

INTELLIGENT BUS NOTIFICATION SYSTEM

Submitted in partial fulfillment of the requirements for the award of
Bachelor of science Degree in Computer Science

By

VIJAYA BHARATHI.S(Reg. No.38290101)

SRINIVASAN.V(Reg. No.38290087)



**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
SCHOOL OF COMPUTING**

**SATHYABAMA
INSTITUTE OF SCIENCE AND TECHNOLOGY
JEPPIAAR NAGAR, RAJIV GANDHI
SALAI, CHENNAI-600119, TAMILNADU**

APRIL 2021



SATHYABAMA
INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)
Accredited with Grade “A” by NAAC
(Established under Section 3 of UGC Act, 1956)
JEPPIAAR NAGAR, RAJIV GANDHI SALAI, CHENNAI-600119
www.sathyabama.ac.in



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of **VIJAYA BHARATHI.S(38290101)** and **SRINIVASAN.V (38290087)** who carried out the project entitled “**INTELLIGENT BUS NOTIFICATION SYSTEM**” under our supervision from November 2020 to April 2021.

Internal Guide

Dr.A.Viji Amutha Mary, M.Tech., Ph.D.,

Head of the Department

Dr.L.LAKSHMANAN, M.E., Ph.D.,

Submitted for Vivavoce Examination held on _____

Internal Examiner

External Examiner

DECLARATION

I, **VIJAYA BHARATHI (Reg. No. 38290101)** hereby declare that the Project Report entitled" **INTELLIGENT BUS NOTIFICATION SYSTEM**" done by me under the guidance of **Dr.A.Viji Amutha Mary,M.Tech.,Ph.D.**, is submitted in partial fulfillment of the requirements for the award of Bachelor of science degree in Computer Science.

DATE:

PLACE:CHENNAI

SIGNATUREOF THE CANDIDATE

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to Board of Management of **SATHYABAMA** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr. T. Sasikala M. E., Ph.D., Dean**, School of Computing, **Dr. S. Vigneshwari M. E., Ph. D.,** and **Dr. L. Lakshmanan, M. E., Ph. D., Heads of the Department** of Computer Science for providing me necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide **Dr.A.Viji Amutha Mary, M.Tech., Ph.D** for her valuable guidance, suggestions and constant encouragement paved way for the successful completion of my project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the **DEPARTMENT OF COMPUTER SCIENCE** who were helpful in many ways for the completion of the project.

ABSTRACT

Now days, it is not possible for a new person to determine which bus is coming at the bus terminal. They are always dependent on somebody to help them out on the same issue. So we have come up with a project which would allow the person to determine the buses which are coming at the bus terminal and also give the information about the route through which they are going.itgives an overview of the Mobile Application developed for bus tracking and notification system for Transport Systems.It also include user-friendly application for user to track bus on smart their phones. The smart system designed has both online (GPS) and offline (GSM) for user friendly service. The obvious reason behind is increased personal vehicle, causing traffic jams. The requirement of a reliable public transport monitoring system is actually the outcome from this unnecessary congestion of traffic observed. Since most of the modern technologies for the Intelligent Transportation System (ITS) have developed since more than 15 years. The need of a Global Positioning/ inertial (GPS/INS) augmentation with a reliable communication system stems from the shortcomings of each individual technology.Smart city is very much important for developing country now days
Leads to Development of technology has allowed the concept of an intelligent transport system

TABLE OF CONTENTS

CHAPTER NO.	TITLE	PAGE NO.
	ABSTRACTV	
	LIST OF FIGURES viii	
	LIST OF ABBREVIATIONS	ix
1	INTRODUCTION	10
1.1	OVERVIEW OF PROJECT	10
2	LITERATURE REVIEW	11
	3AIM AND SCOPE OF PRESENT INVESTIGATION	13
3.1	AIM OF THE PROJECT	13
3.2	SCOPE AND OBJECTIVE	13
3.3	SYSTEM REQUIREMENTS	13
3.3.1	HARDWARE REQUIREMENTS	13
3.3.2	SOFTWARE REQUIREMENTS	14
3.4	SOFTWARE USED	14
3.4.1	ANDRIOD SDK	14
3.4.2	NODE.JS	15
3.4.3	VS CODE IDE	15
3.5	OPERATIONAL REQUIREMENTS	16
3.5.1	GPS APPLICATIONS	16
3.5.2	RTLS	16
4	EXPERIMENTAL OR MATERIAL METHODS	17
4.1	DESIGN METHODOLOGY	17
4.1.1	EXSISTING SYSTEM	17
4.1.2	PROPOSED SYSTEM	17
4.1.3	INTELLIGENT TRANSPORT SYSTEM	17
4.2	MODULE DESCRIPTION	18
4.3	ARCHITECTURE	21

	4.3.1 PROCESS EXPLANATION	22
	4.3.2 ARCHITECTURE EXPLANATION	23
5	RESULTS AND PERFORMANCE ANALYSIS	25
6	CONCLUSION AND FUTURE ENHANCEMENT	28
	6.1 CONCLUSION	28
	6.2 FUTURE ENHANCEMENT	28
	REFERENCES	29
	APPENDIX	30
	SOURCE CODE	

