MINI PROJECT
(2022-23)

"Track My Vehicle"
Project Report

Institute of Engineering & Technology

Submitted By -

Harshil Gupta (191500321)
Harsh Tripathi (191500320)
Nishkarsh Jain (191500513)
Hemant Sharma (191500340)

Under the Supervision Of

Mr.Vinay Agrawal

Assistant Professor

Department of Computer Engineering & Applications

Department of Computer Engineering and Applications

GLA University, 17 km. Stone NH#2, Mathura-Delhi Road,

Chaumuha, Mathura – 281406 U.P (India)

Declaration

I/we hereby declare that the work which is being presented in the Bachelor of technology. Project **"Track My Vehicle"**, in partial fulfillment of the requirements for the award of the *Bachelor of Technology* in Computer Science and Engineering and submitted to the Department of Computer Engineering and Applications of GLA University, Mathura, is an authentic record of my/our own work carried under the supervision of **Mr.Vinay Agrawal,Assistant Professor, Dept. of CEA,GLA University.**

The contents of this project report, in full or in parts, have not been submitted to any other Institute or University for the award of any degree.

Sign: *Harsh Tripathi*

Name of Candidate: Harsh Tripathi

University Roll No.: 191500320

Sign: *Harshil Gupta*

Name of Candidate: Harshil Gupta

University Roll No.:191500321

Sign: *Hemant Sharma*

Name of Candidate: Hemant Sharma

University Roll No.:191500340

Sign: *Nishkarsh Jain*

Name of Candidate: Nishkarsh Jain

University Roll No.:191500513

Department of Computer Engineering and Applications

GLA University, 17 km. Stone NH#2, Mathura-Delhi Road,

Chaumuha, Mathura – 281406 U.P (India)

Certificate

This is to certify that the project entitled "**Track My Vehicle**", carried out in Mini

Project
– I Lab, is a bonafide work by Harsh Tripathi, Harshil Gupta, Nishkarsh Jain, Hemant Sharma and is submitted in partial fulfillment of the requirements for the award of the degree Bachelor of Technology (Computer Science & Engineering).

Signature of Supervisor:

Name of Supervisor: Mr. Vinay Agrawal

Date:

Training Certificates

- **Harsh Tripathi**

· Harshil Gupta

CERTIFICATE OF COMPLETION

The Complete 2021 Flutter Development Bootcamp with Dart

Instructors **Dr. Angela Yu**

Harshil Gupta

Date **Nov. 21, 2021**
Length **29 total hours**

ACKNOWLEDGEMENT

Presenting the ascribed project paper report in this very simple and official form, we would like to place my deep gratitude to GLA University for providing us the instructor Mr. Vinay Agrawal, our supervisor.

He has been helping us since Day 1 in this project. He provided us with the roadmap, the basic guidelines explaining how to work on the project. He has been conducting regular meetings to check the progress of the project and providing us with the resources related to the project. Without his help, we wouldn't have been able to complete this project.

And last but not least we would like to thank our dear parents for helping us to grab this opportunity to get trained and also my colleagues who helped me find resources during the training.

Thanking You

Sign: *Harshil Gupta*
Name of Candidate: Harshil Gupta
University Roll No.:191500321

Sign: *Harsh Tripathi*
Name of Candidate: Harsh Tripathi
University Roll No.: 191500320

Sign: *Hemant Sharma*
Name of Candidate: Hemant Sharma
University Roll No.:191500340

Sign: *Nishkarsh Jain*
Name of Candidate: Nishkarsh Jain
University Roll No.:191500513

ABSTRACT

In this project, we created a hardware device and a mobile application, basically, a GPS tracking device which we have named Track My Vehicle. This device will provide us a platform to access device location at the ease of our fingertips. All the users will be having their separate accounts on this app which will be connected to their email id.

Track My Vehicle is a combination of different IOT devices and mobile application which helps user to find its vehicle from anywhere in the world. To track the vehicle user first have to install hardware device in victim's vehicle and then install Track My Vehicle application and then enter credentials in application after that he/she can easily see your device where it is now. This is improved security systems for vehicles. The latest like GPS are highly useful now a days, this system enables the owner to observe and track his vehicle and find out vehicle movement and its past activities of vehicle.

The Android App ecosystem is diverse and is changing people's lives all over the world. Android users are expected to increase because of the advanced changes of the operating system and the way it deals with issues and compatibility with other mobile devices. Furthermore, designing solutions for the problems that we may face in the future is essential. Like this application definitely stands the need of students at any time at their fingertips without any barrier of place**.**

CONTENTS

CHAPTER-1 INTRODUCTION

1.1 CONTEXT

"**Track My Vehicle**" has been submitted in partial fulfillment of the requirements for the award of the degree of Bachelor of Technology in Computer Science and Engineering at GLA University, Mathura supervised by Mr. Vinay Agrawal. This project has been completed approximately one months and has been executed in modules, meetings have been organized to check the progress of the work and for instructions and guidelines.

1.2 MOTIVATION

In recent years, we have realized the importance of GPS tracking device and how important it is for us to have our resources online.

In the century we are living in, the world is progressing at a really great pace, a lot of technologies come up every single day. To keep up with technology is also important to survive in this world of digitalization and learning. Along with this, we need to have a place to keep the resources for areas of our interest so we thought of developing a GPS device that could track location of vehicle in real-time as well as a platform where we could keep the track of our vehicle.

