**

By: Kevin Dawson

Supervisor: Joseph Kehoe

Research Document

Abstract

The purpose of this document is to provide a written account of the research carried out by the author for the development of the application. This includes looking at similar services, technologies which include languages, IDE’s, databases, API’s and design all grouped into their respective sections and concludes with a section outlining each of the selected technologies.

Contents

[Introduction 3](#_Toc468027997)

[Similar Technologies 3](#_Toc468027998)

[Hitchhiker Carpooling App 3](#_Toc468027999)

[Carsharing.ie 4](#_Toc468028000)

[Carpoolworld.com 5](#_Toc468028001)

[Conclusion 6](#_Toc468028002)

[Technologies 6](#_Toc468028003)

[Platforms 6](#_Toc468028004)

[Android 6](#_Toc468028005)

[IOS 7](#_Toc468028006)

[Conclusion 8](#_Toc468028007)

[Programming Languages 8](#_Toc468028008)

[Python with Kivy 8](#_Toc468028009)

[Swift 9](#_Toc468028010)

[Apache Cordova 9](#_Toc468028011)

[Adobe PhoneGap 10](#_Toc468028012)

[Java 10](#_Toc468028013)

[Conclusion 11](#_Toc468028014)

[Development Environments 11](#_Toc468028015)

[JetBrains WebStorm 11](#_Toc468028016)

[PhoneGap Developer App 12](#_Toc468028017)

[Database 13](#_Toc468028018)

[SQL 14](#_Toc468028019)

[NoSQL 15](#_Toc468028020)

[Conclusion 16](#_Toc468028021)

[Application Programming Interfaces (APIs) 17](#_Toc468028022)

[Map APIs 17](#_Toc468028023)

[Conclusion 17](#_Toc468028024)

[Bibliography 19](#_Toc468028025)

# Introduction

Looprac is a car sharing app. The purpose of this app is to allow people to travel together to a mutual destination. This has benefits of decreasing the number of cars on the road, its environmentally friendly and sociable which would help reduce risk of people falling asleep behind the wheel. The app will allow registered users to advertise space in their car or to request a spot in a car in their surrounding area. The drivers and passengers will be able to rate each other and store favourites for future use. The app will also support a ranking and achievement system to try and encourage the use of the app and car sharing such as badges for milestone achievements and periodically informing them with be progress bars of how much distance they have travelled sharing lifts and how many lifts they have shared.

This document will look at technologies that are going to be used to create this app, this will include the target platform for the app, similar technologies, implementation languages – such as possible languages, the development environment, databases, user experience and API’s that will be used. The first section that will be looked at is similar technologies or services as the app and what they are like, this will give insight into what should be included in the app or if anything should be added to enhance the experience.

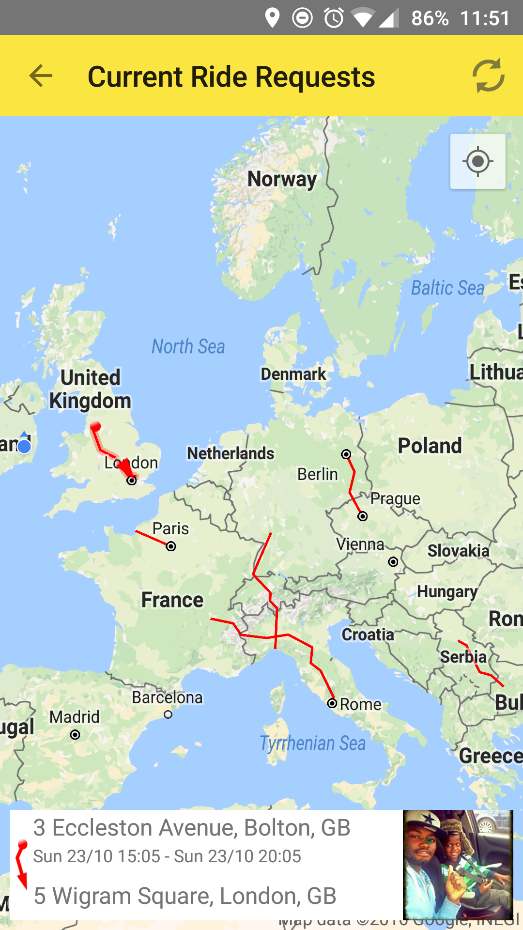
# Similar Technologies

With the move to green and a push towards being more environmentally friendly there are similar apps with the car sharing idea behind them in an effort to reduce the carbon footprint in the Google Play store and on iTunes. Upon research the author found few apps that were able to be used in Ireland, one such app that could be used is below, more popular were websites that carried out the same service to users.