LISTOFFIGURES

FIGURE NO:	FIGURENAME	PAGE NO
4.3.1	Flow of process	21
4.3.2	Architecture	23
5.1	Login page	25
5.2	Home page	26
5.3	Stops page	27

LISTOFABBREVIATIONS

GSM	Global system for mobile application
GPS	Global positioning system
ITS	Intelligent transport system
RF	Radio frequency
SDK	Software development kit
JS	Java script
VS	Visual studio
RTLS	Real-time locating systems

CHAPTER

1INTRODUCTION

1.1 OVERVIEW OF PROJECT

Smart city is very much important for developing country now days. The smart life style of today allows us to arrive at our destination as soon as possible. The changing lifestyle is also important with the safety and comfort of vehicle. Development of technology has allowed the concept of an intelligent transport system. Now a day's tremendous growth increases in transport system. The system proposed in this paper states smart way to public transport. Considering all the public issues and problems such as; time, easy of transport, safety and user-friendly environment this system is been designed.

Now days, it is not possible for a new person to determine which bus is coming at the bus terminal. They are always dependent on somebody to help them out on the same issue. So we have come up with a project which would allow the person to determine the buses which are coming at the bus terminal and also give the information about the route through which they are going.

Now when bus having a Bluetooth device comes in range of the handset Bluetooth, notification will appear on the android mobile. This notification also consists of the Mac-id of the sending device, we will use this Mac-id and send the Mac-id of the connecting device to the cloud server by performing Jason parsing (a technique used for communicating android device with the cloud server). At the cloud server end the Mac-id would be compared with available db and the corresponding bus number, name & the route would be send to the request handset. When our handset receives the result of the query system would display the result to him on his handset application.

This would help the person to determine which bus is coming at the bus terminal and what is its route along with the stops without relying on anybody. The above stated system would be implemented using android. The Advantages of the system is that will get to know which bus is going in which direction. Its Time saving. It Identify the route. System gets to know faster which bus is arriving at the stop.. System needs of android device. The Application of the system is that it provides route and stops of the buses of the remote areas.

CHAPTER

2LITERATURESURVE

Y

Vehicle tracking systems are commonly used by fleet operators for fleet management functions such as fleet tracking, routing, dispatching, on-board information and security. Some vehicle tracking systems are bundled with or interface with fleet management software. Along with commercial fleet operators, urban transit agencies use the technology for a number of purposes, including monitoring schedule adherence of buses in service, triggering automatic changes of buses' destination sign displays once the vehicle approaches the bus terminus (or other set location along a bus route such as a particular bus stop along the route), and triggering pre-recorded (or even synthetic speech) bus stop, route (and its destination) or service announcements for passengers.

The American Public Transportation Association estimated that, at the beginning of 2009, around half of all transit buses in the United States were already using a GPS-based vehicle tracking system to trigger automated stop announcements. This can refer to external announcements (triggered by the opening of the bus's door) at a bus stop, announcing the vehicle's route number and destination, primarily for the benefit of visually impaired customers, or to internal announcements (to passengers already on board) identifying the next stop, as the bus (or tram) approaches a stop, or both; the latter are often also displayed on an internal LED display or LCD monitor connected to the system while the loudspeakers play them. Data collected as a transit vehicle follows its route is often continuously fed into a computer program which compares the vehicle's actual location and time with its schedule, and in turn produces a frequently updating display for the driver, telling him/her how early or late he/she is at any given time, potentially making it easier to adhere more closely to the published scheduled. Such programs are also used to provide customers with real-time information as to the waiting time until arrival of the next bus or tram/streetcar at a given stop, based on the nearest vehicles' actual progress at the time, rather than merely giving information as to the *scheduled* time of the next arrival. Transit systems providing this kind of information

assign a unique number to each stop, and waiting passengers can obtain information by entering the stop number into an automated telephone system or an application on thetransit system's website.

The first study that we conducted is to prove the existence of the problem of buses being late to bus stops. At the same time, we assessed the level of significant of the problem. We observed the expected arrival time to each bus stop for a period of time for two weeks. The result of the observation is as expected that is most of the time the bus failed to arrive at the bus stop as scheduled. Furthermore, we had got a shocking observation result that is sometime the bus skipped its trip and causing the passengers to wait for the next bus that might arrive an hour later.

Besides, sometimes the bus is late to arrive at the first bus stop for more than forty five minutes. It is causing the passengers for the entire route to be delayed. In addition, the problem happens even during peak hours where most of the passengers are rushing to their working place. These observation results successfully proved that our problem.

statement is valid and these problems have to be solved urgently for the benefit of the city

and the bus passengers.

The second study in meaning to understand the current demand and the market acceptance for our proposed solution via survey. We dispatched one hundred survey

forms to passengers in bus stops to understand the culture of the passengers, the need of

our proposed solution, and the actual condition of public transportation.