1.3 OBJECTIVE

The main objective to build this application is to provide a user a satisfied environment by which he/she can easily detect there vehicle with real-time and exact data . In This project we use Google API which helps user to create exact route device and show necessary information like time to cover distance, speed, direction, best route to choose, traffic condition on different route. This is improved security systems for vehicles. The latest like GPS are highly useful now a days, this system enables the owner to observe and track his vehicle and find out vehicle movement and its past activities of vehicle.

1.4 EXISTING SYSTEM

In present there are many existing fitness apps which are in use.
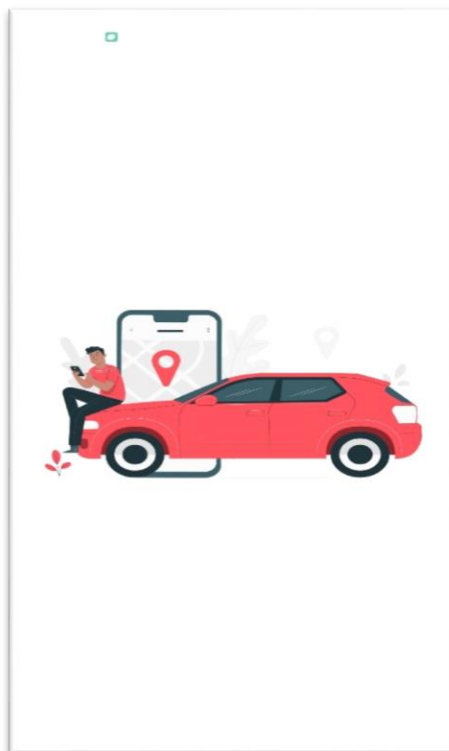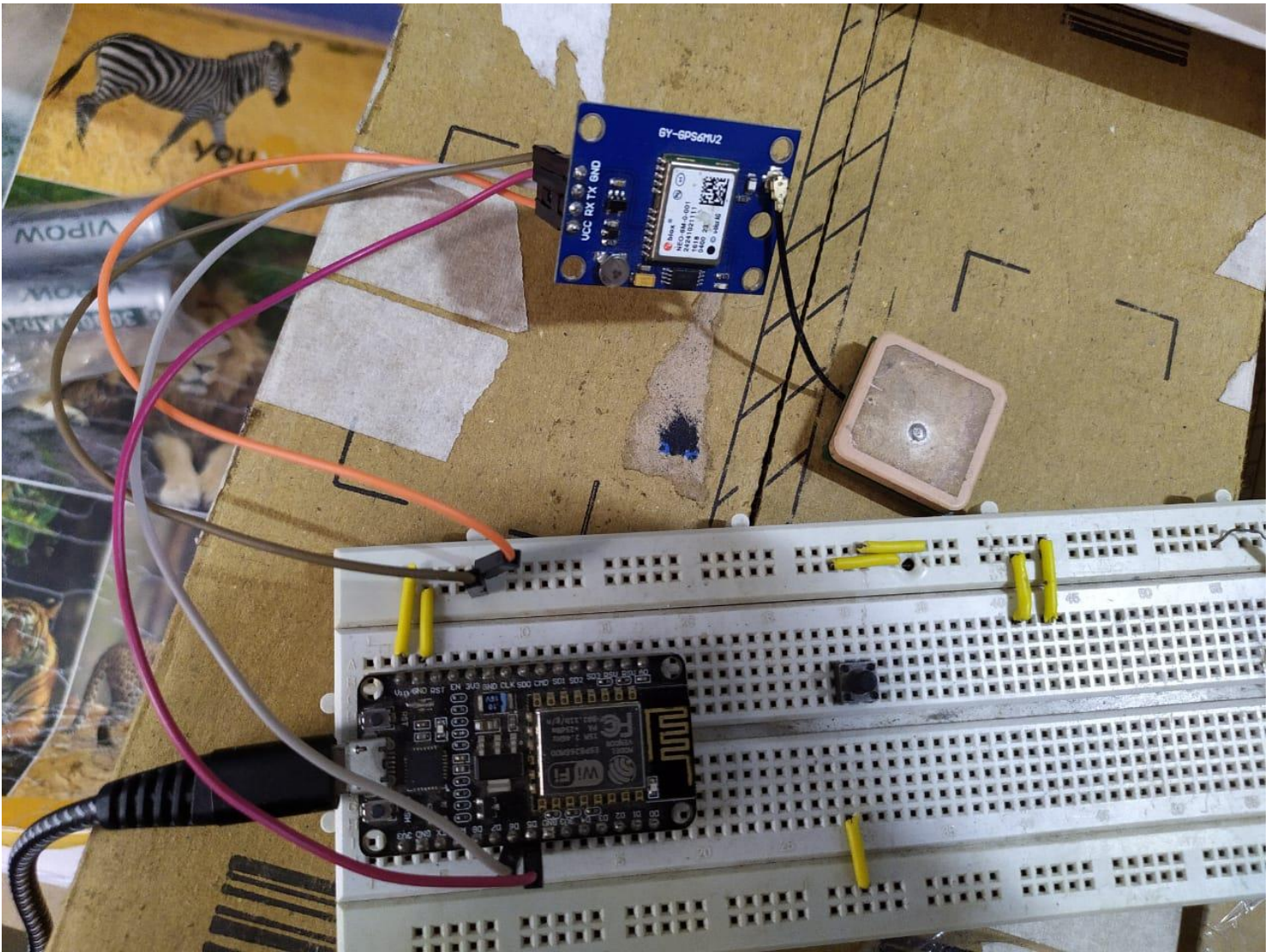
(a) (b)

Figure-1: Existing System

1.5 SOURCES

The source of our project (including all the project work, documentations and presentations) will is available at the following link

https://github.com/prog-harsh/TrackMyVehicle

CHAPTER -2

SOFTWARE REQUIREMENT ANALYSIS

2.1 IMPACT OF GPS Device ON DAILY LIFE

This new technology, popularly called vehicle Tracking Systems which created many wonders in the security of the vehicle. This hardware is fitted on to the vehicle in such a manner that it is not visible to anyone who is inside or outside of the vehicle. Thus it is used as a covert unit which continuously or by any interrupt to the system, sends the location data to the monitoring unit.