### Hitchhiker Carpooling App



Hitchhiker Carpooling is a car sharing app developed by Mendrejk, which is available for Android. With use of the app the author found the app to be simple to use. On initial use it asks the user for a login the choices were through Facebook or Google+, this is then the profile that the user is logged in as with future use. The UI of the app is very simple with minimal colours applied. The map system for requesting a new ride is a little difficult to get working as it needs the exact address to where the user wants to go, no general area can be used. It has an option to view ‘current ride offers’ to look at any routes that drivers have put up offering a space in their car. Upon selecting this option, it shows the map with current route offers being shown in red



It allows the user to select a monetary amount to offer drivers or request off passengers for every hundred miles travelled, this is left for the users to complete as there is no function within the app for sending or receiving money. Overall the app is available around the world where people want to offer or request a lift.

### Carsharing.ie

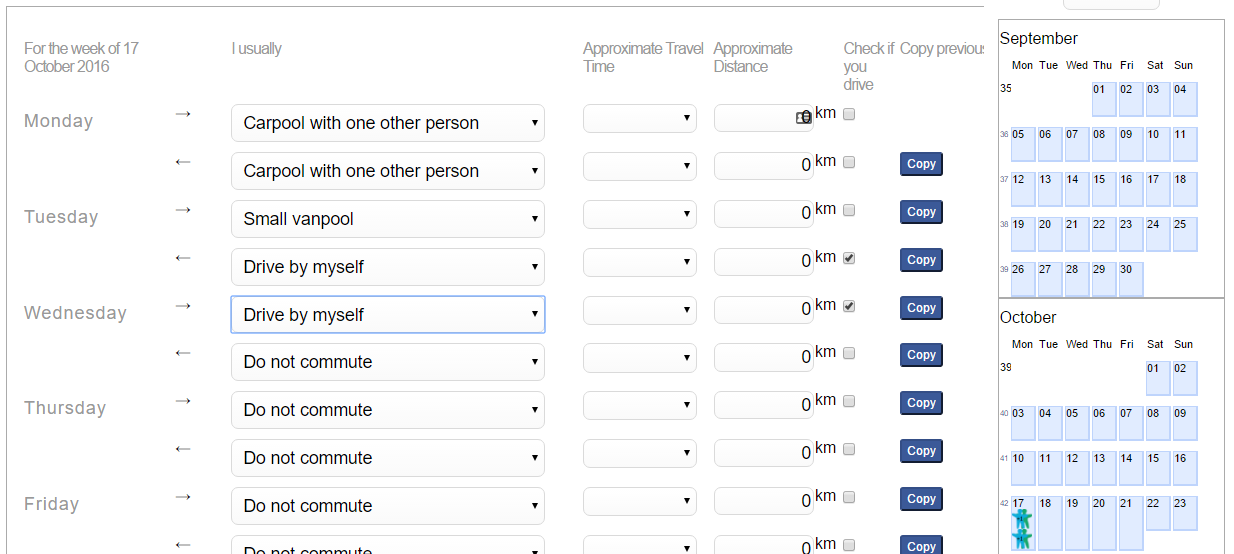


This is a site run by the National Transport Authority, it shares the same idea and principles of the app that users can join for free, find people looking to go to the same destination either looking for a lift or driving and arrange to meet with them. The site has incentives for its users to promote and use the car sharing idea such as being entered into a draw if they log over twenty car sharing trips. This site has partnerships with Universities and organisations to promote car sharing within them such as Deloitte, NUI Maynooth, ESB, Accenture, ESB and DIT to name a few. They have groups associated with each of the partners for their users to join and find people in the same organisation or college to make it easier to find people to arrange with.

### Carpoolworld.com



This is another site that offers the car sharing idea. It is easy to organise, whether you are a driver offering a spot or requesting a lift. The user can choose their collection point and destination either by typing it into the requirements input boxes or by selecting both points on a Google map and it will draw out the route for them. It also offers a commuting schedule that the user can update each week and submit to make it easier to find a car to share.



### Conclusion

These are just a few similar technologies that were signed up to and researched for this project. The author found that there were more website services available to the Irish market than there were mobile apps available. The experiences on each of them were similar in nature – to sign up, choose a destination, find people to share with. Some tried to encourage monetary contributions for the service while others didn’t bring the issue up. As well as that only one that was looked at had any form of incentive to try and encourage the users to participate. The next section will look at the possible technologies that can be used to develop the app. This includes the platform, language, the development environment and types of databases.