The result of our survey clearly shows that majority of the passengers are workforce uses bus to travel to working place daily. The major problems faced by the passengers are

most of the time the buses do not reach the bus stop on time. It causes the passengers have to wait for an unpredictable time frame. In addition, majority of the passengers

demand for SMS service to update them with the bus arrival time. With such information, we are very positive that our proposed solution will easily be accepted by bus passengers.

Now, we are sure that the problem exists and current bus passengers are demanding for such a service. Thus we derived some hypotheses to assist us in understanding the factors

which are affecting the bus's arrival time. These hypotheses are extremely important for

us to understand the parameters involved in this system.

CHAPTER3

AIMANDSCOPEOFPRESENTINVESTIGATION

AIMOFTHEPROJECT

Aims at giving passengers the information about the departure of the subscribed bus from the starting point or the passengers can also subscribe for the number of stops from which he wants the notification and also can arrive at the busstop at the time of departure.

SCOPEANDOBJECTIVE

To create an application which is able to estimate and predict the arrival time of public transportation to passengers by taking traffic congestion factor into consideration.

To offer a convenience mobile information service which ensures passengers are able to organize multiple personal daily planning efficiently.

To offer a mobile information service which can assist passengers to eliminate the waiting time for the arrival of bus at a possibly unpleasant or unsafe bus stop. To make more accessible and constantly updated dynamic bus information to bus passengers.

SYSTEMREQUIREMENTS

HardwareRequirements

○ Hard Disk : 40GB and Above

○ RAM : 512MB and Above

○ Processor : PentiumIIIandAbove

SoftwareRequirement

- Andriod SDK
- Node.js
- VS code IDE

SOFTWARE USED

3.4.1ANDRIOD SDK

The Android SDK (software development kit) is a set of development tools used to develop applications for the Android platform that has become Apple's biggest rival in the smartphone space. The Android SDK includes the following:

- Required libraries.
- Debugger.
- An emulator.
- Relevant documentation for the Android application program interfaces (APIs).
- Sample source code.
- Tutorials for the Android OS.

This SDK provides a selection of tools that are required to build Android applications and ensures the process goes as smoothly as possible. SDKs can include APIs (or multiple APIs), IDE's, Documentation, Libraries, Code Samples, and other utilities.

3.4.2 NODE.JS

Node.js is a platform built on Chrome's JavaScript runtime for easily building fast and scalable network applications. Node.js uses an event-driven, non-blocking I/O model that makes it lightweight and efficient, perfect for data-intensive real-time applications that run across distributed devices.

Node.js is an open source, cross-platform runtime environment for developing server-side and networking applications. Node.js applications are written in JavaScript, and can be run within the Node.js runtime on OS X, Microsoft Windows, and Linux.

Node.js also provides a rich library of various JavaScript modules which simplifies the development of web applications using Node.js to a great extent.

3.4.3 VS CODE IDE

Visual source code is a freeware source-code editor made by Microsoft for Windows, Linux and macOS. Features include support for debugging, syntax highlighting, intelligent code completion, snippets, code refactoring, and embedded Git. Users can change the theme, keyboard shortcuts, preferences, and install extensions that add additional functionality. Visual Studio Code is a source-code editor that can be used with a variety of programming languages, including Java, JavaScript, Go, Node.js, Python and C++. It is based on the Electron framework, which is used to develop Node.js Web applications that run on the Blink layout engine. Out-of-the-box, Visual Studio Code includes basic support for most common programming languages. This basic support includes syntax highlighting, bracket matching, code folding, and configurable snippets. Visual Studio Code also ships with IntelliSense for JavaScript, TypeScript, JSON, CSS, and HTML, as well as debugging support for Node.js. Support for additional languages can be provided by freely available extensions on the VS Code Marketplace.

OPERATIONAL REQUIREMENTS

3.5.1 GPS APPLICATIONS

GPS has global coverage but can be hindered by line-of-sight issues caused by buildings and urban canyons. RFID is excellent and reliable indoors or in situations where close proximity to tag readers is feasible, but has limited range and still requires costly readers. RFID stands for Radio Frequency Identification. This technology uses electromagnetic waves to receive the signal from the targeting object to then save the location on a reader that can be looked at through specialized software.

3.5.2 RTLS

RTLS(Real-Time Locating Systems) are enabled by Wireless LAN systems (according to IEEE 802.11) or other wireless systems (according to IEEE 802.15) with multilateration. Such equipment is suitable for certain confined areas, such as campuses and office buildings. RTLS requires system-level deployments and server functions to be effective.

EXPERIMENTAL OR MATERIAL METHODS

4.1 DESIGN METHODOLOGY

4.1.1 EXSISTING SYSTEM:

In the existing system the user has no idea of the location of the bus and the degree of traffic in that particular route. Basically, the main thing where passengers do in step 1 planning stage is just to decide how much time they need to spend for waiting at the bus stop in case the bus doesn't run on the schedule. Their initial plan may completely get affected if the bus doesn't arrive at their anticipated time. That means a single factor may ruin their planning. Nobody would wonder, waiting, frustration, exposure to weather waiting at the bus stop is disliked by most of the frequent bus passengers. Nevertheless, they can do nothing about that. Complaints to the bus agency will only get the reply with the answer – traffic congestion delayed the entire trip, not having sufficient bus to cater for a particular route and so. For anyone who emphasizes punctuality as their principle will definitely opt out bus as their choice to travel between two distances.