A Program has been developed which is used to locate the exact position of the vehicle and also to navigated track of the moving vehicle on Google Map.

2.2 PROBLEM STATEMENT

"Track My Vehicle" is a combination of IOT device and mobile application which helps user to track his/her vehicle from any where at any time, user can also choose different routes according to his/her needs and also live track the traffic condition on that route and etc.

2.3 HARDWARE AND SOFTWARE REQUIREMENTS

Hardware Requirement

- Processor : inteli5

- Operating System : Any Operating System

- RAM : 8 GB (or higher)

- Hard disk : 256GB

Software Requirement

- Software used : Android Studio / VS Code / Arduino IDE

- Language used : Dart, Flutter (Framework),C++

- Database : Firebase
· User Interface Design : Android Application

2.4 MODULES AND FUNCTIONALITIES

▪ **Splash Screen**: The first screen with which the user interacts will be this screen containing the logo and the app name .This will disappear within 3 seconds after the app is displayed.

▪ **Login Page**: This page is for the user to login into the application using google sign-in.

Main Page: This page is the root of the whole application. This entire app revolves around this screen. This screen would allow users to see what exact location of vehicle device and will take them to that desired page or screen.

2.5 Track My Vehicle ON ANDROID APPLICATION

To build this application is to provide a user a satisfied environment by which he/she can easily detect there vehicle with real-time and exact data . In This project we use Google API which helps user to create exact route device and show necessary information like time to cover distance, speed, direction, best route to choose, traffic condition on different route. This is improved security systems for vehicles. The latest like GPS are highly useful now a days, this system enables the owner to observe and track his vehicle and find out vehicle movement and its past activities of vehicle.

CHAPTER- 3 SOFTWARE DESIGN
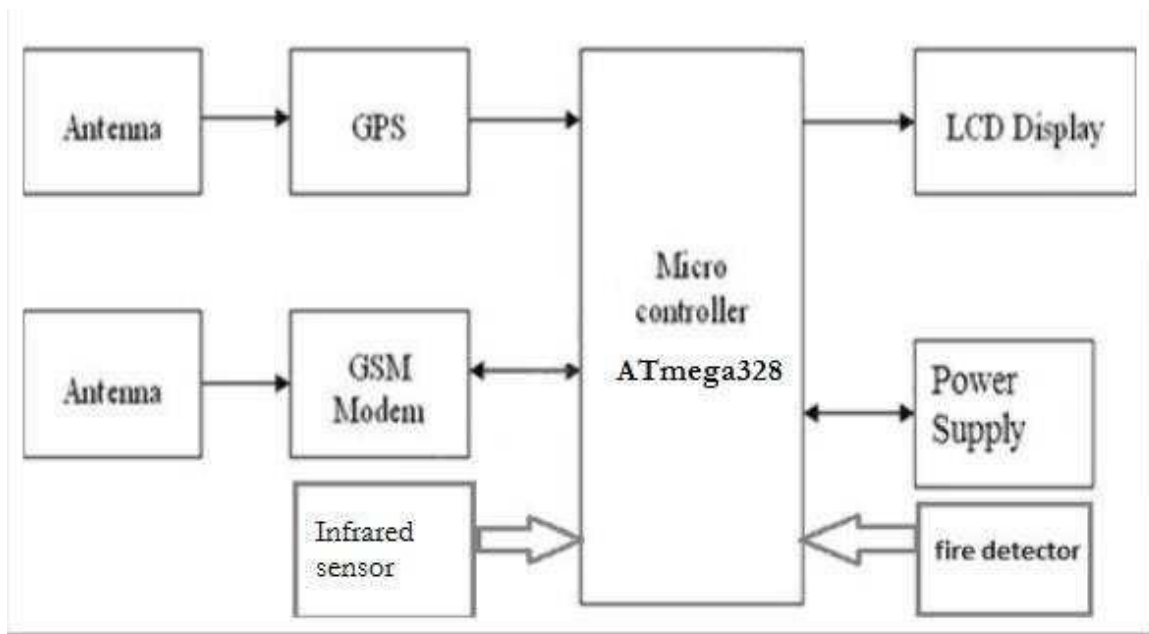
3.1 USE-CASE DIAGRAM:

Figure-2: Use–Case Diagram

Tracking in India is mainly used by transport systems, taxi companies, traffic operators. Taxi operators use this to estimate how far the vehicle is from a particular area and send this information to call centers and they can inform general public about the distance of the taxi location and time it takes tom come to them. Another use is for traffic police if this system is located in every vehicle they can estimate the traffic by looking on the map and if any accident is detected then they can route the traffic in to another way. This is how tracking is useful because India is one of busy traffic countries and this system can control many of the traffic problems.

.

