ABSTRACT

Navigation is a methodology that focuses on the process of tracking and regulating the movement of people, vehicles, and craft from one location to another, such as land navigation, marine navigation, and aeronautic navigation. The campus navigator is an Android smartphone application that is used to navigate routes inside any campus premises, such as a mall, college, or hospital. Mobile phones are now becoming more than just communication devices. Smartphones, in particular, are products that make our jobs and daily lives easier. The use of mobile apps has exploded in recent years, owing to advancements in technology and the popularity of these devices. Location- based applications of augmented reality views are also possible using new techniques such as GPS, cameras, compass, and accelerometer, which can be used to calculate the direction of the user. There are many commercial navigation apps that provide users with directions from one location to another, such as Google Maps, Yahoo Maps, and Mapquest. These applications, on the other hand, must search for existing roads and are unable to provide routes as precise as an on-campus path would require. The primary goal of this paper is to present the results of research and development of an intelligent device for cell phones that can assist users in determining the shortest path between two points of interest on campus. The aim of the research is to find the best navigation solution with the shortest path calculation as the primary goal. An Android- based mobile application that allows users to quickly and easily locate a spot. According to preliminary assessment, the proposed methods and mobile application assist users in quickly locating their point of interest. The application's main features include a voice-based information provider, a location identifier, and a map-based route selection function that allows users to choose the best path to a specific destination within the premises.

TABLE OF CONTENT

LIST OF FIGURES

LIST OF ABBREVIATIONS

DOC	Document
API	Application Programming Interface

CHAPTER 1

INTRODUCTION

INTRODUCTION

In recent years the mobile has become the valuable part of the human beings. It is necessary for human begins to have a powerful device which will provide all the facilities other than basic facility available in mobile phones. Android provide such functionality which enables the developers to design such applications which will make a simple mobile to smart one. “Android is built on the open Linux Kernel. Furthermore, it utilizes a custom virtual machine that has been designed to optimize memory and hardware resources in a mobile environment. Android is open source it can be liberally extended to incorporate new cutting edge technologies as they emerge. The platform will continue to evolve as the developer community works together to build innovative mobile applications. A campus is a complex infrastructure. Especially new students and people who are on it for the first time have a hard time to orientate themselves and find places. The campus occupies more than two square kilometres and thus is even larger than that. The campus has many different buildings. Most of the buildings are connected to each other, some of them even by underground walkways. Even if there are maps at some points on the campus, users do not have continuous help to get to their destination. They can try to figure out a way to get to their target on these static maps, but as soon as they start walking in the target direction they have no help any more. Whereas it is very common to use navigation systems in cars to reach designated locations, systems for pedestrian navigation are quite hard to find. So, how it is possible to help freshmen and other inexperienced people orientate themselves in the campus and support them finding places on campus with the help of modern techniques. In recent decenniums, mobile contrivance magnification has progressed significantly with deference to recollection, advanced computing capacity and higher data transfer speeds. Most

students, faculty, and staff now use Android telephones for their own personal use. A map application predicated on the Ecumenical Situating

System (GPS) would be subsidiary to locate the target location and potential route from the genuine location. The GIS is the heart of LBS to have all of LBS's valuables. Geographical Information Solutions People can monitor their own position and navigate very facilely from one place to another. There are many technologies to track position, such as cell identification, GPS, different systems for radio location, accelerometers and electronic compass etc. Compared to other methods, GPS provides much higher precision latitude and longitude. There are many apps and commercial contrivances that include directions and navigation, such as Google Navigation [Google Maps] and Magellan navigation contrivances [Magellan Perspicacious GPS]. Position tracking techniques can be incorporated with astute phones that will operate with sundry networks such as GSM (Ecumenical System for Mobile Communication) and GPRS (General Packet Radio Accommodation). With the fortification of Google Maps on GPS-enabled Android contrivances, this navigation became much more facile. Users may use GPS applications to locate a destination predicated on their current location. As a result of the coalescence of Google Maps and GPS, location probing has become an incipient trend. It has several supplemental features, such as highlighting congested routes, preserving time and energy while travelling to an unfamiliar location, so we will develop a system that will monitor, provide information about upcoming and perpetual events, and notify users if the events change. We demonstrated our system utilizing Google Maps and a GPS-enabled Android smartphone, which provided information on roads, traffic, and how to get to the destination. The primary goal of this research is to provide an optimal navigation solution through mobile applications that can exhibit a shortest-path calculation that benefits everyone.

EXISTING SYSTEM

The use of Google Maps for location-based navigation on the Android platform is

becoming more common. The researchers have produced an application called Guide-My-Tour [8]. This chart shows and monitors a user's current position and direction of travel. It can be rotated in any direction and zoomed in and out. It

combines a locality's conventional paper map picture with a satellite map image. Although user walking information was updated on a regular basis in response to the user's movements.

PROPOSED SYSTEM

A university campus may be very large or it may have many campuses. Every year lots of new students get admitted in the university. Many new buildings are built, new courses are started and some departments may be relocated inside the campus. There are no facilities to find places like administrative building, departments, library, canteen, etc. in the campus and how to find those places from current location. It creates problem to the new comer to reach easily and timely in the desired location. The new faculty member, staff and visitors also face same problem inside campuses. The Global Positioning System (GPS), Wireless-Fidelity (Wi-Fi), Bluetooth, and other technology are now built into mobile phones. The data from the Global Positioning System (GPS) is used to give the mobile device location information. When map-based services are needed, “Google Maps” is used. The demand for location-based services (LBS) is growing in parallel with the rapid growth of the smart phone industry. Android operating system is the most widely used operating system and it is dominating the smartphone market with its huge presence. As GPS works very accurately in large range so we have chosen GPS technology for location tracking. It can be used by existing university students, faculty members, and staff and also by parents, visitors. Mobile smartphones based on android operating system is the most easily accessible device and technology that every person has. Thus android phone can be served for a variety of technological purposes. Many tertiary institutions around the world have taken the lead in researching and developing a campus navigation system. At first, the user's app requests Google API from a GPS satellite. The map is then loaded into the user's app. The current location of the user is monitored and shown on a map using a GPS activated system on the user's mobile phone. At

first, the application in the user's phone, requests for Google API from Google server. Then the map is loaded
on user's phone. By GPS of users mobile, current location of user is tracked and

displayed on map. A HTTP GET request is sent through the cellular data network services and internet to the Map Information Server (MIS). A cellular data network is used to send a User-app message. A mobile application that runs on Android platform will be created to solve the problem. The mobile application comprises the following features:

• Function to select initial position (Point A) and final destination (Point B) Scenario: If a student wants to go point A from point B it displays shortest path between Point A and Point B Scenario: The application searches for all possible paths, the shortest path among all possible paths that direct to B from A. It displays the shortest path on map.

MainActivity class extends ***FragmentActivity*** class and implements ***LocationListener*** interface to view map and user's own location on map. ***onLocationChanged*** method of ***LocationListener*** interface is implemented to track user move ment with a 'ML' marker. The user's location is sensed by the GPS of user's mobile. Three inner classes have been taken inside the ***MainActivity*** class, which extends ***AsyncTask<String, Void, String>*** class. First is for loading and showing event information on map application with its proper position. Second is for loading and showing new location information on map application with its proper position. Third is for loading and drawing route on map. ***AsyncTask<String, Void, String>*** is extended only for doing those works in background. An XML (mainmenu.xml) file is created to generate menu options.

Google Maps API Key is added under value properties of meta-data Tag.

```
<meta-data
android:name="com.google.android.maps.v2.API_KEY"
android:value="obtained_api_key"/>
```

Some permission must be added to access Google Maps and for various purposes.

```
<uses-permission android:name="android.permission.ACCESS_NETWORK_S
TATE"/>
```

It allows the application to check the connection status in order to determine whether map data can be downloaded.

<uses-permission android:name="android.permission.INTERNET"/>

It allows the application to open network sockets and accessing internet.

<uses-permission
android:name="com.google.android.providers.gsf.permission.
READ_GSERVICES"/>

It allows the application to access Google web-based Services.

<uses-permission
android:name="android.permission.WRITE_EXTERNAL_ST ORAGE"/>

It allows the application to cache map tile data in the device's external storage area.

Next two permissions are necessary not only for opening map but also for showing current location of user's mobile.

<uses-permission
android:name="android.permission.ACCESS_COARSE_LOC ATION"/>

It allows the application to use Wi-Fi or mobile cell data (or both) to determine the device location.

<uses-permission android:name="android.permission.ACCESS_FINE_LOCATI
ON"/>

It allows the application to access GPS location within a very small area.

CHAPTER 2
LITERATURE SURVEY

On the Android platform, a Campus Assistant Application is planned and built for Florida Atlantic University's Boca Raton campus [7]. The application allows the user to choose the starting and destination locations, as well as the shortest driving and walking paths and parking lots. It also displays the user's current position using GPS. If the consumer deviates from the projected path, the application offers rerouting. They created a Map Editor app to help students edit and manage campus maps. Another research group inserted contextual details in the direction of destination from the user's current position to help the user with this application [5].

For the SRM University campus, an app called "Mobile Campus" was created and planned [1]. This campus tour guide application will run on Android phones with NFC (near field communication) capabilities. Visitors, teachers, and parents will all take advantage of it. The basic features of this app are included. When a user's NFC- enabled phone comes within range of another NFC enabled server/phone, he or she will see important landmarks and information about upcoming events such as seminars and sports.