# Technologies

In this section research will be carried out on different technologies to be considered for this project. This includes technologies for Android or IOS development, languages and backend technologies.

## Platforms

### Android



Android is owned by Google and is one of the most popular mobile operating systems in the world powering devices in more than 190 countries. Its built using open-source Linux and has more than 300 hardware, software and carrier partners (Developer.android.com, 2016). With this open-source in mind it allows the creation of powerful and customizable apps with little to no restrictions. Android has expanded beyond just mobile phones and now includes wearables such as tablets, smart watches, televisions and cars.

Before it was owned by Google, Android Inc. was a standalone software company with no products produced for two years. The idea behind the company was to make the next generation of smartphone software. This idea began to gain traction in 2005 when their plan on focusing on the best web-connected experience and creating an environment any developer could build on got the attention of investors. Google was in need of a smartphone company to compete with Microsoft and Blackberry and Android wanted more phones with Google as the default search engine. With these similar interests Google acquired Android in 2005 (AndroidCentral.com, 2015).

Then the release of the first Android smartphone in 2008. The HTC made, Google backed G1 or HTC Dream was released. By 2009 they first major updates to Android after the initial release were versions 1.5 and 1.6, Cupcake and Donut respectively. Cupcake paved the way for touchscreen only Android phones with the built in on screen keyboards that are the standard of today. By the end of the year Android was also making advances in the areas of speech recognition and text-to-speech (Android Central, 2015).

Fast forward through the years and versions – Donut, Éclair, Froyo, Gingerbread, Honeycomb, Ice Cream Sandwich, Jelly Bean, Kitkat, Lollipop, Marshmallow and currently Nougat and Android is dominating its market.

The idea that the original creators of Android had to create an environment that any developer could build on has become one of the most popular development paths, with both native and hybrid development. Developers are able to create apps for this platform using many technologies such as Java, Html, CSS, JavaScript and more.

### IOS

IPhone with its release in 2007 had a huge historical impact on the mobile market, it paved the way for smartphone user experience which paved the way for responsive touch screens that are being used today. With the release of the iPhone came the release of iPhone OS. This OS was lacking to be blunt, as it couldn’t support 3G, copy and paste function, email attachments or multimedia messaging service (MMS). It wasn’t until the following year with the release of the iPhone 3G that the OS got an update along with the introduction to iTunes and third party apps and games. Slowly the OS got better with the release of the iPhone 3GS, the copy and paste function and MMS were added along with other functions. With the 3.2 version it contained improvements for the iPad. In 2010 the IOS 4 was released and was the first one to have the name ‘IOS’, which released with the iPhone 4. Each year new versions were released alongside the new iPhone. IOS 5’s main new feature was Siri, the virtual assistant (Telegraph, 2015). The current version available is IOS 10 which was released in September 2016.

IOS has been known for its strict development, giving more control to Apple than to the developers, some of this has eased with the release of IOS 10 with Siri now open to developers and that native apps can now be removed. The choice of IOS development also includes native, with languages such as Objective-C, Swift and hybrid languages such as Html, CSS and JavaScript.

### Conclusion

The decision on whether to develop this app on Android or IOS was decided on the statistics of market share of both platforms from 2016Q2. With Android dominating the market with a share of 87.6% and IOS behind with 11.7% (idc.com, 2016). From these statistics the decision to make the app on the Android platform first was decided.

With the app being developed on the Android platform, next was to decide the minimal Android OS version that the app will be developed on. As of September 5th 2016, the most popular versions are version 4.4 (KitKat) with a 27.7% distribution share, followed by 5.0 and 5.1 (Lollipop) with 13.1% and 21.9% respectively and then 6.0 (Marshmallow) with 18.7% (Developer.android.com, 2016). Therefore, this app will be developed for version 4.4 onwards.

The next section will look at the possible languages to use for developing the app.

## Programming Languages

### Python with Kivy





Python is an interpreted, object-oriented, high-level programming language and is dynamically typed. It is popular as a scripting language due to its combination of data-structures, dynamic typing and dynamic binding. Python is different to its other languages in the sense that it has simple, easy to learn syntax which emphasizes readability (Python, 2016).