CHAPTER-4
TECHNOLOGY USED

4.1 Flutter

We are using Flutter to develop "Track My Vehicle". The core concept of the Flutter framework is In Flutter, Everything is a widget. Widgets are basically user interface components used to create the user interface of the application.

In Flutter, the application is itself a widget. The application is the top- level widget and its UI is built using one or more children (widgets), which again build using its children widgets. This composability feature helps us to create a user interface of any complexity. Flutter is an open-source UI software development kit created by Google. It is used to develop cross platform applications for Android, iOS, Linux, Mac, Windows, Google Fuchsia, and the web from a single codebase.

First described in 2015, Flutter was released in May 2017.

The core concept of the Flutter framework is In Flutter, Everything is a widget. Widgets are basically user interface components used to create the user interface of the application.

In Flutter, the application is itself a widget. The application is the top- level widget and its UI is built using one or more children (widgets), which again build using its children widgets. This composability feature helps us to create a user interface of any complexity.

4.2 Types of Application

· **Native Apps:** An executable program coded in the machine language of the hardware platform it is running in. **Native applications** are compiled into the machine language of that CPU. For example, **Windows** and Mac executable **apps** are in x86 machine language, while **mobile apps** are ARM based. Native apps are the most common. They're coded in a specific language like Swift for **iOS** or Java for Android. A popular example is WhatsApp.

· **Web Apps:** are accessed via the internet browser and will adapt to whichever device you're viewing them on. They are not native to a particular system, and don't need to be downloaded or installed. Due to their responsive nature, they do indeed look and function a lot like mobile apps
— and this is where the confusion arises.

· **Hybrid Apps:** Hybrid apps are deployed in a native container that uses a mobile Web View object. When the app is used, this object displays web content thanks to the use of web technologies (CSS, JavaScript, HTML, HTML5).It is in fact displaying web pages from a desktop website that are adapted to a Web View display. The web content can either be displayed as soon as the app is opened or for certain parts of the app only i.e. for the purchase funnel. In order to access a device's hardware features (accelerometer, camera, contacts...) for which the native apps are installed, it is possible to include native elements of each platform's user interfaces (iOS, Android): native code will be used to access the specific features in order to create a seamless user experience. Hybrid apps can also rely on platforms that offer JavaScript APIs if those functionalities are called within a WebView

4.3 VERSION OF ANDROID

Codename	Version	API level/NDKrelease
Android12	12	API level31

Android11	11	API level30
Android10	10	API level29
Pie	9	API level28
Oreo	8.1.0	API level27
Oreo	8.0.0	API level26
Nougat	7.1	API level25
Nougat	7.0	API level24
Marshmallow	6.0	API level23
Lollipop	5.1	API level22
Lollipop	5.0	API level21
KitKat	4.4 -4.4.4	API level19
Jelly Bean	4.3.x	API level18
Jelly Bean	4.2.x	API level17
Jelly Bean	4.1.x	API level16

Ice CreamSandwich	4.0.3 -4.0.4	API level 15, NDK 8
Ice CreamSandwich	4.0.1 -4.0.2	API level 14, NDK 7
Honeycomb	3.2.x	API level13
Honeycomb	3.1	API level 12, NDK 6
Honeycomb	3.0	API level11
Gingerbread	2.3.3 -2.3.7	API level10
Gingerbread	2.3 -2.3.2	API level 9, NDK 5
Froyo	2.2.x	API level 8, NDK 4
Eclair	2.1	API level 7, NDK 3
Eclair	2.0.1	API level6
Eclair	2.0	API level5
Donut	1.6	API level 4, NDK 2
Cupcake	1.5	API level 3, NDK 1
(no codename)	1.1	API level2

(no codename)	1.0	API level1

Each year Android releases a new version with better features, better security and better User Interface experience and a new symbol. Here is the table of list of versions.
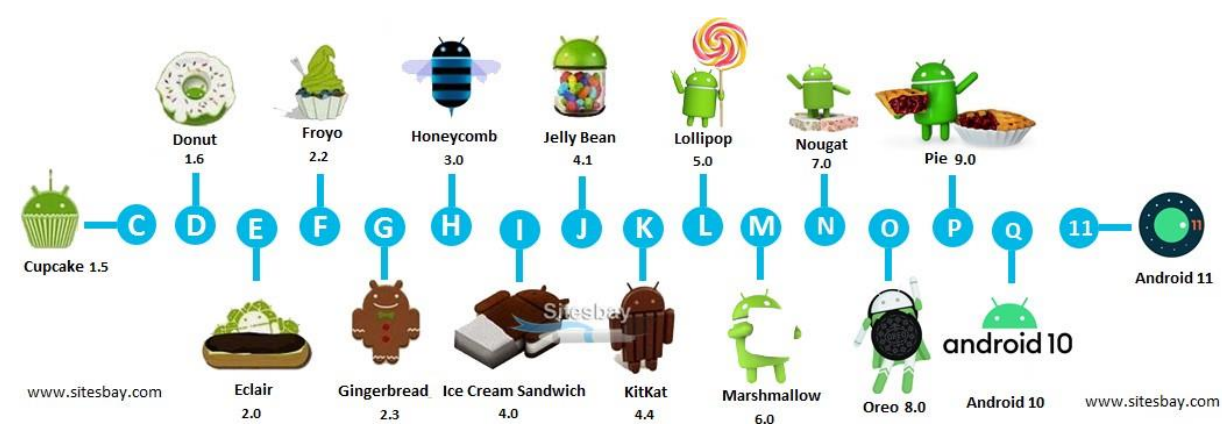

Figure-5: **Android Kitkat**

Table -1: Versions of Android

1

4.4 TOOLS AND LANGUAGES

Tools used to build the Android App are:-

· **Android Studio**: Android Studio is an environment that help us create and edit Android applications. It is the official IDE for Android App Development. It has intelliJ's powerful code editor and developer tools and various features that enhance productivity while developing apps.