For the 2013 Taiwan Lantern Festival [9], a comprehensive guiding and navigation service on a smartphone was developed. This app uses a custom map rather than a Google map to provide directions and navigation. This application displays the content of a 3D animation, which is a digitised version of a water painting. It graphically transforms a public map into a custom map. The customised map is saved in the KMZ format, which is a compressed version of the KML format. When the visitor is interested in the POI (Point of Interest), the guiding service can be triggered.

On Android, a map navigation system is created to solve the problem of the travelling salesman [6] using Google Maps and Google GeoCoder API. Using Google Maps,

a campus spatial information service system [10] is developed. The system is based on a Google Maps and MySQL database combination. The device includes interactive features such as an image, definition, connection, and a useful measurement tool. User can add his POI and knowledge about it to the database, and it also allows for user reviews.

The Wuhan University of Technology has introduced a useful model for the future digital campus [3]. This is a campus navigation device that is GIS-based. The system was built specifically for teachers and students in schools. It uses GIS technology to integrate the school's teaching tools, facilities, programmes, and other data, providing teachers and students with digital and intelligent information services.

The use of Google Maps for location-based navigation on the Android platform is becoming more common. The researchers have produced an application called Guide-My-Tour [8]. This chart shows and monitors a user's current position and direction of travel. It can be rotated in any direction and zoomed in and out. It combines a locality's conventional paper map picture with a satellite map image. Although user walking information was updated on a regular basis in response to the user's movements.

A research team has created a location-based nearest ATM search [2]. It's a GPS-based location monitoring service programme. A new position tracking algorithm was proposed and implemented in [4].

NFC has some drawbacks, such as the fact that it only operates at a distance of 4cm or less [1]. We developed our system so that users can get event information from any distance and from any venue. For storing and updating position data, all current systems use complex mechanisms, but we have introduced a very simple mechanism. Our framework offers a very comprehensive and customizable menu

choice for locating a position and determining the shortest route from the current

location. We've successfully integerated an event information system with basic map functionality.

CHAPTER 3

METHODOLOG

Y

AIM OF THE PROJECT

- This mobile app is based on discovering a solution to improve the situation providing convenience to the visitors.
- This proposal presents the work carried out in developing CAMPUS NAVIGATION mobile application to replace the traditional way of enquiring the same in person.
- Main goal of the project is to help the students, teachers and the visitors to find the way on their own.

SYSTEM REQUIREMENTS

SOFTWARE REQUIREMENTS

Vs CODE
Android Studio
React-Native
Google Map API

HARDWARE REQUIREMENTS:

GPS Enabled Mobile phone (ANDROID/ios)
Computer Processor : i7
Ram :16GB

OVERVIEW OF THE PLATFORM

REACT-NATIVE

React Native is a JavaScript framework for writing real, natively rendering mobile applications for iOS and Android. It's based on React, Facebook's JavaScript library for building user interfaces, but instead of targeting the browser, it targets mobile platforms. In other words: web developers can now write mobile applications that look and feel truly "native," all from the comfort of a JavaScript library that we already know and love. Plus, because most of the code you write can be shared between platforms, React Native makes it easy to simultaneously develop for both Android and iOS.

Similar to React for the Web, React Native applications are written using a mixture of JavaScript and XML-esque markup, known as JSX. Then, under the hood, the React Native "bridge" invokes the native rendering APIs in Objective-C (for iOS) or Java (for Android). Thus, your application will render using real mobile UI components, not webviews, and will look and feel like any other mobile application. React Native also exposes JavaScript interfaces for platform APIs, so your React Native apps can access platform features like the phone camera, or the user's location.

React Native currently supports both iOS and Android, and has the potential to expand to future platforms as well. In this book, we'll cover both iOS and Android. The vast majority of the code we write will be cross-platform. And yes: you can really use React Native to build production-ready mobile applications! Some anecdota: Facebook, Palantir, and TaskRabbit are already using it in production for user-facing applications.

Advantages of React Native

The fact that React Native actually renders using its host platform's standard

rendering APIs enables it to stand out from most existing methods of cross-platform

application development, like Cordova or Ionic. Existing methods of writing mobile applications using combinations of JavaScript, HTML, and CSS typically render using webviews. While this approach can work, it also comes with drawbacks, especially around performance. Additionally, they do not usually have access to the host platform's set of native UI elements. When these frameworks do try to mimic native UI elements, the results usually "feel" just a little off; reverse-engineering all the fine details of things like animations takes an enormous amount of effort, and they can quickly become out of date.

In contrast, React Native actually translates your markup to real, native UI elements, leveraging existing means of rendering views on whatever platform you are working with. Additionally, React works separately from the main UI thread, so your application can maintain high performance without sacrificing capability. The update cycle in React Native is the same as in React: when props or state change, React Native re-renders the views. The major difference between React Native and React in the browser is that React Native does this by leveraging the UI libraries of its host platform, rather than using HTML and CSS markup.

For developers accustomed to working on the Web with React, this means you can write mobile apps with the performance and look and feel of a native application, while using familiar tools. React Native also represents an improvement over normal mobile development in two other areas: the developer experience and cross- platform development potential.

Developer Experience

If you've ever developed for mobile before, you might be surprised by how easy React Native is to work with. The React Native team has baked strong developer tools and meaningful error messages into the framework, so working with robust tools is a natural part of your development experience.

For instance, because React Native is “just” JavaScript, you don’t need to rebuild your application in order to see your changes reflected; instead, you can hit Command+R to refresh your application just as you would any other web page. All of those minutes spent waiting for your application to build can really add up, and in contrast React Native’s quick iteration cycle feels like a godsend.

Additionally, React Native lets you take advantage of intelligent debugging tools and error reporting. If you are comfortable with Chrome or Safari’s developer tools (Figure 1-1), you will be happy to know that you can use them for mobile development, as well. Likewise, you can use whatever text editor you prefer for JavaScript editing: React Native does not force you to work in Xcode to develop for iOS, or Android Studio for Android development.

Besides the day-to-day improvements to your development experience, React Native also has the potential to positively impact your product release cycle. For instance, Apple permits JavaScript-based changes to an app’s behavior to be loaded over the air with no additional review cycle necessary.

All of these small perks add up to saving you and your fellow developers time and energy, allowing you to focus on the more interesting parts of your work and be more productive overall.

Code Reuse and Knowledge Sharing

Working with React Native can dramatically shrink the resources required to build mobile applications. Any developer who knows how to write React code can now target the Web, iOS, and Android, all with the same skillset. By removing the need to “silo” developers based on their target platform, React Native lets your team iterate more quickly, and share knowledge and resources more effectively.

Besides shared knowledge, much of your code can be shared, too. Not all the code you write will be cross-platform, and depending on what functionality you need on a specific platform, you may occasionally need to dip into Objective-C or Java. (Happily, this isn't too bad, and we'll cover how so-called native modules work in Chapter 7.) But reusing code across platforms is surprisingly easy with React Native. For example, the Facebook Ads Manager application for Android shares 87% of its codebase with the iOS version, as noted in the React Europe 2015 keynote. The final application we'll look at in this book, a flashcard app, has total code reuse between Android and iOS. It's hard to beat that!

Risks and Drawbacks

As with anything, using React Native is not without its downsides, and whether or not React Native is a good fit for your team really depends on your individual situation.

The largest risk is probably React Native's maturity, as the project is still relatively young. iOS support was released in March 2015, and Android support was released in September 2015. The documentation certainly has room for improvement, and continues to evolve. Some features on iOS and Android still aren't supported, and the community is still discovering best practices. The good news is that in the vast majority of cases, you can implement support for missing APIs yourself, which we'll cover in Chapter 7.

Because React Native introduces another layer to your project, it can also make debugging hairier, especially at the intersection of React and the host platform. We'll cover debugging for React Native in more depth in Chapter 8, and try to address some of the most common issues.

React Native is still young, and the usual caveats that go along with working with new technologies apply here. Still, on the whole, I think you'll see that the benefits outweigh the risks.

Summary

React Native is an exciting framework that enables web developers to create robust mobile applications using their existing JavaScript knowledge. It offers faster mobile development, and more efficient code sharing across iOS, Android, and the Web, without sacrificing the end user's experience or application quality. The trade-off is that it's new, and still a work in progress. If your team can handle the uncertainty that comes with working with a new technology, and wants to develop mobile applications for more than just one platform, you should be looking at React Native.

Global Positioning System (GPS)

The Global Positioning System (GPS), originally Navstar GPS, is a satellite-based radionavigation system owned by the United States government and operated by the United States Space Force. It is one of the global navigation satellite systems (GNSS) that provides geolocation and time information to a GPS receiver anywhere on or near the Earth where there is an unobstructed line of sight to four or more GPS satellites. Obstacles such as mountains and buildings block the relatively weak GPS signals. GPS position is not perfect. Some errors you have some control over. As a GPS user, you have some control over the first type of error - you can wait for the satellites to move into better geometry or plan your data collection for a good time of day. You cannot control multi-path error, but you do need to recognize when it may occur and pay close attention to the measurements that you are taking. Orbital and Satellite Clock errors as well as atmospheric delays are invisible to you, but you can avoid them by using differential correction. There is nothing that you can do about your receiver clock errors. GPS receivers will generally report an estimate of the accuracy of the position being reported by the unit at the time.