Kivy is an open source Python framework for natural user interface development with support for all major desktop and mobile operating systems. Kivy is a tool that brings Python to mobile platforms. Once an interface to the Python code has been created using Kivy, there are methods for packaging the code for mobile platforms. Python for Android is a project that allows developers to create an APK (GitHub, 2016). Another packaging option is Buildozer, which automates the entire build process, downloads the prerequisites like python-for-android, Android SDK, NDK and others (Buildozer, 2016).

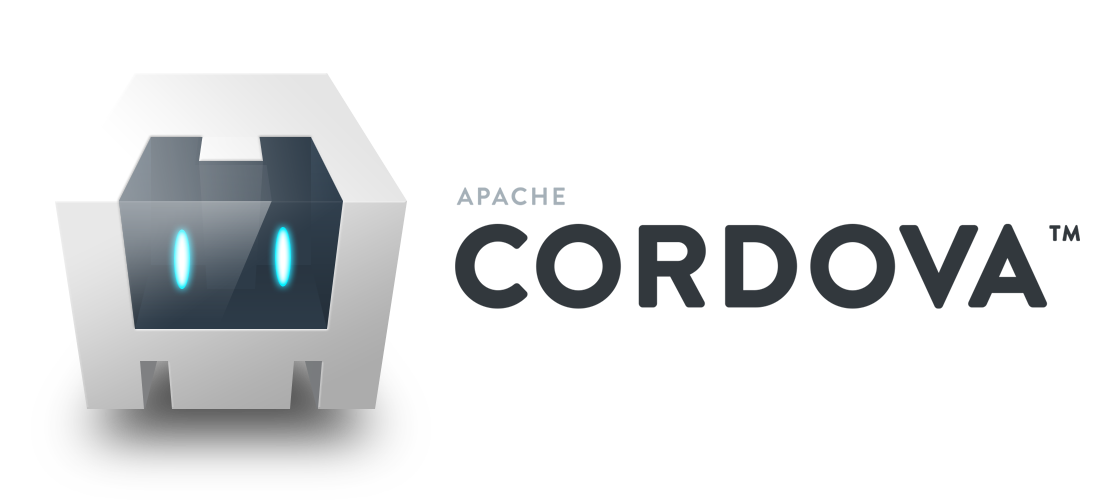
### Swift



Swift is the programming language intended for the use on all IOS devices, its currently being pushed as the successor to Objective-C by Apple. It adopts the best of C and Objective-C with less constraints. It uses the same runtime as the existing Objective-C system which enables Swift programs to run on many existing IOS 6 and OS X 10.8 platforms (Tutorialspoint, 2016).

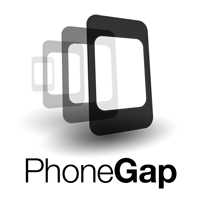
This would be used for native IOS app development if the car sharing app were to be ported to the IOS platform.

### Apache Cordova



Apache Cordova is an open source mobile development framework. It creates hybrid applications, which allows the use of HTML5, CSS3 and JavaScript for cross-platform development. Applications developed in this are executed and wrapped in a native application, targeted to each platform and use API bindings to access each devices capabilities such as sensors, data and other components. This offers a bridge for connection between web app and mobile device. Cordova however, since it creates hybrid apps, they depend on the native browser, which means they don’t perform as fast as native apps. (Sitepoint.com, 2014)

### Adobe PhoneGap



Adobe PhoneGap is a distribution of Apache Cordova, in fact they are the same but not the same. Apache Cordova was originally called PhoneGap created and managed by a company call Nitobi, which was acquired by Adobe. Before the acquisition was complete, Nitobi donated PhoneGap to the Apache Software Foundation so it could continue to be invested in by the community. Once Nitobi was settled within Adobe the team created a distribution of Apache Cordova and called it Adobe PhoneGap (Wargo, 2016). So Cordova is the engine that powers PhoneGap. PhoneGap creates hybrid apps for mobile using technologies such as HTML, CSS and JavaScript. This enables the developer to develop the application once and build and deploy it to multiple platforms.

### Java

Java is a high- level, object oriented language by Oracle. It works by taking the code and compiling it into bytecode which can be interpreted by the Java Virtual Machine(JVM). A major advantage of Java is that any device that has the Java Runtime Environment(JRE) installed on it can run Java, meaning it is platform independent. Java code can be structured based on real-world “things” and most importantly, the code can be reused. With Android and Java, the code is changed into another form which is understood by Android, this form is called Dalvik Executable(DEX) code. The part of the system that executes the DEX code is called the Dalvik Virtual Machine (DVM), which itself is a piece of software that runs on a specially adapted version of Linux. The purpose of the DVM is to hide the complexity of the hardware and software that Android runs on (Horton, 2015).