· **Visual Studio Code**: Visual Studio Code is a streamlined code editor with support for development operations like debugging, task running, and version control. It aims to provide just the tools a developer needs for a quick code-build-debug cycle and leaves more complex workflows to fuller featured IDEs, such as Visual Studio IDE.

· **Software Development Kit (SDK)**: Android Studio requires a collection of libraries and data therefore SDK is mandatory.

Languages used in building an Android Application are classified as per the Front End and Back End. For designing the Front End of an application we have used XML and for designing the Back End we have used Kotlin.

· **Dart**: The Dart language is type safe; it uses static type checking to ensure that a variable's value *always* matches the variable's static type. Sometimes, this is referred to as sound typing. Although types are mandatory, type annotations are optional because of type inference. The Dart typing system is also flexible, allowing the use of a dynamic type combined with runtime checks, which can be useful during experimentation or for code that needs to be especially dynamic.
Dart offers sound null safety, meaning that values can't be null unless you say they can be. With sound null safety, Dart can protect you from null exceptions at runtime through static code analysis. Unlike many other null-safe languages, when Dart determines that a variable is non-nullable, that variable is *always* non-nullable. If you inspect your running code in the debugger, you'll see that non-nullability is retained at runtime (hence *sound* null safety).

. **Arduino IDE**

The Arduino IDE is a cross-platform application written in Java, and is derived from the IDE for the Processing programming language and the Wiring project. It is designed to introduce programming to artists and other newcomers unfamiliar with software development. It includes a code editor with features such as syntax highlighting, brace matching, and automatic indentation, and is also capable of compiling and uploading programs to the board with a single click. There is typically no need to edit make files or run programs on a command-line interface. Although building on command-line is possible if required with some third-party tools such a Ino.

The Arduino IDE comes with a C/C++ library called "Wiring" (from the project of the same name), which makes many common input/output operations much easier. Arduino programs are written in C/C++.

4.5 BASIC TERMINOLOGY

• **Widget:** Widgets are the fundamental UI unit in Flutter. They represent an interactive element of a user interface such as a button or text field. Widget classes have their own build function to create a tree structure for drawing into the parent's Build Context, which is used to manage layout calculations and coordinate with native platform elements at runtime.

• **StatelessWidget**: Basically a stateless widget is one that doesn't store any internal data about the user's interaction with it. It may read its information from props passed in by parent widgets or other sources like local or session storage etc., but does not maintain its own separate set of values for things such as text content entered into an input field.

• **StatefulWidget**: A stateful widget is one that stores data between invocations of its build method, and updates the UI in response to those changes.

• **Emulator**: An emulator is an Android virtual device through which you can select the target Android version or platform to run and test your developed application.

• **Manifest file**: Manifest file acts as a metadata for every application. This file containsall the essential information about the application like app icon, app name, launcher activity, and required permissionsetc.

• **MaterialApp:** Material apps are a set of predefined widgets which implement Google's material design guidelines on both mobile and desktop platforms. The Material App provides prebuilt implementations for common user interface patterns such as lists, grids, menus etc., so you can focus more time on building your app rather than reinventing the wheel!

• **Scaffold:** Scaffolds let you quickly add screens with basic content into your application without writing any code at all by using a templating engine called Stencils. You will need this when developing multiple screen applications (commonly referred to as multi-page applications).

APK: Short for "Android application package." The extension used in Android app installation files (e.g., app.apk). Similar in nature to an EXE file on Windows.

• **SDK:** Short for "Software Development Kit." As it pertains to Android, the SDK is a set of tools such as code libraries, a debugger, and a handset emulator that can be run on Windows, Mac, or Linux to facilitate the creation of Android apps by developers. While the SDK is generally intended for use by developers, end users can install the software on their home computer to execute ADB and Fast boot commands.

• **App Bar**: The app bar is an important design element, usually at the top of each screen in an app that provides a consistent familiar look between Android apps. It is used to provide better user interaction and experience by supporting easy navigation through tabs and drop-down lists.

• **Navigation bar**: Navigation Drawer is a sliding left menu that is used to display the important links in the application. Navigation drawer makes it easy to navigate to and fro between those links. It's not visible by default and it needs to open either by sliding from left or clicking its icon in the App Bar.

• **BuiltContext**: Build Context refers to an object that provides information to the widget tree as it renders. For example, layout calculations and coordinates with native platform elements at runtime are done based on Build Context objects.

• **Firebase** is a Backend-as-a-Service (Baas). It provides developers with a variety of tools and services to help them develop quality apps, grow their user base, and earn profit. It is built on Google's infrastructure. Firebase is categorized as a NoSQL database program,

which stores data in JSON-like documents. Firebase has three core services: a real-time database, user authentication and hosting. With the Firebase iOS SDK, you can use these services to create apps without writing any server code.

