (RDBMS) model. Joins, concurrency and foreign keys are not supported by MongoDB - only SQL databases. MongoDB’s are created and maintained in an agile development practise while SQL databases are supported mainly by waterfall life cycles practises. This makes MongoDB’s more popular with modern companies who use agile development and many older companies are now switching to use MongoDB and agile also. Data Schemes are Dynamic in MongoDB but static/ﬁxed in SQL databases. MongoDB’s are more usable and ﬂexible as they can be run on Windows, Linux and os X operating systems in comparison to MySQL which can only be run on Windows operating systems.

### MongoDB vs Cassandra

Both MongoDB and Cassandra are NoSQL databases. Large companies such as Facebook use both databases. Cassandra can handle large amounts of unstructured data e.g. Instagram has about 80 million photos uploaded daily to its Cassandra database. MongoDB, because it is schema-free, documents can be created without creating their structures ﬁrst, in comparison to structures having to be predeﬁned in Cassandra databases. Cassandra databases are queries with CQL querying language. This is similar to SQL querying language. MongoDB currently had no support for any querying languages. MongoDB has queries which are structured as JSON fragments. Cassandra was released in 2008 by Facebook but currently being maintained by Apache software foundation. This is a newer NoSQL database than MongoDB and already has become more advanced and popular among large companies.

### MongoDB vs Microsoft SQL Server

Our final option was a relational database management system developed by Microsoft, called Microsoft SQL Server, and is a database management and analysis system for e-commerce, line-of-business, and data warehousing solutions. Whereas MongoDB is the database for giant ideas. MongoDB stores data in JSON-like documents that can vary in structure, offering a dynamic, flexible schema. MongoDB was also designed for high availability and scalability, with built-in replication and auto-sharding. Microsoft SQL Server and MongoDB can be primarily classified as "Databases" tools. "Reliable and easy to use", "High performance" and "Great with .net" are the key factors why developers consider Microsoft SQL Server; whereas "Document-oriented storage", "No SQL" and "Ease of use" are the primary reasons why MongoDB is favoured. MongoDB is an open source tool with 16.3K GitHub stars and 4.1K GitHub forks. Here's a link to MongoDB's open source repository on GitHub. Microsoft SQL Server also had expensive licensing which there were cheaper alternatives for. [<https://stackshare.io/stackups/microsoft-sql-server-vs-mongodb>]

# Chapter 3

## Methodology

### Project Management

Project Management is key element to ensure that the project is laid out correctly and each component part is complete at a given date etc. Because this project was completed by a team it was important both members completed certain aspects of the project.

In order to manage this project, GitHub was used as it was very useful due to it being a team project. It enables both members of the team to work on separate individual branches. After each iteration, these branches were merged into the master branch. After each integration, the project was at a working state of the project. This could be used for industry standard as continuous delivery. For this project it meant that after each Agile sprint there was a working version to present to customer and supervisor. Although it is a working version is may still need changes in the next sprint. At the beginning of each sprint all branches pull from the master so as every team member is working oﬀ the same latest release of code.

### Agile

The project used Agile project management methodologies. An iterative approach was taken. This meant that each week certain tasks are completed. The project team met once a week to discuss what has been completed since the last meeting, what must be completed before the next and discuss any difficulties experienced throughout the duration. The meetings are short and concise. Once every week to two weeks, the team met their mentor and explained what was completed since the last meeting and is to be done before the next. These weekly meetings were the iterative approach. Each week the process would be repeated to meet, plan, design, develop, test and evaluate.

### Project Planning

During the meeting and planning phase, the milestones for the project were identiﬁed and broken down into simpler, more manageable tasks. The tasks are then grouped into sprints or iterations lasting one week each (in industry normally 1/3 weeks). These are the tasks which must be be complete before each meeting. Each sprint started after the weekly meeting with the supervisor and the aim is for the work to be complete before the meeting the following week.[43].The plans for the project and the sprints are taken from the user stories. Development iterations convert the iteration plan into working code.