TRACK RIDER

**1. INTRODUCTION**

**1.1 Background**

This Android Application is developed to track your Friends or Family or People travelling in groups towards a common destination. One of the traditional methods to get the current location of a person is to call that person & hope he/she gives you the correct information of the surroundings of their particular location. Another popular method is to ‘share live location’ on Whatsapp. While this is a good way to share 1’s location, it becomes a tedious task to track all the people at the same time. This Application overcomes the above problems & shows the live location of each user connected with each other in the Application in just a single click!

**1.2 Objective**

1. Get live location of connected users.

2. Share live location of connected users with eachother.

**1.3** **Purpose & Scope**

**1.3.1 Purpose**

This Android Application provides facility to track Friends or Family or People travelling in groups towards a common destination. However, it can also be used by a user to track their loved ones to check if they reached their destination. User has the privilege to add, remove users from his/her account. User can also create a room & add other users to it. The people in the room can view each other’s location. Users can leave the room at any time if they wish too.

**1.3.2 Scope**

Scope of the project is very broad.

They are :-

* This app can be widely used by Bike Riders as they usually travel in groups.
* Also it can be us8ed by anyone anywhere anytime.

**1.3.3 Applicability**

**1.4 Achievements**

**1.5 Organisation of Report**

# 2. SURVEY OF TECHNOLOGIES

Available technologies for developing the proposed system are as follows :

* C++
* C# (Xamarin)
* Html 5
* Hybrid Applications
* Java
* Kotlin
* **C++ :**
  + C++ is a middle-level programming language which can be used to develop Android Applications.
  + Java, with the JVM-optimized byte-code, can generate pretty fast code, but native (i.e., machine code) can be faster and useful in areas such as gaming, physics simulations and signal processing.
  + As C++ usually has no standard user Interface, the user-interface code is written in the native language and C++ used for the business logic.
  + C++ has a smaller memory footprint, as it is nearer to the metal and has no garbage collection.
  + C++ is a superset of C and compiles virtually all C programs, so it can reuse C software.

* C# (Xamarin) :
  + C# is a programming language developed by Microsoft which can be used to develop Android Applications.
  + Xamarin, a Microsoft owned software company has created a Cross Platform development tool which enables developers to develop iOS and Android apps in C# language.
  + Xamarin is offered in different licenses from free to enterprise levels.
  + The beauty of Xamarin is that despite the differences under the hood, Xamarin.iOS and Xamarin.Android (coupled with Microsoft’s Windows SDKs) offer a seamless experience for writing C# code that can be re-used across all three platforms.
  + Business logic, database usage, network access, and other common functions can be written once and re-used on each platform

* **HTML 5 :**
  + A HTML5 app refers to a mobile app built completely using HTML, CSS and Javascript only.
  + HTML5 apps are web apps and they must be run using the underlying OS browser.
  + A well written HTML5 app can be used even when the device is offline, or at the very least, show an error message.
  + HTML5 apps are portable across different OSes and device types.
  + HTML5 apps are generally cheaper to develop and maintain than native apps.
* **Hybrid Applications :**
  + Hybrid apps are built using on language/framework like HTML5, CSS and Javascript and are then wrapped with native specific code for each desired mobile OS.
  + A hybrid app is no different from a native app.
  + Hybrid apps can be made available and distributed via the relevant app store, just like native apps.
  + Hybrid apps have greater access to the native hardware resources than plain HTML5 apps, usually through the corresponding framework’s own APIs.
  + Popular hybrid app frameworks include Apache Cordova (formerly PhoneGap), Appcelerator Titanium, Appear IQ, CocconJS and Appzillon among others.
* **Java :**
  + Java is a Programming Language developed by James Gosling at Sun Microsystems.
  + Java is the official programming language for Android app development.
  + It is the most widely used programming language for android application development.
  + Java itself is used by Google for large parts of the Android internals.
  + Java has many frameworks and classes for features like networking, threading, IO operations and thus, programmers can leverage these qualities in their apps.
* **Kotlin :**
  + Kotlin is a statistically typed programming language that runs on the Java virtual machine.
  + Kotlin is a Java-based programming language and interoperable.
  + As of now, Koltin is an official Android Programming Language.
  + Kotlin is easy & simple to use.
  + Kotlin is crisp, concise, and reduces a lot of much of the boilerplate code.

**The Proposed Android Application will use Java-Kotlin :**

**Why Java-Kotlin ?**

Both Java & Kotlin are now the official languages for Android Development. While Java is the oldest & most widely used language for Android Development, Kotlin being a newly developed language has taken the world of Android by storm. Kotlin is a Java based programming language which runs on JVM (Java Virtual Machine). Hence it is interoperable with Java. Because Kotlin generates Java bytecode, we can use our favorite Java frameworks and libraries in Kotlin. Kotlin is much more productive, less boilerplate code, concise & so it can be used in many areas of the project. And well, what better place to start learning Kotlin than this, eh?

**Advantages of Native Development over Xamarin :**

* Unlike Native Languages, Xamarin has slightly delayed support for the latest platform updates since its impossible for third-party tools to provide the immediate support for the latest Android releases.
* Native development makes extensive use of open source technologies. With Xamarin, you have to use only the components provided by the platform and some .Net open source resources.
* Xamarin community is significantly smaller than those of Android Native Community.
* When using Xamarin.Android to build mobile apps with truly native look and feel, we still need to write a platform-specific layer of code. Thus, at least a basic knowledge of native technologies (Java/Kotlin for Android) is required.
* Xamarin’s main benefit is the ability to share our code across the platforms. Yet, we can only share the logic, UI code will be mostly platform-specific. This makes building games, rich custom UI, or complex animations in Xamarin pretty pointless.
* Depending on their type and complexity, Xamarin apps are typically larger than native ones.