4.1.2 PROPOSED SYSTEM:

Here the design and implementation of smart public Transport system for smart cities is proposed. A detailed survey concluded for implementation of various features to the system prototype. Features such as; smart application for user friendly use, GPS/GSM system interface, emergency and bus fail switch, etc included. After we identified the problem and the possible solution we took initiative to conduct a few studies on bus passengers in order to affirm our problem statements and justify our proposed solution.

4.1.3 INTELLIGENT TRANSPORT SYSTEMS

Intelligent transport systems vary in technologies applied, from basic management systems such as car navigation; traffic signal control systems; container management systems; variable message signs; automatic number plate recognition or speed cameras to monitor applications, such as security CCTV systems, and automatic incident detection or stopped vehicle detection systems; to more advanced applications that integrate live data and feedback from a number of other sources,

such as parking guidance and information systems; weather information; bridge de-icing (US deicing) systems; and the like. Additionally, predictive techniques are being developed to allow advanced modelling and comparison with historical baseline data. Some of these technologies are described in the following sections.

4.2 MODULE DESCRIPTION

The module consists of a user interface which gets input from the user. The inputs include bus number, name of the bus stop and threshold time. The threshold time implies the time at which the user should be notified about the arrival of the bus.

The information from the user interface is transferred to the web server (Local Host) which validates it with the help of the database.

Then it is transferred to the processor which sends signal to the buses based on the input using GSM

It keeps tracking the corresponding bus(i.e.) it monitors the arrival time of all the buses with the same number and find the bus with the minimal arrival time using GPS, GSM and Speed Sensor

Once the bus with the minimal arrival time is found, it starts comparing the arrival time of the bus with that of threshold time given by the user. When both are equal, a notification message is sent to the user.

The present invention relates to a system and methods for notifying passengers of an approaching vehicle. Utilizing this invention, passengers can remain in a safe, controlled environment, avoiding harsh environmental conditions and excessive waiting times, instead arriving at their pick-up point just prior to a vehicle's arrival. More specifically, the present invention relates to a bus notification system that will provide a passenger with adequate warning of an approaching bus well in advance of its arrival at the bus stop.

In many cities and towns, school systems are required to provide transportation to and from school for children living more than a specified distance from school. Generally this transportation is in the form of busing whereby school buses pick up

school children at several bus stops along several bus routes and then deliver the children to their school. The arrival time of a school bus at a given bus stop can vary significantly from day to day for any of a number of reasons. As a result, children typically arrive at the bus stop well before the bus is expected to arrive to avoid missing the bus. These children frequently lack parental supervision.

Furthermore, on inclement weather days, children waiting for their bus are exposed to harsh weather conditions including rain, snow, or extreme cold. To avoid this situation, concerned parents frequently wait with their children at the bus stop in a car, causing unnecessary pollution.

Buses can be delayed for numerous reasons including inclement weather, such as fog, snow, ice or extreme cold, which reduces the speed of the bus and thus impedes the arrival of the bus at the planned time. Similarly, bus mechanical problems, heavy traffic or substitute bus drivers can generate lengthy delays in bus arrival time. In addition, many school buses will make several sequential runs to pickup children for different schools such as the high, middle and grade schools for a town.

A delay in picking up children in the first run can result in similar or longer delays for subsequent runs. Correspondingly, it is possible, when there is good weather and light traffic conditions, for the bus to arrive at the bus stop earlier than the planned time.

In areas where bus routes cover many miles there frequently is a large variance in arrival time of a bus at the bus stops along the route and in many instances the bus stop will not be visible from the home. Similarly in areas where children live closer together, frequently there are communal bus stops for several children such that the bus stop is not visible from the home of each child. Advance notification of bus arrival also allows time to prepare the children for school without rushing to catch the bus.

In a preferred application of such a notification system, children and their parents are alerted to the approach of a school bus to pick up the children at a school bus stop. The accurate notification of an approaching school bus can minimize if not significantly eliminate the children's wait at the bus stop and consequently their exposure to inclement weather. Children can spend a few extra minutes inside their home with their parents instead of enduring lengthy waits at the bus stop in potentially adverse conditions. In addition to notifying households of an approaching bus picking

children up to go to school, the receiver will also alert parents of an approaching school bus dropping children off at the end of the school day.

The system includes the means to accurately determine the location of the vehicle and the location of any specified vehicle stop so that the distance between the vehicle and the vehicle stop and the corresponding time can be accurately measured. Global positioning satellite (GPS) technology generates extremely accurate location coordinates by analysis of signals from a plurality of global positioning satellites. Receiving units within the range of the transmission apparatus' RF signal receive the GPS determined location of the vehicle regardless of the strength of the RF signal and calculate the distance between the vehicle and the vehicle stop using GPS determined locations for both the vehicle and the vehicle stop.