JSON stands for JavaScript Object Notation. It is an independent data exchange format and is the best alternative for XML. JSON is used for data interchange (posting and retrieving) from the server. Hence knowing the syntax and it's usability is important. JSON is the best alternative for XML and its more readable by human

CHAPTER -5

IMPLEMENTATION AND USER INTERFACE

Creating an app concept design with screen sketches and functional flow diagrams is the best way to communicate your vision to the mobile app developer. Making the concept clear to the developer is probably the most important factor in successful mobile app development. Yet it is one of the most common problems or obstacles in a mobile app development outsourcing project.

No matter what the marketing and profit goals are or if you are outsourcing an app for your personal use, you need to fully design and document the app concept if you expect a programmer to make your vision a reality. Developers are not mind readers and even descriptions given during conversations can be very fleeting or interpreted differently. Fully documenting your concept, therefore, leaves little to chance. The two most important things to do are: A) make a comprehensive description of how the app works and what it does (functionality) and B) create a comprehensive description of what the user sees and does (look and feel).

5.1 Implementation of the Track My Vehicle:

Implementation of Track My Vehicle is taking place in various phases. Firstly we build the login interface then login page and then make various layouts for the supporting features.

5.1.1 Step to be followed to develop the app:

1. Firstly we created the splash screen with animated text and images.

2. After that we create a login phase which consists of various phases that allows users to login into the app.▪

▪ **For authenticating the user we have used firebase authentication**.