### Conclusion

The languages were selected for their types, Cordova and PhoneGap being hybrid application development and Java being native application development. They come with their strengths and weaknesses, such as with Cordova and PhoneGap, the development is faster, simpler and the application is easier to maintain. It also has a large repository of plugins available for phone hardware that can be used. Furthermore, you can change platforms when needed with a small bit of code. However, compared to native apps the performance is slower, due to it relying on the native browser.

With Java, native app development the main advantage is performance. As mentioned above they are compiled into bytecode, which gives the best performance for mobile. This includes fast and fluid animations as well as full access to the phones hardware and latest API’s. However, native development is far harder compared to hybrid development and furthermore the app would have to be remade in Objective-C for IOS development (Sitepoint.com, 2014).

The conclusion is that the selected language is Adobe PhoneGap. The reasons for this are that Cordova and PhoneGap are very similar and in some areas will work alongside each other. Although the hybrid approach has performance differences than with the native approach the strength of being able to develop the app once and deploy on multiple platforms is very appealing, saving both time and effort. Furthermore, the author has experience with web development.

The next section will look at the chosen integrated development environment and why it was chosen.

## Development Environments

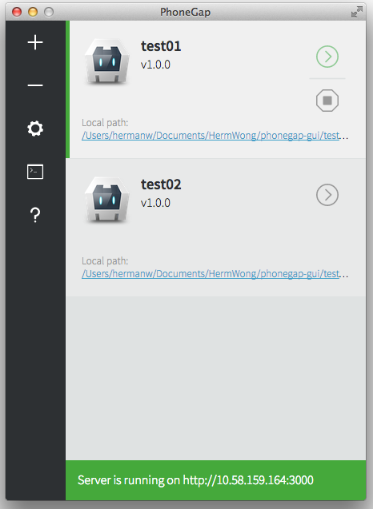
### JetBrains WebStorm

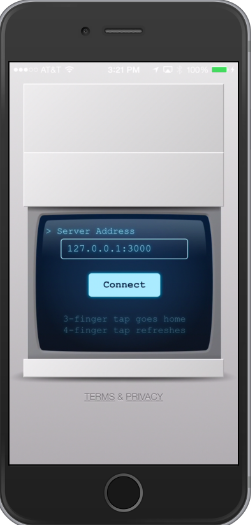


WebStorm is an IDE that provides first-class support for JavaScript, Node.js, HTML and CSS, as well as modern successors. It also supports many frameworks such as AngularJS, React and more. As Phonegap uses the technologies HTML, CSS and JavaScript this IDE provides a suitable work environment to develop Looprac. It has easy integration with source control such as GitHub, which is the selected source control for this project.

### PhoneGap Developer App

Developer App is an app developed by Adobe PhoneGap. When developing the app within WebStorm, using PhoneGap’s desktop app to build the application, it sets up a server to run the app on. Using PhoneGap Developer App on an Android device and putting in the IP address of the running server it connects to the app in development and displays it on the device. Any changes that are made using the IDE, both the desktop run web browser version and the PhoneGap Developer App version are updated simultaneously. This provides the developer with real time code updates showing what the current version of the build is like.



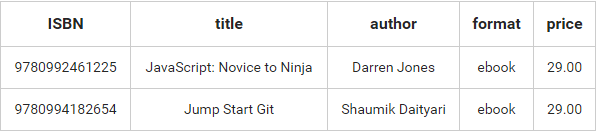


## Database

The app will need the ability to store user’s data for the purpose of registration, user profile information, driver and passenger ratings and other information. For this the app will need a database and one that will be lightweight to be used with the app, able to hold the required amount of data and to be cross platform for the option of future portability across platforms. Furthermore, it must be secure as it will be holding some sensitive data belonging to users.

Research was carried out on both SQL and NoSQL databases to find the database that hit the requirements above.

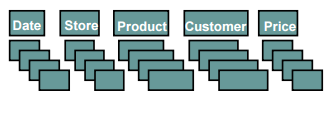
SQL stands for Structured Query Language and its databases have been primary data storage mechanisms for decades, their usage increased in the 1990s with the use of MySQL, SQLite and others. NoSQL stands for ‘Not Only SQL’ and has been in existence since the 1960’s but has only recently gained traction due to popular databases such as MongoDB, CouchDB and others (SitePoint, 2015). They both do the same thing but have different approaches on how they do it. SQL databases use a relational approach which uses predefined tables to store information, the records are represented as columns (fields) and rows (records). With SQL the tables relations can be one-to-one, one-to-many or many-to-many.