Unlike systems relying on signal strength to determine the distance that are subject to inaccuracies resulting from obstructions in the wireless radio frequency path, the vehicle location as determined by GPS is independent of signal strength. As a result, a receiving unit can accurately calculate an associated arrival time and the distance between receiving unit and vehicle, thereby providing accurate arrival information.

In specific embodiments, the system further includes the means to determine if an approaching vehicle is traveling in a specified direction so that passengers are only notified when a vehicle is approaching a vehicle stop from a specified direction. The transmission apparatus further includes the means to determine the direction of vehicle travel by GPS technology and transmit a signal further including information about the direction of vehicle travel. Each receiving unit further compares the received direction of vehicle travel with a specified direction of vehicle travel such that an alarm function is activated when the vehicle is closer than a specified threshold location

In preferred embodiments of the invention, the receiving unit further includes the means to acquire information about a vehicle by a simple information acquisition protocol herein referred to as the learn function. Activating the learn function when a vehicle approaches the vehicle stop causes the receiving unit to acquire the vehicle route number and direction of vehicle travel from the vehicle transmission unit signal. The receiving unit location coordinates are also acquired by the receiving unit from the bus transmission unit signal. After the learn function has been activated, the

receiver listens for the transmission signal of greatest amplitude.

When signal amplitude is maximized and distance between the transmitter and receiver is minimized, the bus stop location is stored into the receiver memory. Alternatively, the bus stop location can be simply stored into the receiver memory when the learn button is depressed.

4.3 ARCHITECTURE

The system architecture is as follow.,

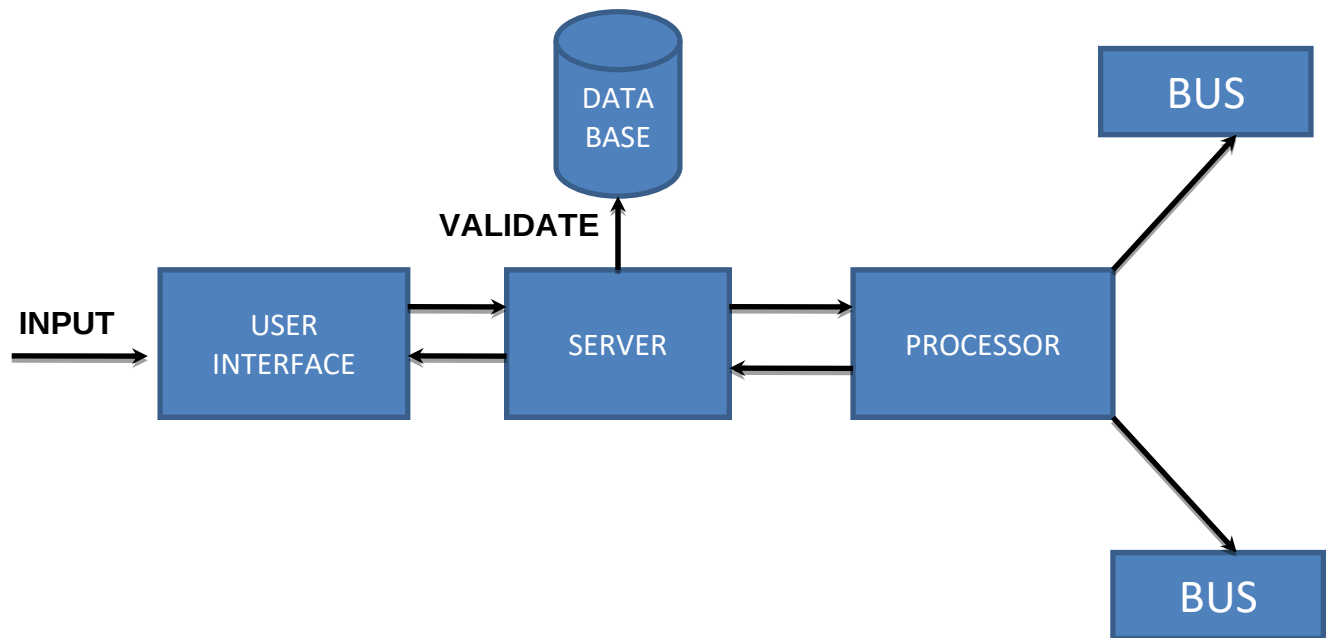


FIG 4.3.1 FLOW OF PROCESS

4.3.1 PROCESS EXPLANATION:

STEP 1

The first step before a passenger board a bus is to plan a journey. Among the concernsthat is taken into account during the planning phase are consisting where

he/she want to go, what time he/she plan to depart, when the bus will arrive and so on.

STEP 2

After gone through the initial planning stage, the passenger should have decided the departure time based on the reference on bus schedule.

STEP 3

Upon deciding the departure time, the passenger is ready to leave for his/her destination. He/She will wait at the bus stop without any prior notice of bus arrival time. The bus may arrive on schedule or it may arrive later than the expected time due to some external and stochastic factor such as traffic condition and accident.

STEP 4

The last stage obviously is the arrival of the bus regardless the bus arrive on time or arrive late to that particular bus stop, The question marks besides the bus picture has implied that the arrival time of a bus is never predictable and the time is subject to changes.