3. Now, we are going to create the homepage of our application which consists of various widgets.

4. Now the main screen comes in picture here you track your device

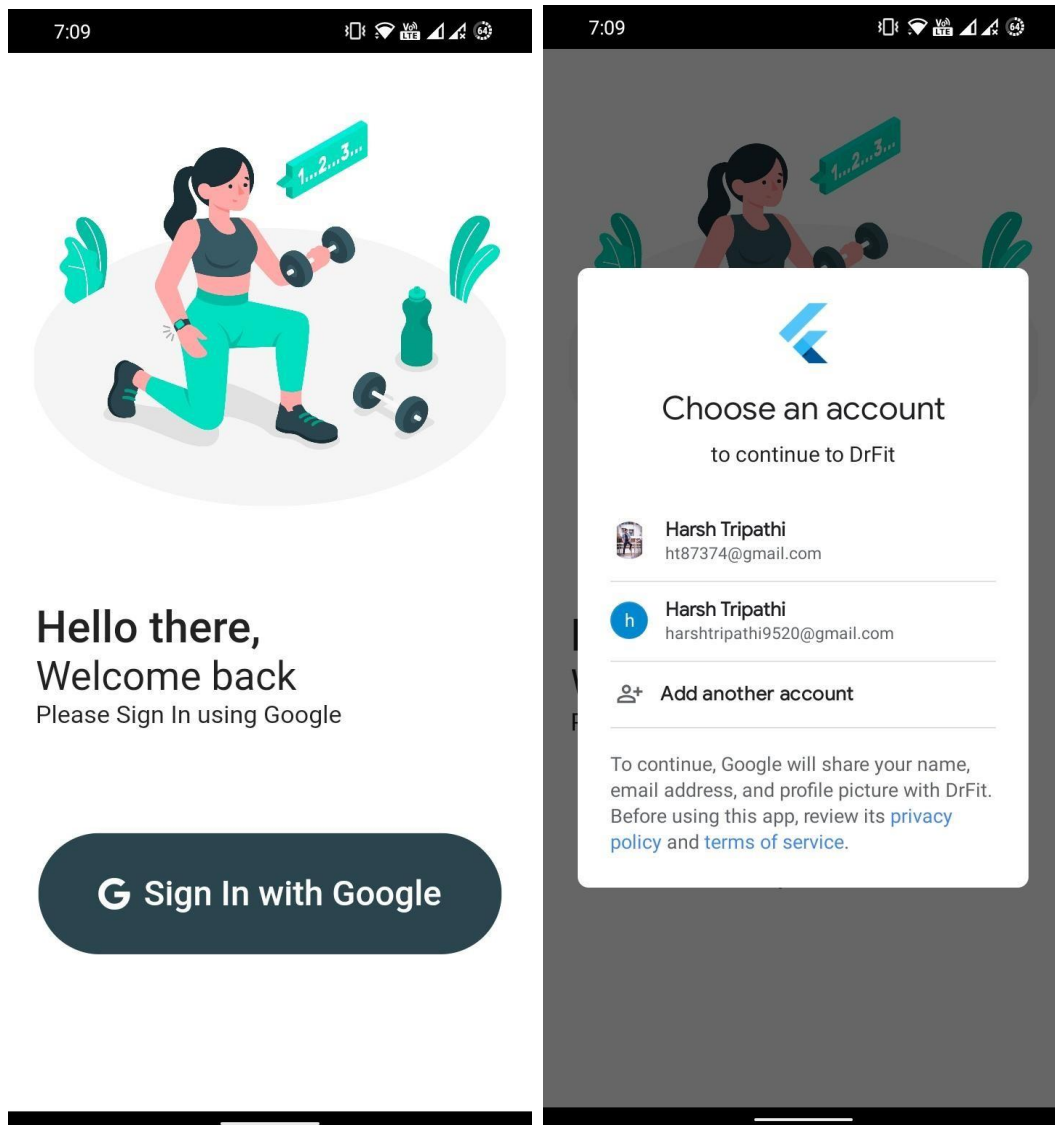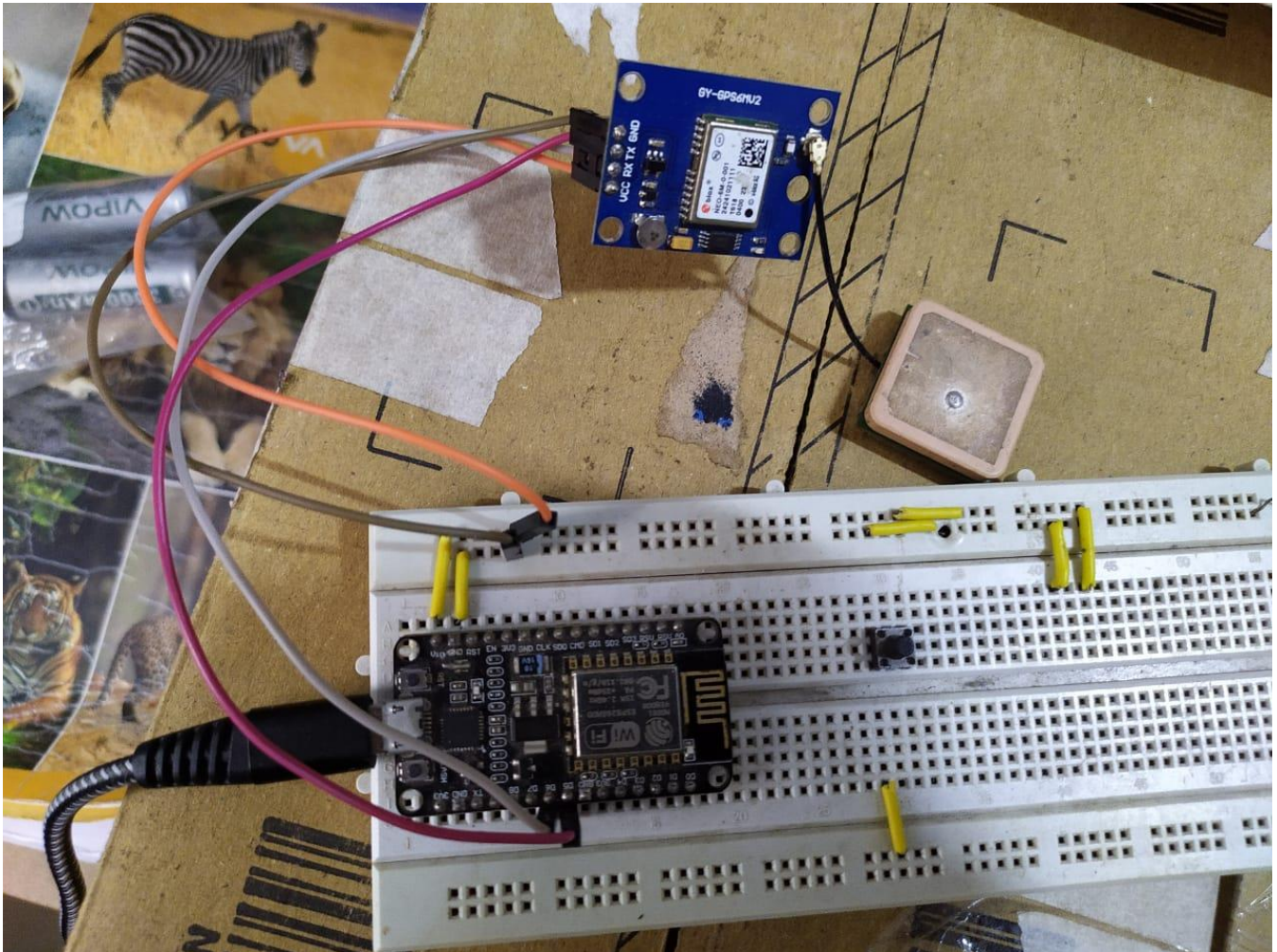# Figure-6: Flowchart for User
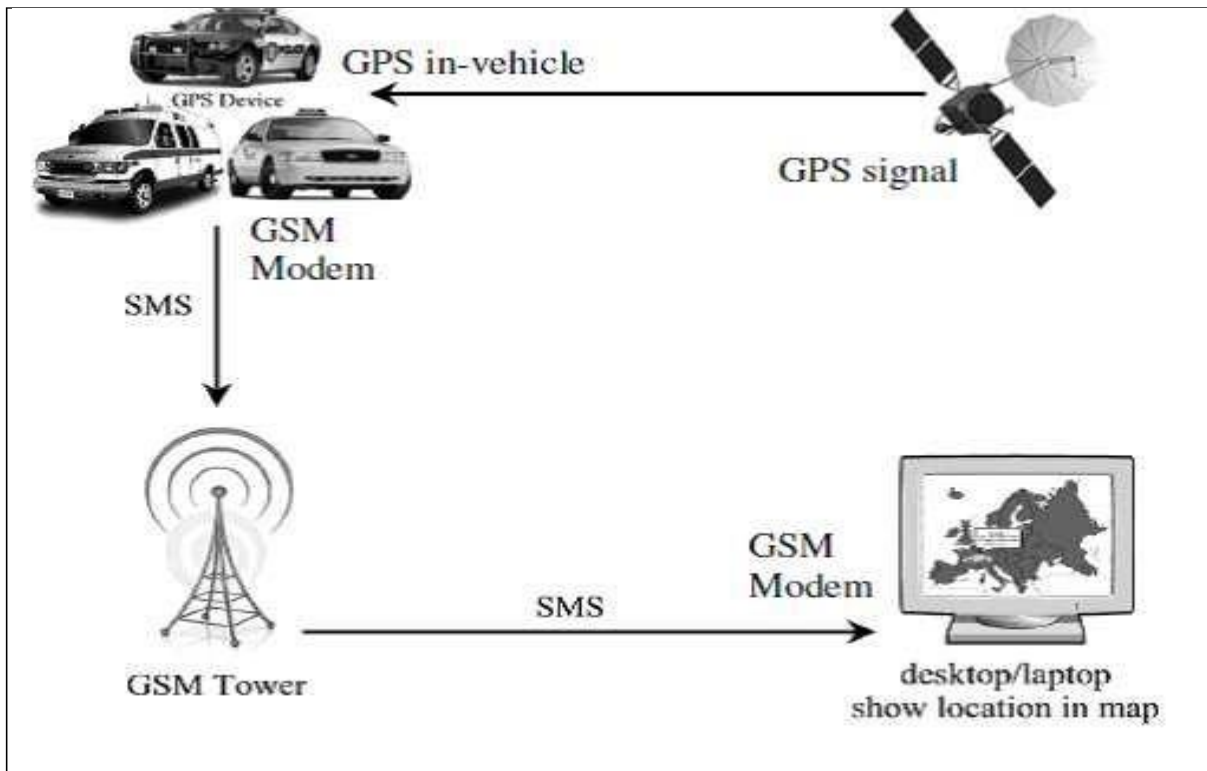
5.2 User Interface