NoSQL do not implement the use of explicit structured mechanisms to link data like relational databases. Instead they can store the data by implementing three different types: key-value store, column store and document store.

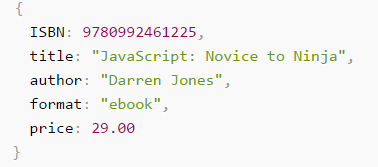
With key-value store the databases are designed for storing the data in key-value pairs and they do not have a schema.

Column store databases store the data in cells grouped in columns of data, which are then grouped into column families which can only contain a certain number of columns.



(Harizopoulos, et al., 2009)

Document store databases developed on the idea of key-value stores where documents contain more complex data. Each document is assigned a unique key which is used to retrieve the document. They use a JSON style document structure. This gives the schema flexibility over the relational tables which is why they are also referred to as non-relational databases.



Research was first carried out on SQL relational databases.

### SQL

#### SQLite

SQLite is a relational database that is lightweight, can process large amounts of data and is cross platform with Android, IOS, Windows Phone and Blackberry. Furthermore, Android comes with a built in SQLite database implementation.

However, it doesn’t fit the requirements needed. Although it can process large amounts of data, it stores the entire database in a single disk file and many filesystems limit the maximum size of files to something less than could be required. In regards to concurrency, SQLite supports an unlimited number of simultaneous readers, but will only allow one writer at a time. This can result in writer queues, this wouldn’t last long, but for this application this is not desired (Sqlite.org, 2016).

#### MySQL

MySQL is a relational database, meaning that all the data is stored into different tables and relations are established using keys. It’s a fast, easy-to-use relational database management system(RDBMS). Its open-source, works with many languages including PHP, PERL, C, C++, JAVA and more. Its quick and can handle large sets of data. It has excellent documentation and support online meaning that any problems that are encountered can be quickly dealt with. Its popularity has come from its performance, reliability and ease of use.

#### MariaDB



MariaDB is similar to MySQL created by several of the former core developers of MySQL who left soon after the Sun Microsystems acquired it in 2008, because they were dissatisfied with the quality of MySQL releases after the merger and slow rate of improvement. The similarities between MySQL and MariaDB are that apart from a lot of the files and binaries being the same the SQL commands are the same between the two, with a few MariaDB specific parts. They both use the same ports and sockets, client API’s and protocols.

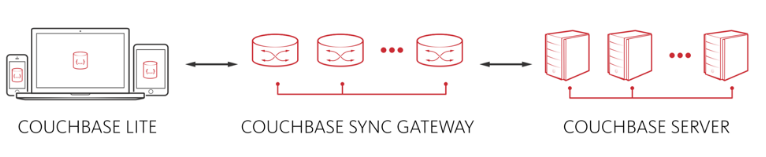
The differences between the two are that MariaDB has new features, performance improvements, better testing and bug fixes which are not found in MySQL – which were developed both in house and by external contributors such as Facebook, Twitter, Google and others (Bartholomew, 2012).

With performance, one area they focused on was the optimizer, which is the engine that sits at the core of MySQL and MariaDB and takes the entered SQL commands and turns them into instructions for the database, in MariaDB it performs faster than the optimizer in MySQL on complex workloads (MariaDB.org, 2012).

### NoSQL

#### CouchBase Mobile



CouchBase Mobile is a NoSQL database that is designed to build fast, powerful and secure mobile applications. It has a fully-integrated database, real- time synchronization, enterprise level security and highly scalable server. CouchBase Mobile offers three components – CouchBase Lite, CouchBase Sync Gateway and CouchBase Server.

With Sync Gateway comes the element of synchronization where it will sync between local database – CouchBase Lite and databases in the cloud CouchBase Server. CouchBase Lite manages and stores data locally on the device in document oriented JSON formats, this helps the app with network connection problems so that it will still be able to function if the app goes offline. When the app goes back online Sync Gate automatically syncs the data with the cloud servers. With data being sensitive, i.e. emails, passwords, addresses, etc. security is important, CouchBase offers security on an enterprise level which 256-bit AES full database encryption (CouchBase, 2016).