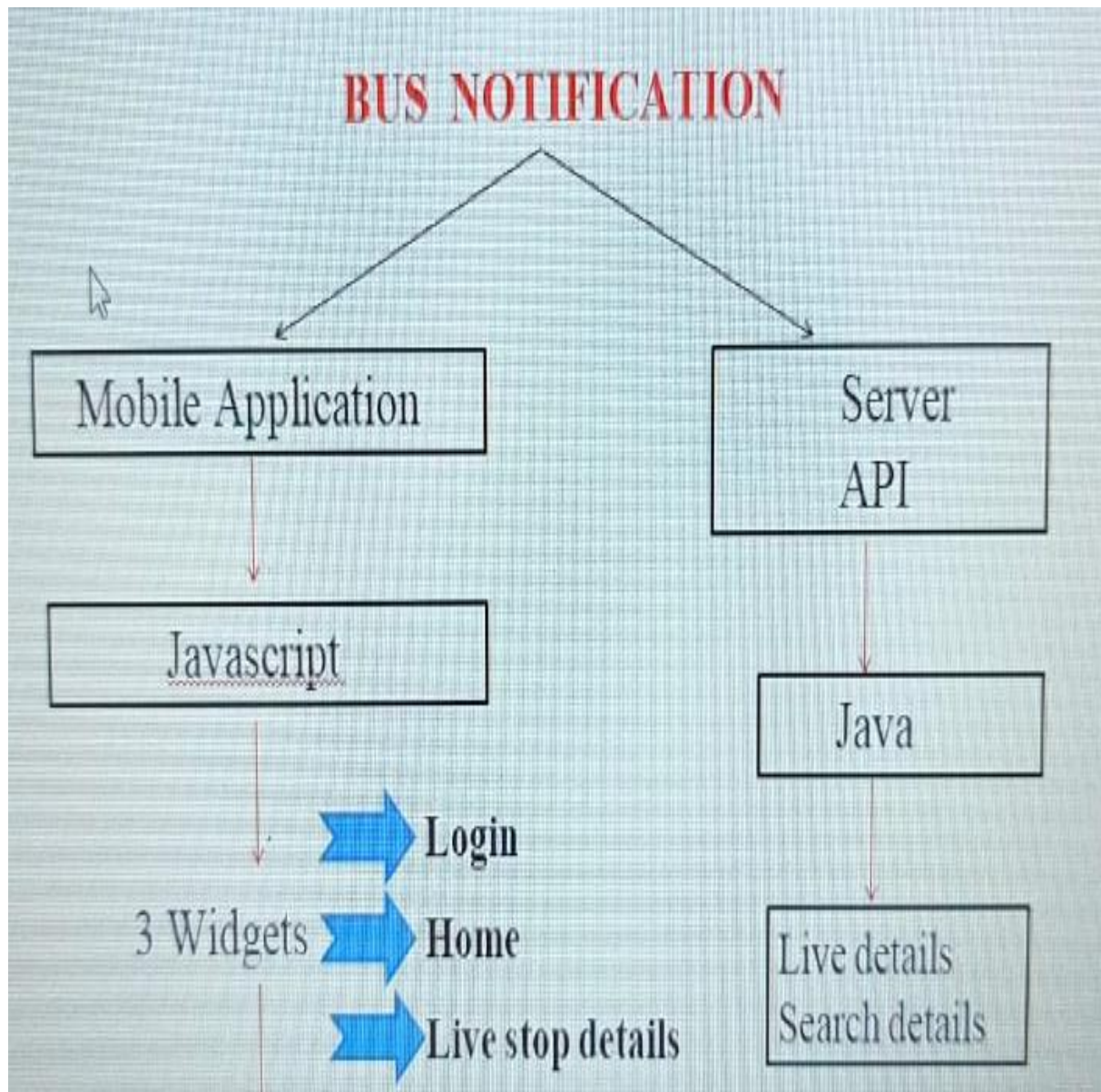


FIG.4.3.2 ARCHITECTURE

4.3.2 ARCHITECTURE EXPLANATION:

Firstly, online passengers are required to choose from a list of six-selection menu. Upon choosing the Agency, it is followed by Route/Line. Each bus agency has its own coverage route. The fourth data that need to be chosen is the direction. After that, passengers need to enter their From Stop (Origin) and To Stop (Destination). The purpose of entering this information is due to the reason that sometimes there are multiple routes which might go to the passenger's destination.

All possible routes that serve passenger's destined stops can be predicted, as long as the passengers specify both stops (Origin and Destination) correctly. By selecting particular destination, only the prediction of passenger's desired bus will be displayed. Besides obtaining the static arrival time, passengers have the option to see the Live Map of all the moving bus. Functions that can be found are zoom in, zoom out, lines (extra route on the map) and help. The third facility provided by is the availability of bus schedule. The existing bus schedule will serve as guidance for travelers to plan their journey. Coincidentally, the objective is to tackle the problem of unpredictable bus arrival time. This problem always cause travelers to miss the bus because the bus can be either arrives earlier than expected or may cause the travelers to wait for a late-coming bus as there is no prior knowledge of the bus arrival information.