The GPS does not require the user to transmit any data, and it operates independently of any telephonic or internet reception, though these technologies can enhance the usefulness of the GPS positioning information. The GPS provides critical positioning capabilities to military, civil, and commercial users around the world. The United States government created the system, maintains it, and makes it freely accessible to anyone with a GPS receiver.[better source needed]

The GPS project was started by the U.S. Department of Defense in 1973, with the first prototype spacecraft launched in 1978 and the full constellation of 24 satellites

operational in 1993. Originally limited to use by the United States military, civilian use was allowed from the 1980s following an executive order from President Ronald Reagan. Advances in technology and new demands on the existing system have

now led to efforts to modernize the GPS and implement the next generation of GPS Block IIIA satellites and Next Generation Operational Control System (OCX). Announcements from Vice President Al Gore and the Clinton Administration in 1998 initiated these changes, which were authorized by the U.S. Congress in 2000.

During the 1990s, GPS quality was degraded by the United States government in a program called "Selective Availability"; this was discontinued on May 1, 2000 by a law signed by President Bill Clinton.

The GPS service is provided by the United States government, which can selectively deny access to the system, as happened to the Indian military in 1999 during the Kargil War, or degrade the service at any time. As a result, several countries have developed or are in the process of setting up other global or regional satellite navigation systems. The Russian Global Navigation Satellite System (GLONASS) was developed contemporaneously with GPS, but suffered from incomplete coverage of the globe until the mid-2000s. GLONASS can be added to GPS devices, making more satellites available and enabling positions to be fixed more quickly and accurately, to within two meters (6.6 ft). China's BeiDou Navigation Satellite System began global services in 2018, and finished its full deployment in 2020. There are also the European Union Galileo positioning system, and India's NavIC. Japan's Quasi-Zenith Satellite System (QZSS) is a GPS satellite-based augmentation system to enhance GPS's accuracy in Asia-Oceania, with satellite navigation independent of GPS scheduled for 2023.

When selective availability was lifted in 2000, GPS had about a five-meter (16 ft) accuracy. The latest stage of accuracy enhancement uses the L5 band and is now

fully deployed. GPS receivers released in 2018 that use the L5 band can have much higher accuracy, pinpointing to within 30 centimeters (11.8 in).

Android

There is a lot of advancement in the OS (Operating Systems) since the last few decades.

When we look at the early stages of the cell phones, the black and white phones were came into focus but with the passage of time, cell phones were pushed onto the next level and now smart phones are particularly in focus. Mobile OS has come into horizon.

The operating systems for smart phones & tablets, there was a need of change, so after the year 2000 the OS like Android and Blackberry were urbanized. With the passage of time, among all the greatest and widely used mobile operating systems there was android. Android Inc. was founded in Palo Alto of California, U.S. by Andy Rubin, Rich miner, and Chris White in early 2000. Then later on in 2005 Android was picked up by Google. With the advancement in Android OS, numerous versions were introduced. After that there have been a number of updates in the original version of Android [1]. Android Operating System gives the flexibility for both (users & developers).

Challenges for Android

- Hard to Integrate for Vendors

When talking about the iOS, the developers have to deal with fewer complications in the developing process because of the fewer versions. The same is not the case with Android applications. There are different brands in the market offering different screen sizes and using various kinds of processors. So this makes a great deal of opportunity for the android developers to innovate something new which can be nearly compared to as a difficult task.

- Performance Consideration

Every device has its own way of handling itself. Different brands, offering different devices with various screen sizes and using various kinds of processors results in performance issues.

- Too Much Google Dependent

When talking about android, we automatically came to know that it is a lot in the handled by Google. Google makes Android a bit more terms independent which makes it much better.

Google Map's API

Google Map's API is a robust tool that can be used to create a custom map, a searchable map, check-in functions, display live data synching with location, plan routes, or create a mashup just to name a few.

Google Map's APIs are very powerful, but what is even more powerful is the mashup of more than one API. This is actually very cool, and probably the best discovery I have found for what initially began as a mini-project. Now, I am thinking of utilizing Google Map's API for an app that no one else has thought of. For example, one cool mashup idea a programmer came up with blends Google Maps and Craigslist computer gig listings.

Once all that is set up you can enable the Google Maps JavaScript API within the Google developers console. It should give you back an API key to use but if not you can get there through clicking the navigation menu and going into "**API & Services**" and then "**Credentials**". Now you can click on "**Create credentials**" to get your API key. It may ask you to restrict your key so nobody else can use it and it's probably a good idea to do so.

Set up file

First you will need some boilerplate HTML so create a directory with an index.html file inside. I like to use Emmet, which is available in a lot of different text editors. With emmet I can just type '**!**' and then hit the '**tab**' button to create some basic HTML. If you don't want to do all that then you can just use this boilerplate HTML:

```
<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">
```

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<meta http-equiv="X-UA-Compatible" content="ie=edge">

<title>My Google Map</title>

</head>

<body>

</body>

</html>

Initialize Google Maps

Now that we have our HTML we can start adding our <script> tags to communicate with the Google Maps API. This script will give us access to all the goodies. At the end of your body tag you will want to add this script tag:

<script
src="https://maps.googleapis.com/maps/api/js?key=YOUR_API_KEY&callback=initMap"

async defer></script>

You will need to replace "**YOUR_API_KEY"** with the key you created inside your Google Developer console. If you take a close look at this script tag, you can see **"callback=initMap"** parameter at the end of the link. This will call the function **initMap()** so we will need to set up some JavaScript to initialize up our map. Technically you don't need this callback, there are other ways to do it, but we will use it for this exercise.

Next lets put in some script tags above the ones we just inserted and start our map initialization function. I am going to name the function the same thing that our API script is calling, initMap(). Inside this function, we are going to create a new map. In order to do so, we need to call and pass in a few things:

```
new google.maps.Map(ui_element_to_attach_map_to,
```

We can store these arguments into variables and then pass them into the function to keep our code a little cleaner. Before we can do that, we need to create an element within our <body> to attach our map to. Let's throw in a with an id set equal to "map" above our <script> tags:

Now we have our element we can grab and attach our map to. Second thing we need is the map options object that sets the zoom level of the map and where it will center the map. All together our body should look like this:

<body> <script>

function initMap() {const mapDiv = document.getElementById('map')const options = {

zoom: 12,

center: { lat: 29.7604, lng: -95.3698 }

}const map = new google.maps.Map(mapDiv, options)

}

</script><script src="https://maps.googleapis.com/maps/api/js?key=YOUR_API_KEY&callback=initMap" async defer></script></body>

We aren't done just yet, our map will not appear yet though because we need to add some styling to the element containing our map. We can simply put in a <style> tag at the end of our <head> tag:

<style>

#map {

height:

}

</style>

If you have followed these steps correctly you should be able to open up index.html and see the map we just created. You can play around with the “options” variable we created to pass into:

new google.maps.Map(mapDiv, options)

By decreasing the zoom, you will zoom further out of the map and increasing the value will zoom in. To change the latitude and longitude, you will need to search for the coordinates online to replace in the code and center the map wherever you’d like.

Creating and Customizing a Marker

Next we will run through creating a marker. To create a new marker we will need to simply call and pass in these values:

new google.maps.Marker(marker_attributes)

Our **“marker_attributes”** will be an object with the necessary attributes for creating a new marker. We need to give it the marker’s position in the form of an object with latitude and longitude keys. We also need to pass in what map we want to add the marker to. In this case we saved our new map to a variable called “map”. Our code to create a new marker might look like this:

const marker = new

google.maps.Marker({ position: { lat:

29.7604, lng: -95.3698 } map: map

CHAPTER 4
MODULE DESCRIPTION

SYSTEM ARCHITECTURE:

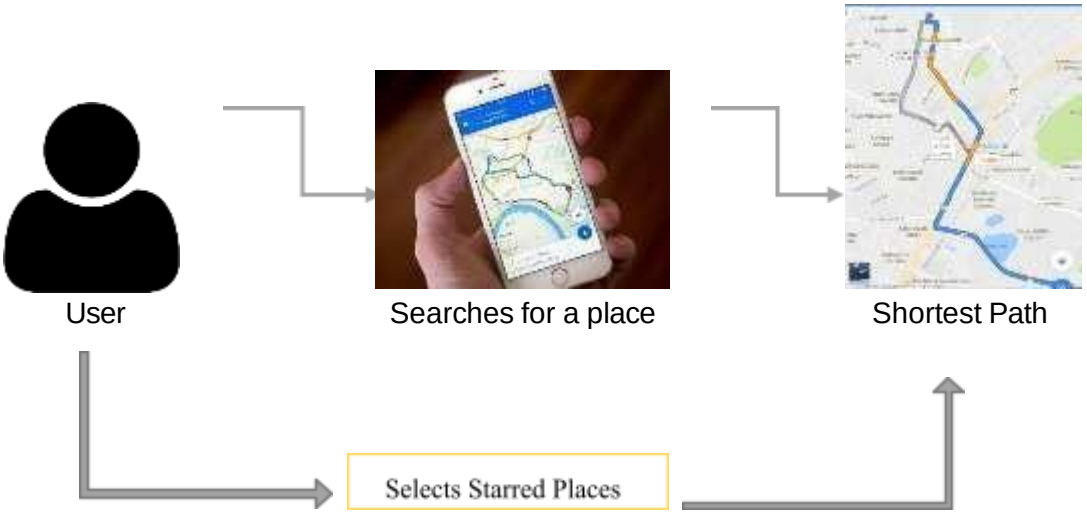

FIG 4.1 System Architecture

MODULE DESCRIPTION

At first, the user's app requests Google API from a GPS satellite. The map is then loaded into the user's app. The current location of the user is monitored and shown on a map using a GPS activated system on the user's mobile phone. A cellular data network is used to send a User-app message.