#### CouchDB



CouchDB is an open source database developed by Apache. This database allows the user to store their data safely on their own servers or with a cloud provider. It uses JSON document storage. The Couch Replication Protocol which is a protocol for synchronizing JSON documents between two peers over HTTP by using a REST API (couchDB, 2016) allows the data to transport between servers to mobile phones and web browsers (CouchDB, 2016).

### Conclusion

After looking at all these types of databases the chosen database for Looprac will be MySQL. The reasons for this choice is that Looprac doesn’t need much in terms of a database. It is only storing and retrieving data. With MySQL its free to use, easy to use and the performance it offers is more than is needed for Looprac.

The next section will look at application programming interfaces (APIs) that will be used in the development of the app.

# Application Programming Interfaces (APIs)

Application developers often do not have direct access to system calls or other components, but can access them through an application programming interface. The functions that are included in the API invoke calls or services. By using APIs, it makes it possible for applications to share data and take actions on one another’s behalf without requiring developers to share all of their software’s code. They expose some of a programs internal functions to the outside world in a limited fashion (Proffitt, 2013). In this document it will look at web APIs for the chosen database and also looking at map API’s so it can be incorporated into the app.

## Map APIs

Interactive maps within an app add a huge amount of value to the app including functionality with regards to location information and visually helps to improve the app. They can offer real time updates on the map view to help track objects, they can offer optimized routes

Google Maps is a free web mapping service by Google that provides various types of geographical information. It has an API to customize the maps and display them either on webpages or mobile apps. Google offers a free version of this API to be used by its customers, it offers separate API keys for Android, IOS and web apps.

Mapbox is another interactive map service that it offers to Android, IOS and web apps like Google. They also provide a free service. Mapbox offers huge customisation on its maps.

With having a map service in the app the next step is to be able to access the devices location so that it can be found on and use the map. This is done by a Device API plugin that is available with PhoneGap which gives access geolocation. With the map API and the phones geolocation it provides powerful functionality to the app.

The decided map API for the app is Googles, this is because of documentation and support for it due to its popularity amongst the mobile app world.

# Conclusion

This concludes the research document, research was carried out on similar technologies and concluded that most of the similar services were available to Irish market were websites, there were few app versions of this service available. This shows that there is a need for an app version however viewing the sites and apps showed that they were all similar in nature with only one offering some form of incentive to the users. With this in mind the car sharing app will have a form of gamification to it which will provide users with goals to reach and give them achievements when they are met in the form of badges throughout their use of the app helping to keep their interest, it will also keep them updated with how they are helping the environment as they clock up miles sharing a vehicle instead of multiple.

With the chosen platform the decision came to Android, this was based on research carried out on the market share that Android had over its competitors. This is followed by IOS. With the version of Android that will be the minimum requirement version for the app will be version 4.4 and this was decided based on the distribution statistics where it had the highest. This means that Android 4.4 to 7.0 will be targeted for the app.

The development of the app will be the hybrid process, which is developed with the use of HTML, CSS and JavaScript. The choice for hybrid app development was for the ability to develop the app and deploy it cross-platform without having to develop it again natively. Furthermore, the hybrid development is easier and quicker than that of the native approach. The author found that these strengths outweighed the weakness that hybrid apps performance is less than that of their native counterparts, with the hybrid approach the selected target platform will be the aim of the development but the finished app will also be deployed to IOS. To create the hybrid app Adobe PhoneGap was chosen, this was chosen over Apache Cordova so that it could be implemented with the target database.

The integrated development environment that was chosen was Intellij Android Studio. This was chosen as firstly it is able to import PhoneGap projects so that it can be developed within the editor. It was mainly chosen for the fact that the target platform is Android 4.4 and greater and this IDE provides the user with a virtual device to build the app with or to connect and build it to a physical device such as their Android mobile. Furthermore, this gives the advantage of emulating the app on multiple Android versions and devices if need be.

The database that was chosen was MySQL. The decision for this came after research of looking at both SQL and NoSQL databases. It was concluded that MySQL is the best choice for the project. This is due to the fact that Looprac just needs the basics of writing to and reading from the database. The performance and reliability that MySQL provides and the endless documentation on the internet, makes MySQL a good choice for the project.

The map API that was chosen was Google Map API, this was the choice because of the popularity of the API and the documentation and help provided for it. It allows the customisation of maps so that it fits in with the design of the app.

# Bibliography

Android Central, 2015. *Android's Early Days.* [Online]   
Available at: http://www.androidcentral.com/androids-early-days  
[Accessed 22 October 2016].

AndroidCentral.com, 2015. *Android Pre-History.* [Online]   
Available at: http://www.androidcentral.com/android-pre-history  
[Accessed 22 October 2016].