CHAPTER 5

RESULT AND PERFORMANCE ANALYSIS

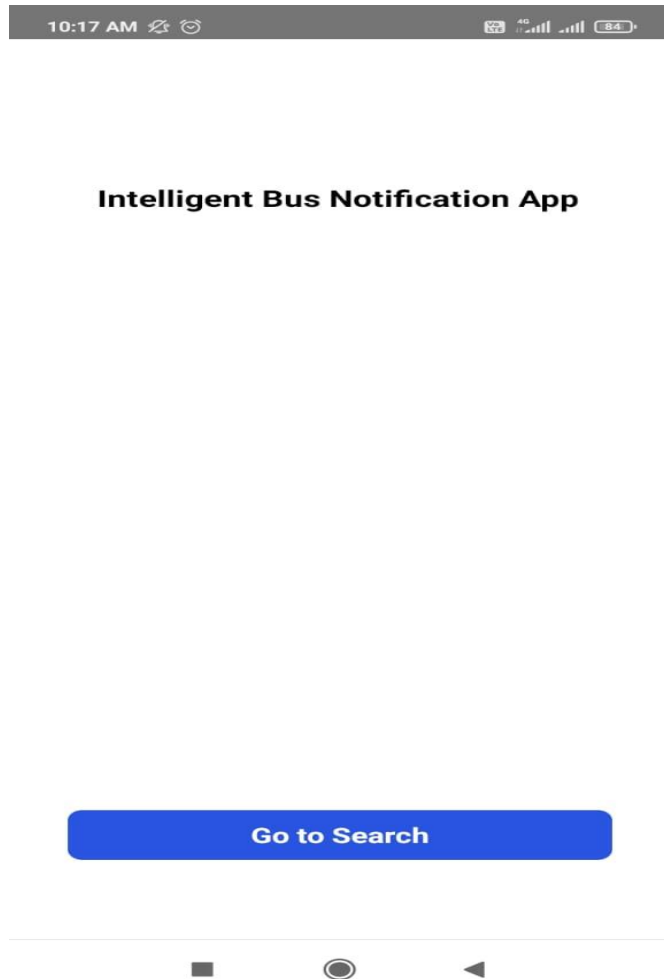


FIG.5.1 LOGIN PAGE

The above fig.5.1 depicts the login page of the mobile application, where the user can login to the app in search of his bus arrival time to his location.

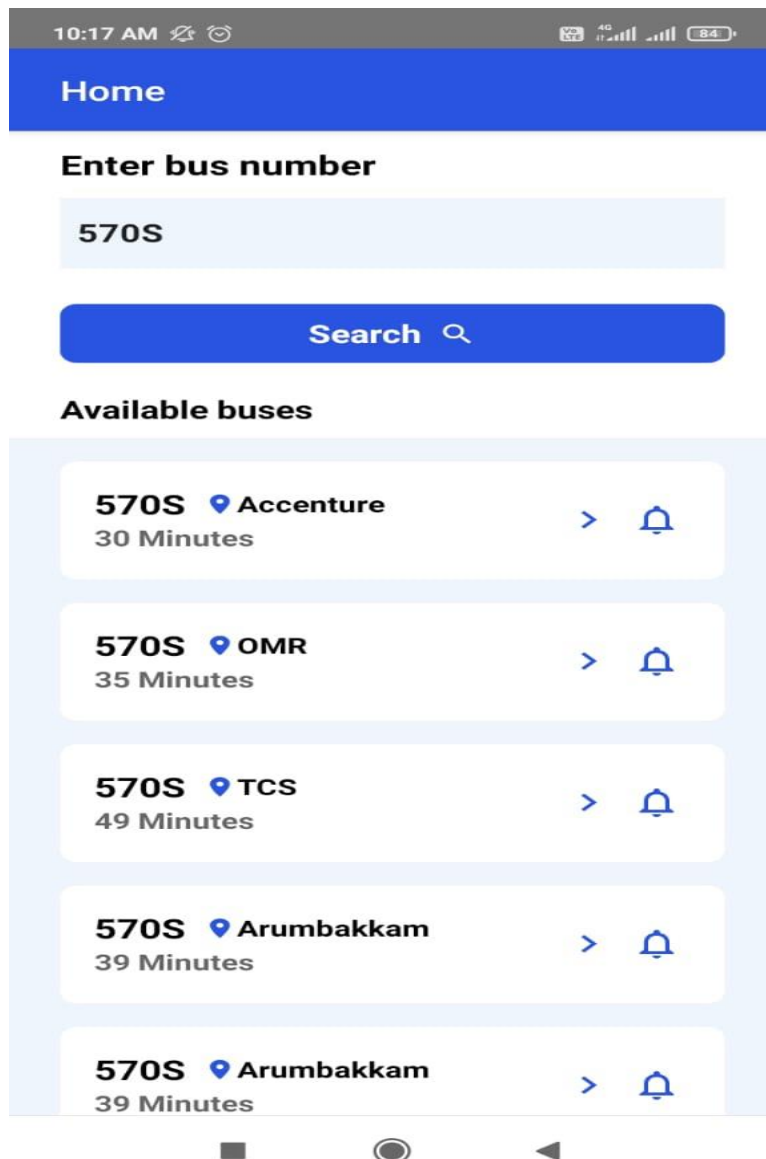


FIG.5.2 HOME PAGE

The above fig.5.2 depicts the home screen of the app, where the user can specify his/her starting point and destination. Then the bus with minimal arrival time to maximum time on the specified route will be displayed so that the user can catch the bus on time without waiting for a long time.