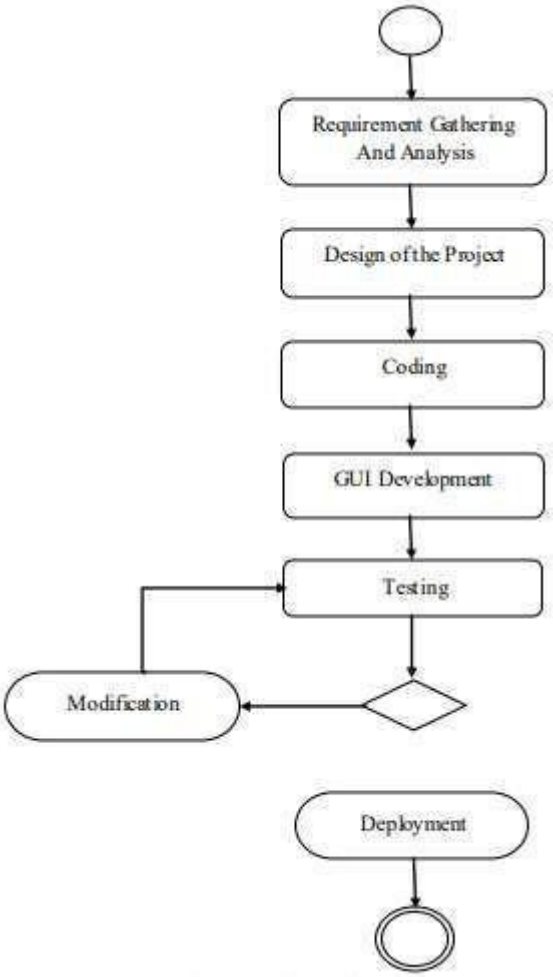

Fig 4.2 Project Plan

A mobile application that runs on Android platform will be created to solve the problem. The mobile application comprises the following features:

• Function to select initial position (Point A) and final destination (Point B)

Scenario: If a student wants to go point A from point B It displays shortest path between Point A and Point B

Scenario: The application searches for all possible paths, the shortest path among all possible paths that direct to B from A. It displays the shortest path on map.

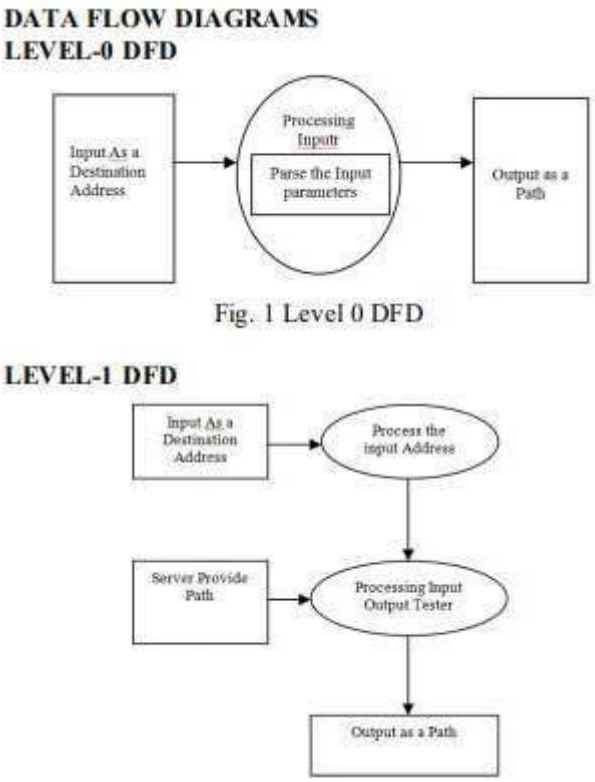

Fig 4.3 Data Flow Diagram

All the places in the campus will be marked on the Google Map. This will allow us to find the location easily. To provide location information to the mobile user, data from the Global Positioning System (GPS) is used. When map-based services are needed, "Google Maps" is used. To access the tour guide details, the user interacts with the mobile phone. The user can access the stored information by selecting the interesting places indicated on the map.

The application will provide this information to the user when they are on the premises or as off-site information. The application is a voice-based provider of information, a location ID and a map-based path selection feature to select the best path for the destination. Also the important places will be starred particularly so that one can locate instantly. The user can search the place or also anyone can directly

CHAPTER 5

RESULT AND DISCUSSION

In the past few years, checking location with Google maps has become a new trend when people don't know where they are. Google maps include several functionalities, such as viewing any spot, alternate track from somewhere to another and estimate time to reach the place. But for college campuses it's not well known or really helpful. It is very easy for newly admitted students and tourists to enter any position within the school campus, such as entrance gates, offices, canteen, library, playground, parking lot, etc.

Login and Authentication in Firebase

To authenticate accounts, simply obtain the user's authentication credentials and pass them to the Firebase Authentication SDK, which can be an email address and password, a mobile phone number, or any token from identity providers such as Facebook, Google, Twitter, GitHub, and others. To sign a user into your app, you first get authentication credentials from the user. These credentials can be the user's email address and password, or an OAuth token from a federated identity provider. Then, you pass these credentials to the Firebase Authentication SDK. Our backend services will then verify those credentials and return a response to the client.

After a successful sign in, you can access the user's basic profile information, and you can control the user's access to data stored in other Firebase products. You can also use the provided authentication token to verify the identity of users in your own backend services.

After that we will add below code line to type of app build.gradle file and Sync Now

```
dependencies {
    implementation fileTree(dir: 'libs', include: ['*.jar'])
    implementation 'com.android.support:appcompat-v7:26.1.0'
    implementation 'com.android.support:design:26.1.0'
    implementation 'com.google.firebase:firebase-auth:11.8.0'
    implementation 'com.google.android.gms:play-services-auth:11.8.0'
}
apply plugin: 'com.google.gms.google-services'
```

After passing the credentials, Firebase will validate them, and you will receive a response indicating whether or not the authentication was successful.

Fig. 6.1 Login Screen

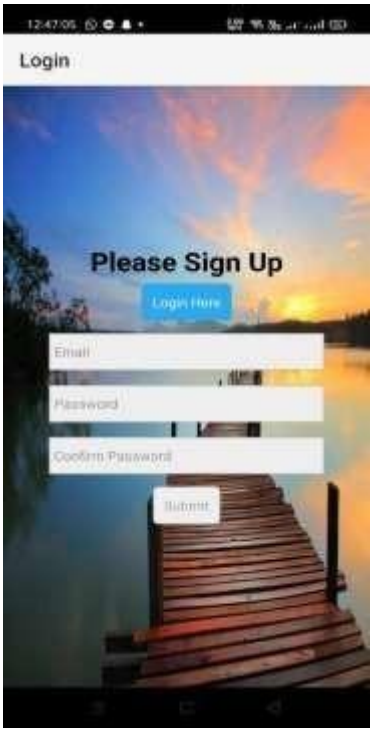

Fig. 6.2 Sign-up Screen

There is no mechanism in place to easily notify all current students, teachers, and staff of any activities that may begin minutes, hours, or days later or earlier. As a consequence, there's a good chance you'll miss important events or facts, such as accidents or strikes. Departments, libraries, and canteens can relocate. It is excruciatingly traumatic for all current students, teachers, and staff, as well as newcomers.

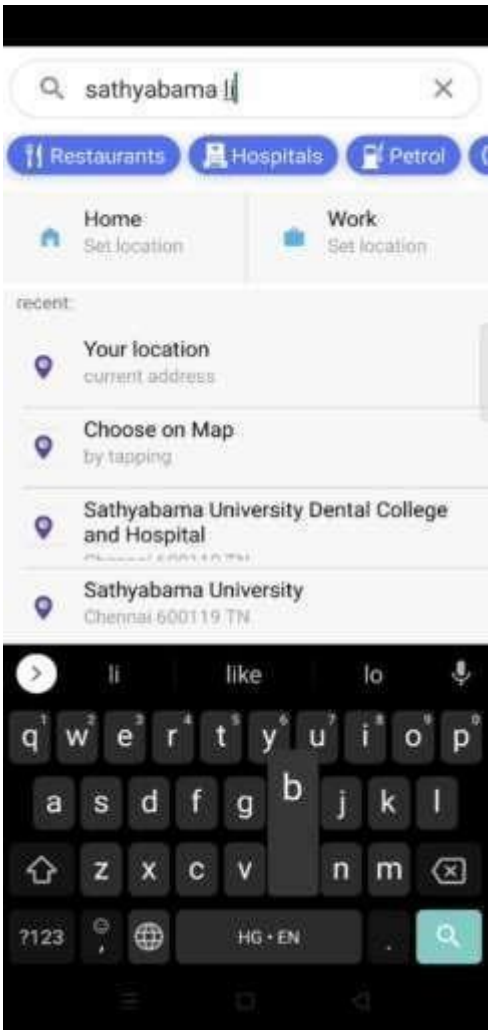

Fig. 6.3 Searching Place

Fig. 6.4 Campus Boundary

This work successfully developed, introduced, and tested. This application provides users with a handy route guide from their current location to their destination, as well as event notifications with their proper location. It also helps you to customise maps by adding new locations and modifying existing ones.