Figure 6 Splash screen

Figure 7 Login Screen (a) (b)

Figure-8 Home Page

CHAPTER - 6 TESTING

Once source code has been generated, software must be tested to uncover as many errors as possible before delivery. It is very important to work the system successfully and achieve high quality software. Testing includes designing a series of test cases that have a high likelihood of finding errors by applying software-testing techniques.

System testing makes logical assumptions that if all the parts of the system are correct, the goal will be successfully achieved. The system should be checked logically. Validations and cross checks should be there. Avoid duplications of record that cause redundancy of data.

In other Words, Testing is the process of evaluating a system or its component(s) with the intent to find whether it satisfies the specified requirements or not. It is executing a system in order to identify any gaps, errors, or missing requirements in contrary to the actual requirements.

The Android framework includes an integrated testing framework that helps you test all aspects of your application and the SDK tools include tools for setting up and running test applications. Whether you are working in Eclipse with ADT or working from the command line, the SDK tools help you set up and run your tests within an emulator or the device you are targeting.

There are different types of testing some of them are listed below:

6.1 Installation Testing:

There are two types of apps on an Android device i.e., Pre-installed applications and the applications which are installed later by the user.

For both of the above, installation testing is carried out by our teammates. It is ensuring smooth installation of the application without ending up in errors, partial installation etc.

6.2 Unit Testing

It focuses on smallest unit of software design. In this we test an individual unit or groups of inter related units. It is often done by programmer by using sample input and observing its corresponding outputs. In this testing technique we are primarily focuses on

· Loop methods and functions are working fine or not.
· Misunderstood or incorrect Arithmetic precedence
· Incorrect Initialization

Unit Testing of the app:

Test cases	Description	Expected Outcome	Result
1	Start Page–Launch Screen	Should display splash screen with animated text	Pass
2	Login Screen	Should display a Login Screen through which you can log into the page.	Pass
3	Homepage	Should display homepage through which you can navigate to any screen	Pass

6.3 User Testing

User testing is the process through which the interface and functions of a website, app, product, or service are tested by real users who perform specific tasks in realistic conditions. The purpose of this process is to evaluate the usability of that website or app and to decide whether the product is ready to be launched for real users.

This app was tested by our teammates and friends who are using different mobile phones (and having different android version) also tested on different emulator to check its performance and it seems to be working fine and users of this app are satisfied with the facilities and performance of the app and like the way how the app is worked.

6.4 Performance Testing

In this type of testing we have checked the performances of our application under some peculiar conditions. Those conditions include:

· Low memory in the device.

· The battery is extremely low level.

· Poor/Bad network reception.
Performance is basically tested from 2 ends, application end, and the application server end. Our app is also performing well in this phase of testing as well. And we are getting positive feedback from users of our app.

6.5 Compatibility Testing
This application was tested and used on different devices like Realme C2, Google Nexus 4. The application worked fine and is stable. The application worked fine in portrait mode and there isn't any problem with compatibility.

On all types of testing (that we have performed above) our performing well on our app i.e. Track My Vehicle

CHAPTER -7 CONCLUSION

Vehicle tracking system makes better fleet management and which in turn brings large profits. Better scheduling or route planning can enable you handle larger jobs loads within a particular time. Vehicle tracking both in case of personal as well as business purpose improves safety and security, communication medium, performance monitoring and increases productivity. So in the coming year, it is going to play a major role in our day-to-day living.

PROJECT LINK:- https://github.com/prog-harsh/TrackMyVehicle

REFERENCES

1. Introduction to Android:
http://developer.android.com/guide/index.html

2. *Android API:*

http://developer.android.com/reference/packages.html

3. *Android User Interfaces:*

http://developer.android.com/guide/topics/ui/index.html

4. *Layout:*

http://developer.android.com/guide/topics/ui/declaring-layout.html

5. *Android Training:*

http://developer.android.com/training/index.html

6. Internshala:

https://trainings.internshala.com/

7. Android developer Guide:

https://developer.android.com

/

8. For rectifying the error :

https://stackoverflow.com/