vfgfg

g00338607

May 2020

1 Introduction

There is a theory which states that if ever anyone discovers exactly what the Universe is for and why it is here, it will instantly disappear and be replaced by something even more bizarre and inexplicable. There is another theory which states that this has already happened.

Figure 1: Logo

2 Background

2.1 Collaboration

The idea came through Sean Moylan for making a spot finder app. This application will enable the easy accessibility of information for outdoor sports people. 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 2019. 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, but due to the unforeseen circumstances surrounding the spread of the Coronavirus; we were unable interact in person and this led to meetings having to take place online via Microsoft Teams. that 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.

2.2 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. [3] [https://stackshare.io/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.

2.2.1 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 [4] [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 is 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 develop-

ers along the way. Ionic provides all the functionality which can be found in SDKs of native mobile environment. Ionic application enables inventor to build their applications and customize them according for different platforms such as for iOS, for Android, and deploy through Cordova. It includes various mobile components, extensible base themes and typography. Ionic uses framework of JS i.e., Angular, which offers custom components and method to interact. Therefore, Ionic is a backdrop support which operate on top of Cordova to build hybrid apps. It allows to develop an application interface which are comparable as they were web pages i.e., using HTML, CSS, JavaScript. And these apps run inside WebView of native application. Virtue of ionic framework is – it is an open source framework in which code needs to be written once and execute on numerous mobile devices. The use of only one programming language enables the app on all the different mobile OS. Availability of various plugins to use hardware and other extra services for mobile app. Ionic apps has the well build graphics. This is the developer's favourable framework where simple technologies are used to build the entire interface of the application. [5] [https://www.academia.edu/download/58274892/IRJET-V5I5609.pdf] [Chaudhary, Priyanka. "IONIC FRAMEWORK." (2018).]

2.2.2 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 React Native. This is because Android is older and developed by Google, therefore it is becoming more and more easier to develop in each day. Both are primarily classified as "Frameworks (Full Stack)" and "Cross-Platform Mobile Development" tools respectively. [6] [https://stackshare.io/stackups/android-vs-react-native] The key thing about React Native is that it's still in development and we are yet to see its full potential. In the future, it may be more powerful and efficient and allow for even more use cases, but for now it cannot fully replace native mobile development. However, its write once, use everywhere paradigm can be a tremendous time and money saver if used on the right projects. [7] [https://medium.com/mop-developers/mobile-app-development-react-native-vs-native-ios-android-49c5c168045b]

2.2.3 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 did not 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. [8] [https://stackshare.io/stackups/android-vs-flutter]

2.2.4 Comparison

Native applications for Android are built in Java or Kotlin, and native applications for iOS are generally build in Swift. For example, when developing in Flutter there is a single code base; this means the application is written once and it works for both iOS and Android. Third Party libraries are widely available for native language applications. This is due to the popularity of their language and therefore there are very few problems that cannot be solved by referencing websites such as Stack overflow, etc. , but, Native languages and Flutter both give a native application appearance. [9] [https://academind.com/learn/flutter/react-native-vs-flutter-vs-ionic-vs-nativescript-vs-pwa/]

2.2.5 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 per cent of 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]

2.3 Server

2.3.1 Flask

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.

2.3.2 Pros and Cons

Its framework is light, there are little dependency to update and watch for security bugs. For its 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 does not 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/]

2.3.3 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.

2.3.4 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.

2.3.5 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]

2.4 Database

2.4.1 MongoDB

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.

2.4.2 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.

2.4.3 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 hav-

ing 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.

2.4.4 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]

3 Methodology

"I always thought something was fundamentally wrong with the universe" [1]
Methodology

4 Methodology

"I always thought something was fundamentally wrong with the universe" [1]

References

[1] D. Adams. *The Hitchhiker's Guide to the Galaxy*. San Val, 1995.