Fig. 6.5 Route Navigation

Fig. 6.6 Route Guide

Advantages of App

1) The user will locate the correct location and read the appropriate description.
2) The user can get shortest path to the destination.

Basic requirement for Campus Navigation

1) This App needs a network connection.
2) User must have Mobile with inbuilt GPS.

CHAPTER 6

CONCLUSIO

N

When people are uncertain of their location, location searching has become a new trend in recent years thanks to Google maps. Google Maps has a lot of features, such as showing any location, showing alternative routes from any location to another location, and estimating travel time .But it is not well developed or so much helpful for college campuses. We have a lot of navigation issues when we go to another college for a case. This application's shortest path function will save the user time. As a result, the application's strength is its simple navigation function, which can locate paths on campus to user-defined locations. This application provides users with a route guide from their current location to their desired location.

Though Google map is very helpful to find a location and alternative route to the desired location from user own location. But it is not so much helpful for our university campus because there are many departments, schools, library, canteen, playground, gates, hostel, and parking lots etc. which are not properly shown on Google map. University campus is growing faster and as a result changes are happening inside campus. Dependability on Android Application is growing faster with the advancement of smart phone technology. There are lots of Google map based apps on market. There are some technologies to display custom map. This can be achieved through Storing location data on XML file and place this XML on a server Displaying map taking location information data from MySQL server Spread event information with the help of NFC technology. The aim of this research work is to develop a user friendly, Google map based application for university campus navigation and updates of event information on android mobile in some different way. This application will show user’s current location in the campus or outside

campus with a marker. Users can get the desired places with a marker and also shortest path from current location. Users can also get updates about event information. Users can view any new building, canteen, playground, etc.

The following topics can be added to this work:

Audio Controls: Visually impaired people will benefit if speech enabled control is applied to this architecture. Having a frequent audio warning when their position changes will help them navigate to the correct location. This will help them relax while they're looking for a location or trying to get somewhere.

Multimedia Based Advertisement: Any event may include a rich multimedia-based advertising such as a poster, audio promotion, and video promotion, as well as a map with the event's exact location. This is very successful and removes the need for physical posters.

Since we used GPS-based position tracking, it will not work in an enclosed room. Wi-Fi-based position monitoring, on the other hand, works well inside buildings and over short distances. Inside houses, Wi-Fi-based location monitoring can be implemented to make the device more useful.

More Sophisticated and Customized Map: It ishelpful for any places, mainly for faster changing rural places.

REFERENCES

[1] "Design and Development of Mobile Campus, an Android based Mobile Application for University Campus Tour Guide,"Sagnik Bhattacharya and M. B. Panbu, (IJITEE) Issue 3, February 2013.

[2] "Mobile Banking With Location Tracking Of Nearest ATM Center Using GPS", Gugapriya A, Vaitheki J and Kaviyarasi S, ISSN: 2320 –5547, April - May 2013.

[3] "Development of a Campus Information Navigation System Based on GIS," Jiejun HUANG, Yunjun Zhan, Wei CUI, Yanbin YUAN and Peipei QI, IEEE, Vol. 5, 2010.

[4] "A New Approach for Location based Tracking", Shaveta Bhatia and Saba Hilal, (IJCSI), Vol. 10, Issue 3, No 1, May 2013.

[5] "A Pattern for Context-Aware Navigation", Mihaela Cardei, Brandon Jones and Daniel Raviv, 20th (PLoP'13), Allerton Park, Monticello, Illinois USA, October 23-26, 2013.

[6] "An Android Application for Google Map Navigation System Implementing Travelling Salesman Problem," Anupriya and Mansi Saxena, ISSN: 2249- 2593, Vol. 3, Issue 4, 2013.

[7] "Campus Assistant Application on an Android Platform", Mihaela Cardei, Iana Zankina, Ionut Cardei, and Daniel Raviv, Proceedings of IEEE SoutheastCon, pp 1-6, Jacksonville, Florida, April 04-07, 2013.

[8] "A Context-Aware System for Navigation and Information Dissemination on Android Devices," P.Silapachote, Ananta S, Rasita S, W.Kaewpijit, and N Waragulsiriwan, 2013 IEEE, ISSN: 2159 – 3442, pp.1 – 4, October 22-25, 2013.

Lin, 2013 (ICSEC), ISBN: 978-1-4673-5322-9, pp. 97-102, September 04- 06, 2013.

[10] “Design and Implementation of Campus Spatial Information Service Based on Google Maps”, Yang Yang, Jianhua Xu, Jianghua Zheng and Shouyi Lin, 2009 (MASS, 2009), ISBN: 978-1-4244-4638-4, pp. 1-4, September 20-22, 2009.

APPENDIX
A. CODE SOURCE

I. Authentication Screen

```
import React, { Component } from 'react';
import {
    View,
    Text,
    Button,
    TextInput,
    StyleSheet,
    ImageBackground,
    Dimensions,
    KeyboardAvoidingView,
    Keyboard,
    TouchableWithoutFeedback,
    ActivityIndicator
} from 'react-native';
import { connect } from 'react-redux';

import DefaultInput from '../../components/UI/DefaultInput/DefaultInput';
import HeadingText from '../../components/UI/HeadingText/HeadingText';
import MainText from '../../components/UI/MainText/MainText';
import BackgroundedButton from '../../components/UI/BackgrounedButton/Butto
n';

import bgImage from '../../assets/background.jpg';
import validate from '../../utility/validation';
import { tryAuth, authAutoSignIn } from '../../store/actions/index';

class AuthScreen extends Component {
    state = {
        styles: {
            pwContainerDirection: 'column',
            pwContainerJustifyContent: 'flex-start',
            pwWrapperWidth: '100%'
        },
        authMode: 'login',
```

```
            value: '',
            valid: false,
            validationRules: {
                isEmail: true
            },
            touched: false
        },
        password: {
            value: '',
            valid: false,
            validationRules: {
                minLength: 6
            },
            touched: false
        },
        confPassword: {
            value: '',
            valid: false,
            validationRules: {
                equalTo: 'password'
            },
            touched: false
        }
    }
};

constructor(props) {
     super(props);
    Dimensions.addEventListener('change', this.updateStyles);
}

componentWillUnmount() {
    Dimensions.removeEventListener('change', this.updateStyles);
}

componentDidMount() {
    this.props.onAutoSignIn();
}
```

```
updateStyles = dims =>
    { this.setState({
        styles: {
```

```
    pwContainerDirection:
      Dimensions.get('window').h
    eight > 500 ? 'column' : 'row',
    pwContainerJustifyContent:
      Dimensions.get('window').height > 500
        ? 'flex-start'
        :
    '
    s
    p
    a
    c
    e
    -
    b
    e
    t
    w
    e
    e
    n
    '
    ,
    p
    w
    W
    r
    a
    p
    p
    e
    r
    W
    i
    d
    t
    h
    :
      Dimensions.get('window').h
      eight > 500 ||
        }
      });
};
```

```
this.state.authMode ===                                : '45%'
'login'
    ? '100%'

        authHandler = () => {
            const authData = {
                email: this.state.controls.email.value,
                password: this.state.controls.password.value
            };
            this.props.onTryAuth(authData, this.state.authMode);
        };

        updateInput = (key, value) => {
            let conValue = {};
            if (this.state.controls[key].validationRules.equalTo) {
                const eqControl = this.state.controls[key].validationRules.equalTo;
                const equalTo = this.state.controls[eqControl].value;
                conValue = {
                    ...conValue,
                    equalTo
                };
            }
            if (key === 'password') {
                conValue = { ...conValue, equalTo: value };
            }
            this.setState(prevState => {
                return {
                    controls: {
                        ...prevState.controls,
                        confPassword: {
                            ...prevState.controls.confPassword,
```

```
                valid:
                    key === 'password'
                        ? validate(
                            prevState.controls.confPassword.value,
                            prevState.controls.confPassword
                                .validationRules,
                            conValue
                                        )
                                        : prevState.controls.confPassword.valid
            },
            [key]: {

                ...prevState.controls[key],
                value,
                valid: validate(
                    value,
                    prevState.controls[key].validationRules,
                    conValue
                                        ),
                                        touched: true
            }
        }
    };
});
};


switchAuthModeHandler = () => {
    this.setState(prevState => {
        return {
            authMode: prevState.authMode === 'login' ? 'signup' : 'login'
        };
    });
    this.updateStyles();
};

render() {
    let headingText = null;
    let confPasswordControl = null;
```

```
if (Dimensions.get('window').height > 500) {
    headingText = (
        <MainText>
            <HeadingText>
```

```
            Please{' '}
            {this.state.authMode !== 'login' ? 'Sign Up' : 'Login'}
          </HeadingText>
        </MainText>
    );
}

if (this.state.authMode === 'signup') {
    confPasswordControl = (
        <View
            style={{
                width: this.state.styles.pwWrapperWidth
            }}>
            <DefaultInput
                placeholder="Confirm
                Password" style={styles.input}
                value={this.state.controls.confPassword.value}
                onChangeText={val =>
                    this.updateInput('confPassword', val)
                }
                valid={this.state.controls.confPassword.valid}
                touched={this.state.controls.confPassword.touched}
                secureTextEntry
            />
        </View>
    );
}

return (
    <ImageBackground
        source={bgImage}
        style={styles.backgroundImage}
        imageStyle={{ resizeMode: 'stretch' }}>
        <KeyboardAvoidingView
            style={styles.container}
            behavior="padding">
            {headingText}
            <BackgroundedButton
                bgColor="#29aaf4"
```

```
color="white"
brColor="transparent"
onPress={this.switchAuthModeHandler}>
```

```
    {this.state.authMode === 'login' ? 'Sign Up' : 'Login'}{' '}
    Here
</BackgroundedButton>
<TouchableWithoutFeedback onPress={Keyboard.dismiss}>
    <View style={styles.inputContainer}>
        <DefaultInput
            placeholder="Email"
            style={styles.input}
            value={this.state.controls.email.value}
            onChangeText={val =>
                this.updateInput('email', val)
            }
            valid={this.state.controls.email.valid}
            touched={this.state.controls.email.touched}
            autoCapitalize="none"
            autoCorrect={false}
            keyboardType="email-address"
        />
        <View
            style={{
                flexDirection: this.state.styles
                    .pwContainerDirection,
                justifyContent: this.state.styles
                    .pwContainerJustifyContent
            }}>
            <View
                style={{
                    width: this.state.styles.pwWrapperWidth
                }}>
                <DefaultInput
                    placeholder="Password"
                    style={styles.input}
                    value={
                        this.state.controls.password.value
                    }
                    onChangeText={val =>
                        this.updateInput('password', val)
                    }
                    valid={
```

```
    this.state.controls.password.valid
}
touched={
```

```
                                this.state.controls.password.touched
                            }
                            secureTextEntry
                        />
                    </View>
                    {confPasswordControl}
                </View>
            </View>
        </TouchableWithoutFeedback>
        {this.props.isLoading ? (
            <ActivityIndicator size="large" color="#29aaf4" />
        ) : (
            <BackgroundedButton
                bgColor="#29aaf4"
                color="white"
                brColor="transparent"
                onPress={this.authHandler}
                disabled={
                    (!this.state.controls.confPassword.valid &&
                        this.state.authMode === 'signup') ||
                    !this.state.controls.password.valid ||
                    !this.state.controls.email.valid
                }>
                Submit
            </BackgroundedButton>
        )}
    </KeyboardAvoidingView>
</ImageBackground>
    );
  }
}

const styles = StyleSheet.create({
  container: {
    flex: 1,
    justifyContent: 'center',
    alignItems: 'center'
  },
  backgroundImage: {
```

```
    width: '100%',
    flex: 1
},
```

```
    inputContainer: {
      width: '75%'
    },
    input: {
      backgroundColor: '#eee',
      borderColor: '#bbb'
    }
  });

  const mapStateToProps = state => {
    return {
      isLoading: state.ui.isLoading
    };
  };

  const mapDispatchToProps = dispatch => {
    return {
      onTryAuth: (authData, authMode) =>
        dispatch(tryAuth(authData, authMode)),
      onAutoSignIn: () => dispatch(authAutoSignIn())
    };
  };

  export default connect(
    mapStateToProps,
    mapDispatchToProps
  )(AuthScreen);
```