Bartholomew, D., 2012. *MariaDB vs. MySQL.* [Online]   
Available at: http://www.eandbsoftware.org/wp-content/uploads/2015/03/MariaDB\_vs\_MySQL\_-\_MariaDB\_White\_Paper\_-\_08\_26\_13\_001.pdf  
[Accessed 22 October 2013].

Buildozer, 2016. *Welcome to Buildozer’s documentation!.* [Online]   
Available at: https://buildozer.readthedocs.io/en/latest/  
[Accessed 27 October 2016].

CouchBase, 2016. *Mobile Database | Couchbase.* [Online]   
Available at: http://www.couchbase.com/nosql-databases/couchbase-mobile  
[Accessed 23 October 2016].

couchDB, 2016. *4.2. CouchDB Replication Protocol — Apache CouchDB 2.0 Documentation.* [Online]   
Available at: http://docs.couchdb.org/en/master/replication/protocol.html  
[Accessed 23 October 2016].

CouchDB, 2016. *Apache CouchDB.* [Online]   
Available at: http://couchdb.apache.org/  
[Accessed 23 October 2016].

Developer.android.com, 2016. *Dashboards | Android Developers.* [Online]   
Available at: https://developer.android.com/about/dashboards/index.html#Platform  
[Accessed 22 October 2016].

Developer.android.com, 2016. *the worlds most popular mobile platform.* [Online]   
Available at: https://developer.android.com/about/android.html  
[Accessed 22 October 2016].

GitHub, 2016. *kivy/python-for-android.* [Online]   
Available at: https://github.com/kivy/python-for-android  
[Accessed 27 October 2016].

Harizopoulos, S., Abadi, D. & Boncz, P., 2009. *Column-Oriented Database Systems.* [Online]   
Available at: http://www.cs.yale.edu/homes/dna/talks/Column\_Store\_Tutorial\_VLDB09.pdf  
[Accessed 23 October 2016].

Horton, J., 2015. *Android Programming for Beginners.* 1 ed. Birmingham: Packt Publishing.

idc.com, 2016. *smartphone os market share.* [Online]   
Available at: http://www.idc.com/prodserv/smartphone-os-market-share.jsp  
[Accessed 22 October 2016].

MariaDB.org, 2012. *MariaDB-5.3 optimizer benchmark.* [Online]   
Available at: https://mariadb.org/mariadb-5-3-optimizer-benchmark/  
[Accessed 22 October 2016].

Proffitt, B., 2013. *What APIs Are And Why They're Important - ReadWrite.* [Online]   
Available at: http://readwrite.com/2013/09/19/api-defined/  
[Accessed 23 October 2016].

Python, 2016. *What is Python? Executive Summary.* [Online]   
Available at: https://www.python.org/doc/essays/blurb/  
[Accessed 27 October 2016].

Sitepoint.com, 2014. *Native vs Hybrid App Development.* [Online]   
Available at: https://www.sitepoint.com/native-vs-hybrid-app-development/  
[Accessed 22 October 2016].

Sitepoint.com, 2014. *Native vs. Hybrid App Development.* [Online]   
Available at: https://www.sitepoint.com/native-vs-hybrid-app-development/  
[Accessed 22 October 2016].

SitePoint, 2015. *SQL vs NoSQL: The Differences.* [Online]   
Available at: https://www.sitepoint.com/sql-vs-nosql-differences/  
[Accessed 23 October 2016].

SitePoint, 2015. *SQL vs NoSQL: The Differences.* [Online]   
Available at: https://www.sitepoint.com/sql-vs-nosql-differences/  
[Accessed 23 October 2016].

Sqlite.org, 2016. *Appropriate Uses For SQLite.* [Online]   
Available at: https://sqlite.org/whentouse.html  
[Accessed 22 October 2016].

Telegraph, 2015. *Apple iOS: a brief history.* [Online]   
Available at: http://www.telegraph.co.uk/technology/apple/11068420/Apple-iOS-a-brief-history.html  
[Accessed 22 October 2016].

Tutorialspoint, 2016. *Swift Overview.* [Online]   
Available at: https://www.tutorialspoint.com/swift/swift\_overview.htm  
[Accessed 27 October 2016].

Wargo, J., 2016. *Which to Use: Cordova or PhoneGap?.* [Online]   
Available at: http://www.informit.com/articles/article.aspx?p=2478076  
[Accessed 22 October 2016].