II. FINDING PLACES

```
import React, { Component } from 'react';

import {

  View,

  Text,

  TouchableOpacity,

  TouchableNativeFeedback,
```

StyleSheet,

```
    Platform,
    Animated
} from 'react-native';
import { connect } from 'react-redux';

import PlaceList from '../../components/PlaceList/PlaceList';
import { getPlaces } from '../../store/actions/index';

class FindPlaceScreen extends Component {
    static navigatorStyle = {
        navBarButtonColor: '#29aaf4'
    };

    state = {
        placesLoaded: false,
        btnAnimation: new Animated.Value(1)
    };

    constructor(props) {
        super(props);
        this.props.navigator.setOnNavigatorEvent(this.onNavigatorEvent);
    }

    onNavigatorEvent = event => {
        if (event.type === 'ScreenChangedEvent') {
            if (event.id === 'willAppear') {
```

```
this.props.onLoadPlaces();
```

```
        }
    }
    if (event.type === 'NavBarButtonPress') {
        if (event.id === 'sideDrawerToggle') {
            this.props.navigator.toggleDrawer({
                side: 'left'
            });
        }
    }
};

itemSelectedHandler = key => {
    const selPlace = this.props.places.find(place => {
        return place.key === key;
    });
    this.props.navigator.push({
        screen: 'place-finder.PlaceDetailScreen',
        title: selPlace.name,
        passProps: {
            selectedPlace: selPlace
        }
    });
};

placesSearchHandler = () => {
    Animated.timing(this.state.btnAnimation, {
```

toValue: 0,

```
        duration: 500,
        useNativeDriver: true
    }).start();
    this.setState({ placesLoaded: true });
};

render() {
    let content =
        Platform.OS === 'android' ? (
            <Animated.View
                style={{
                    opacity: this.state.btnAnimation,
                    transform: [
                        {
                            scale: this.state.btnAnimation.interpolate({
                                inputRange: [0, 1],
                                outputRange: [12, 1]
                            })
                        }
                    ]
                }}>
                <TouchableNativeFeedback onPress={this.placesSearchHandler}>
                    <View style={styles.searchButton}>
                        <Text style={styles.searchText}>Find Places</Text>
                    </View>
                </TouchableNativeFeedback>
```

```
</Animated.View>
```

```
    ) : (
      <Animated.View
        style={{
          opacity: this.state.btnAnimation,
          transform: [
            {
              scale: this.state.btnAnimation.interpolate({
                inputRange: [0, 1],
                outputRange: [12, 1]
              })
            }
          ]
        }}>
        <TouchableOpacity onPress={this.placesSearchHandler}>
          <View style={styles.searchButton}>
            <Text style={styles.searchText}>Find Places</Text>
          </View>
        </TouchableOpacity>
      </Animated.View>
    );

if (this.state.placesLoaded) {
    content = (
      <PlaceList
        places={this.props.places}
        onItemSelected={this.itemSelectedHandler}
```

/>

```
            );
        }

        return (
            <View
                style={this.state.placesLoaded ? null : styles.buttonContainer}>
                {content}
            </View>
        );
    }
}

const styles = StyleSheet.create({
    buttonContainer: {
        flex: 1,
        justifyContent: 'center',
        alignItems: 'center'
    },
    searchButton: {
        borderColor: '#29aaf4',
        borderWidth: 2,
        borderRadius: 20,
        padding: 15
    },
    searchText: {
        color: '#29aaf4',
```

fontWeight: 'bold',

```
        fontSize: 20
    }
});

const mapStateToProps = state => {
    return {
        places: state.places.places
    };
};

const mapDispatchToProps = dispatch => {
    return {
        onLoadPlaces: () => dispatch(getPlaces())
    };
};

export default connect(
    mapStateToProps,
    mapDispatchToProps
)(FindPlaceScreen);
```

III. LOCATION PICKER

```
import React, { Component } from 'react';
import { View, StyleSheet, Text, Dimensions } from 'react-native';
import BackgroundedButton from '../UI/BackgrounedButton/Button';
import MapView from 'react-native-maps';
```

```
class PickLocation extends Component {
    componentWillMount() {
        this.reset();
    }

    reset = () => {
        this.setState({
            focusedLoc: {
                latitude:
                -6.89148,
                longitude: 107.6107,
                latitudeDelta: 0.00522,
                longitudeDelta:
                    (Dimensions.get('window').width /
                        Dimensions.get('window').height)
                        *
                    0.00522
                                                    },
                                                    chosedLoc: false

                                                });
    };


    pickLocationHandler = event => {
        const coordinates = event.nativeEvent.coordinate;
        this.map.animateToRegion({
```

```
    ...this.state.focusedLoc,
    latitude: coordinates.latitude,
    longitude: coordinates.longitude
});
```

```
    this.setState(prevState => {
        return {
            focusedLoc: {
                ...prevState.focusedLoc,
                latitude: coordinates.latitude,
                longitude: coordinates.longitude
                                                    },
                                                    chosedLoc: true
                                                };
                                            });

    this.props.onLocationPick({
        latitude: coordinates.latitude,
        longitude: coordinates.longitude
    });
};

getLocationHandler = () => {
    const self = this;
    navigator.geolocation.getCurrentPosition(
        pos => {
            const coordsEvent = {
                nativeEvent: {
                    coordinate: {
                        latitude: pos.coords.latitude,
                        longitude: pos.coords.longitude
```

```
    }
}
```

```
            };
            self.pickLocationHandler(coordsEvent);
        },
        err => {
            alert(
                'Fetching the position failed, please pick manually on the map'
            );
        }
    );
};

render() {
    let marker = null;
    if (this.state.chosedLoc) {
        marker = <MapView.Marker coordinate={this.state.focusedLoc} />;
    }
    return (
        <View style={styles.container}>
            <MapView
                initialRegion={this.state.focusedLoc}
                region={
                    !this.state.chosedLoc ? this.state.focusedLoc : null
                }
                style={styles.map}
                onPress={this.pickLocationHandler}
                ref={ref => (this.map = ref)}>
```

{marker}

```
            </MapView>
            <View style={styles.button}>
              <BackgroundedButton
                color="#29aaf4"
                bgColor="white"
                brColor="#29aaf4"
                onPress={this.getLocationHandler}>
                Locate Me
              </BackgroundedButton>
            </View>
          </View>
        );
    }
}

const styles = StyleSheet.create({
    container: {
        width: '100%',
        alignItems: 'center'
    },
    map: {
        width: '100%',
        height: 250
    },
    button: {
        margin: 8
```

```
}
```

```
});

export default PickLocation;
```

IV. PACKAGE.JSON

```
{
    "name": "place-finder",
    "version": "0.0.1",
    "private": true,
    "scripts": {
        "start": "node node_modules/react-native/local-cli/cli.js start",
        "test": "jest"
    },
    "dependencies": {
        "react": "16.0.0-beta.5",
        "react-native": "^0.63.4",
        "react-native-animatable":
        "^1.2.4", "react-native-dialogs":
        "^1.0.2",
        "react-native-image-picker": "^0.27.2",
        "react-native-maps": "^0.22.1",
        "react-native-navigation": "^1.1.236",
        "react-native-vector-icons":  "^4.4.2",
        "react-redux": "^5.0.6",
        "redux": "^3.7.2",
        "redux-thunk": "^2.3.0"
```

},
"devDependencies": {

```
        "babel-jest": "21.2.0",
        "babel-preset-react-native": "4.0.0",
        "jest": "21.2.1",
        "react-test-renderer": "16.0.0-beta.5"
    },
    "jest": {
        "preset": "react-native"
    }
}
```

Mobile App for Campus Navigation

Ujjwal Tiwari
U.G Scholar
Department of Computer Science Sathyabama Institute of Science and Technology
Chennai, India
ujjwalut18@gmail.com

Arya Verma
U.G Scholar *Department of Computer Science Sathyabma Institute of Science and Technology*
Chennai, India
arya.verma79@gmail.com

S. Revathy
Associate Professor
School of Computing Sathyabama Institute of Science and Technology
Chennai, India
revathy.it@sathyabama.ac.in

Abstract—Navigation is a method that focuses on the method of tracking and regulating the movement of people, vehicles, and craft from one location to another, such as land navigation, marine navigation, and aeronautic navigation. The campus navigator is an Android smartphone application that is used to navigate routes inside any campus premises, such as a mall, college, or hospital. Mobile phones are now becoming more than just communication devices. Smartphones, in particular, are products that make our jobs and daily lives easier. The use of mobile apps has exploded in recent years, owing to advancements in technology and the popularity of these devices. New techniques such as GPS, cameras, compass, and accelerometer, which can be used to measure the user's position, can also be used to create location-based applications with augmented reality views. There are many commercial navigation apps that provide users with directions from one location to another, such as Google Maps, Yahoo Maps, and Mapquest. These apps, on the other hand, are limited to searching for existing roads and are unable to provide routes as precise as an on-campus path. The primary aim of this paper is to present the findings of research and development into an intelligent mobile phone system that can help users find the shortest path between two points of interest on campus.The aim of the research is to find the best navigation solution with the shortest path calculation as the primary goal. An Android-based mobile application that allows users to quickly and easily locate a spot. The proposed methods and mobile application, according to the preliminary evaluation, help users quickly find their focus. The key feature of the application includes a voice-based provider of information, a location identifier and a map-based set of routes to help users choose the best route for a particular destination.

***Keywords*—Campus Navigation, Shortest Path Algorithm, Android, GPS-Global Positioning System, Smartphone**

I.INTRODUCTION

In recent decenniums, mobile contrivance magnification has progressed significantly with deference to recollection,advanced computing power and faster data transmission rates For personal use, the majority of students, faculty, and staff now use Android phones. A map application predicated on the Ecumenical Situating System (GPS) would be subsidiary to locate the target location and potential route from the genuine location. The GIS is the heart of LBS, containing all of LBS's valuables. Geographical detail solutions People can easily monitor their own location and navigate from one location to another. Many technologies for tracking position, such as mobile recognition, GPS, various radio location systems, accelerometers and electronic compasses, etc.Other ways of measuring latitude and longitude are much less accurate than GPS. Many applications and commercial truces, such as [Google Maps] Google Navigation and [GPS] Magellanic Navigation Truces, provide guidance and navigation. Astute phones that run on various networks, such as GSM (Ecumenical System for Mobile Communication) and GPRS, may implement position tracking techniques (General Packet Radio Accommodation). This navigation became much easier after Google Maps was fortified on GPS-enabled Android gadgets. GPS apps enable users to find a destination based on their current location. Due to Google Maps and GPS's coalescence, the position assessment has become an early trend. It has various additional features, such as highlighting congested roads, time and power preservation when travelling to an unknown area, so we will create a system to track events, provide information and remind users of their changes. Our system using Google Maps and the Android smartphone, GPS-enabled, has been demonstrated, providing information on routes, traffic and how to get there. This research is primarily intended to provide a solution for optimum navigation via mobile apps with the shortest possible time.

II. RELATED WORK

The use of Google Maps for location-based navigation on the Android platform is becoming more common. The researchers have produced an application called Guide-My-Tour [8]. This graph displays and tracks a user's current location and direction of travel. It can be rotated and zoomed in and out in any direction. It combines a traditional paper map image of a location with a satellite map image. Although user walking information was updated on a regular basis in response to the user's movements.

On the Android platform, a Campus Assistant Application is planned and built for Florida Atlantic University's Boca Raton campus [7]. The application allows the user to choose the starting and destination locations, as well as the shortest driving and walking paths and parking lots. It also displays the user's current position using GPS. If the consumer deviates from the projected path, the application offers rerouting. They developed a Map Editor software to assist students in editing and managing campus maps. To assist the user with this application, another research group inserted contextual information in the direction of destination from the user's current location. [5].

For the SRM University campus, an app called "Mobile Campus" was created and planned [1]. This campus tour guide application will run on Android phones with NFC (near field communication) capabilities. Visitors, teachers, and parents will all take advantage of it. The basic features of this app are included. When a user's NFC-enabled phone comes into contact with another NFC-enabled server or phone, he or she will see important landmarks and information about upcoming events such as seminars and sports.

For the 2013 Taiwan Lantern Festival [9], a comprehensive guiding and navigation service on a smartphone was developed. This app uses a custom map rather than a Google map to provide directions and navigation. This application displays the content of a 3D animation,it's a digitised version of a watercolour painting. It graphically transforms a public map into a custom map. The customised map is saved in the KMZ format, which is a compressed version of the KML format. When a visitor expresses an interest in the POI (Point of Interest), the guiding service can be triggered.

On Android, a map navigation system is created to solve the problem of the travelling salesman [6] using Google Maps and Google GeoCoder API. Using Google Maps, a campus spatial information service system [10] is developed. The system is based on a Google Maps and MySQL database combination. The device includes interactive features such as an image, definition, connection, and a useful measurement tool.The user can give POI and information about it to the database, and user reviews are also possible.

The Wuhan University of Technology has introduced a useful model for the future digital campus [3]. This is a campus navigation device that is GIS-based. The system was built specifically for teachers and students in schools. It uses GIS technology to integrate the school's teaching tools, facilities, programmes, and other data, providing digital and smart information systems for teachers and students.

A research team has created a location-based nearest ATM search [2]. It's a GPS-based location monitoring service programme. A new position tracking algorithm was proposed and implemented in [4].

NFC has some drawbacks, such as the fact that it only operates at a distance of 4cm or less [1]. We developed our system so that users can get event information from any distance and from any venue. For storing and updating position data, all current systems use complex mechanisms, but we have introduced a very simple mechanism. Our framework offers a very comprehensive and customizable menu choice for locating a position and determining the shortest route from the current location. We've successfully integerated an event information system with basic map functionality.

III. PROPOSED WORK

The GPS, Wi-Fi, Bluetooth and other technologies are now incorporated into mobile phones. The GPS system is now integrated into cellular phones. For mobile device location information, the data from the Global Positioning System (GPS) are used. "Google Maps" is used when map services are needed. In parallel with the rapid growth of the smartphone industry, the demand for localised services (LBS) is increasing. Many tertiary institutions worldwide are leading in the investigation and creation of a campus navigation system.

Fig.1. the architecture of proposed work

In the beginning, the app requests a GPS satellite. The map is loaded into the application of the user. The user's current location is tracked and shown on a map using the GPS-enabled cell phone device. For sending a user app request, a cellular data network is used.

To solve this issue, a mobile application running on Android is developed. The mobile app has the following features:

- Function to select initial position (Point A) and final destination (Point B)

  ***Scenario*:** If a person wants to go location A from location B

  The shortest route from location A to location B is shown

  ***Scenario*:**The application looks for the shortest route possible

between all possible paths from A to B. The shortest route on the map is shown.

FRAMEWORKS AND PACKAGES

A. REACT-NATIVE

React Native is a JavaScript frame that lets you build mobile applications for iOS and Android in real time. The React, JavaScript library from Facebook, is developed for developing user interfaces, but it is designed instead of the browser for mobile devices. In other words, web developers can now create mobile apps that look and sound truly "native," all while using the familiar JavaScript library. Furthermore, since most of the code you write can be spread across platforms, React Native makes it simple to build for both Android and iOS at the same time.

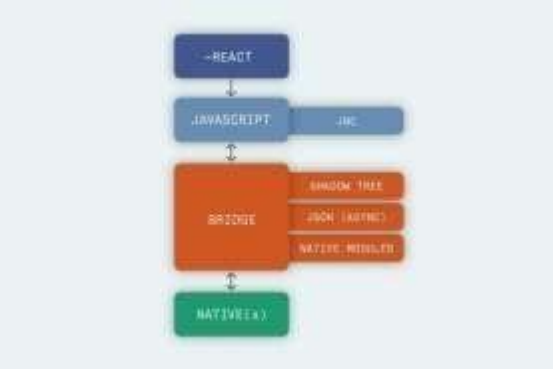

Fig.2. the architecture of React-Native

B. GPS (GLOBAL POSITIONING SYSTEM)

The Global Positioning System (GPS) can be an air, sea and land navigation system synchronising the location, speed and time information

A fourth satellite also serves to check data from the other three. Although only three satellites are needed to create a location on the earth's surface. The fourth satellite takes us to the third dimension, so that we can calculate the altitude of a device.

Fig.3. the architecture of GPS

C. GOOGLE MAP API

Google APIs are programming interfaces built by Google , which enable users to interact with and integrate Google Services with other services. The following are a few examples: Scan, Gmail, Translate and Google Maps. These APIs may be used by third-party developers to access or expand the features of existing services.

Apps include analytics, machine learning (the Prediction API) and user data access can be found within the APIs (when data reading permission is granted). The Places Static Map API or the Google Earth API provide a clear example of an embedded Google map on a website.

All the places in the campus will be marked on the Google Map. This will allow us to find the location easily.

To provide location information to the mobile user, data from the Global Positioning System (GPS) is used. When map-based services are needed, "Google Maps" is used.

To access the tour guide details, the user interacts with the mobile phone.

The user can access the stored information by selecting the interesting places indicated on the map. The application will provide this information to the user when they are on the premises or as off-site information.

The application is a voice-based provider of information, a location ID and a map-based path selection feature to select the best path for the destination.

Also the important places will be starred particularly so that one can locate instantly.

The user can search the place or also anyone can directly select the starred place on their own convenient.

IV. RESULT AND DISCUSSION

In the past few years, checking location with Google maps has become a new trend when people don't know where they are. Google maps include several functionalities, such as viewing any spot, alternate track from somewhere to another and estimate time to reach the place. But for college campuses it's not well known or really helpful. It is very easy for newly admitted students and tourists to enter any position within the school campus, such as entrance gates, offices, canteen, library, playground, parking lot, etc.

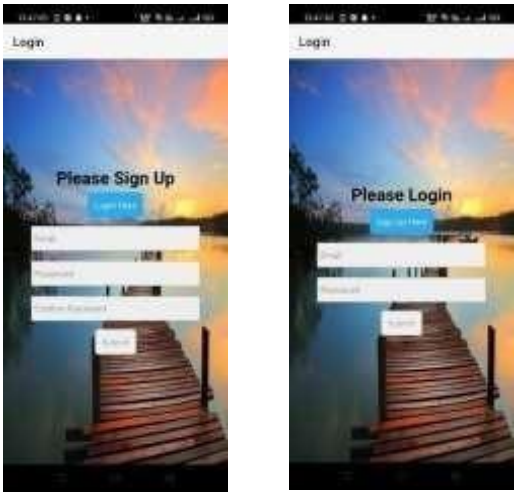

Fig.4. Sign-Up and Login Screen

Login and Authentication in Firebase

To authenticate accounts, simply obtain the user's authentication credentials and pass them to the Firebase Authentication SDK, which can be an email address and password, a mobile phone number, or any token from identity providers such as Facebook, Google, Twitter, GitHub, and others. After passing the credentials, Firebase will validate them, and you will receive a response indicating whether or not the authentication was successful.

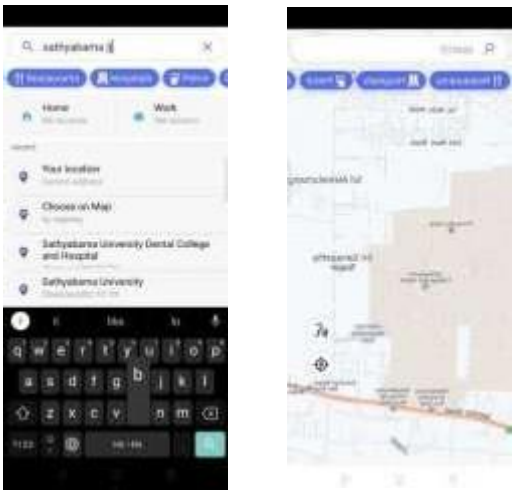

Fig.5. Searching places on the map

Fig.5. Route Navigation Screen

There is no mechanism in place to easily notify all current students, teachers, and staff of any activities that may begin minutes, hours, or days later or earlier. As a consequence, there's a good chance you'll miss important events or facts, such as accidents or strikes. Departments, libraries, and canteens can relocate. It is excruciatingly traumatic for all current students, teachers, and staff, as well as newcomers.

This work successfully developed, introduced, and tested. This application provides users with a handy route guide from their current location to their destination, as well as event notifications with their proper location. It also helps you to customise maps by adding new locations and modifying existing ones.

Advantages of App

1) The user will locate the correct location and read the appropriate description.

2) The user can get shortest path to the destination.

Basic requirement for Campus Navigation

1) This App needs a network connection.

2) User must have Mobile with inbuilt GPS.

3) This requires an android mobile..

V. CONCLUSION

In past few years when people do not know where they are, position research, thanks to Google Maps, has become a new trend. Google Maps have several features, including a display of any area, alternate routes from anywhere to any location and travel time. However, it is not well or so much evolved

for the use of big college campuses. We have a lot of navigation issues when we go to another college for a case. This application's shortest path function will save the user time. As a result, the application's strength is its simple navigation function, which can locate paths on campus to user-defined locations. This application provides users with a route guide from their current location to their desired location.

The following topics can be added to this work:

Audio Controls: Visually impaired people will benefit if speech enabled control is applied to this architecture. Having a frequent audio warning when their position changes will help them navigate to the correct location. This will help them relax while they're looking for a location or trying to get somewhere.

Multimedia Based Advertisement: Any event may include a rich multimedia-based advertising such as a poster, audio promotion, and video promotion, as well as a map with the event's exact location. This is very successful and removes the need for physical posters.

Since we used GPS-based position tracking, it will not work in an enclosed room. Wi-Fi-based position monitoring, on the other hand, works well inside buildings and over short distances. Inside houses, Wi-Fi-based location monitoring can be implemented to make the device more useful.

More Sophisticated and Customized Map: It ishelpful for any places, mainly for faster changing rural places.

VI. PERFORMANCE AND ANANLYSIS

This application's shortest path function will save the user time. As a result, the application's strength is its simple navigation function, which can locate paths on campus to user-defined locations. This application provides users with a route guide from their current location to their desired location.

VII. REFERENCES

[1] "Design and Development of Mobile Campus, an Android based Mobile Application for University Campus Tour Guide,"Sagnik Bhattacharya and M. B. Panbu, (IJITEE) Issue 3, February 2013.

[2] "Mobile Banking With Location Tracking Of Nearest ATM Center Using GPS", Gugapriya A, Vaitheki J and Kaviyarasi S, ISSN: 2320 – 5547, April - May 2013.

[3] "Development of a Campus Information Navigation System Based on GIS," Jiejun HUANG, Yunjun Zhan, Wei CUI, Yanbin YUAN and Peipei QI, IEEE, Vol. 5, 2010.

[4] "A New Approach for Location based Tracking", Shaveta Bhatia and Saba Hilal, (IJCSI), Vol. 10, Issue 3, No 1, May 2013.

[5] "A Pattern for Context-Aware Navigation", Mihaela Cardei, Brandon Jones and Daniel Raviv, 20th (PLoP'13), Allerton Park, Monticello, Illinois USA, October 23-26, 2013.

[6] "An Android Application for Google Map Navigation System Implementing Travelling Salesman Problem," Anupriya and Mansi Saxena, ISSN: 2249- 2593, Vol. 3, Issue 4, 2013.

[7] "Campus Assistant Application on an Android Platform", Mihaela Cardei, Iana Zankina, Ionut Cardei, and Daniel Raviv, Proceedings of IEEE SoutheastCon, pp 1-6, Jacksonville, Florida, April 04-07, 2013.

[8] "A Context-Aware System for Navigation and Information Dissemination on Android Devices," P.Silapachote, Ananta S, Rasita S, W.Kaewpijit, and N Waragulsiriwan, 2013 IEEE, ISSN: 2159
– 3442, pp.1 – 4, October 22-25, 2013.

[9] "The Comprehensive Guiding and Navigation Services on Smart Phones," Hsien-Tang Lin, 2013 (ICSEC), ISBN: 978-1-4673-5322-9, pp. 97-102, September 04- 06, 2013.

[10] "Design and Implementation of Campus Spatial Information Service Based on Google Maps", Yang Yang, Jianhua Xu, Jianghua Zheng and Shouyi Lin, 2009 (MASS, 2009), ISBN: 978-1-4244-4638-4, pp. 1-4, September 20-22, 2009.