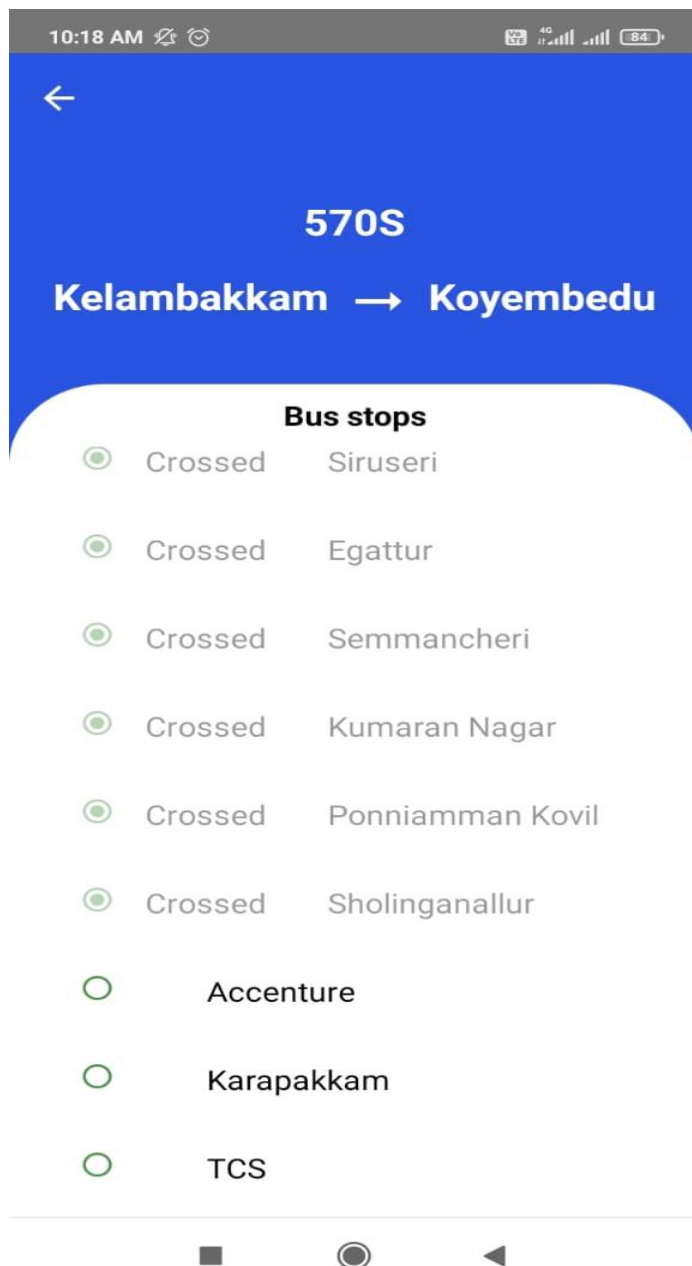


FIG.5.3 STOPS PAGE

The above fig.5.3 depicts the travel route of bus that the user have specified in the home page.this page tracks the route that the bus had crossed and to cross and the time to reach the start or destination point.

CHAPTER 6

CONCLUSION AND FUTURE ENHANCEMENT

6.1 CONCLUSION:

Android operating system is getting more famous day by day and the app is providing the helpful ones their needs. Its more users friendly and easier to have the information about the buses. We can have easy access to the routine as well as provide the navigation of the route. Thus it makes easy for the travelers to travel to their destination place. This app provides a bus route and bus name. From this we can conclude that our system/application can be handfull for common persons who travel through buses in their day to day life.

6.2 FUTURE ENHANCEMENT:

The current version of PredictBus system is a very solid foundation and we had designed PredictBus system to be extremely flexible for future enhancements. To further enhance PredictBus system, we would first increase the factor in ANN prediction whereby we will consider more factors such as the weather. Currently we do not have sufficient data to enable us to draw prediction result with weather factor in consideration. Therefore, we are very positive that the inclusion of weather factor will increase the accuracy of PredictBus system's prediction significantly. Next, we would like to improve the system database's capacity. Currently PredictBus system is able to cope with 2 to 3 buses on one route. However, we are very concerned if the database can manage to cope with more than 10 routes and near 50 buses.

REFERENCES:

- [1] Mei Chen, Xiaobu Liu, Jingxin Xia, Steven I.Chien, "A Dynamic Bus-Arrival Time Prediction Model Based on APC Data", Journal of Computer-Aided Civil and Infrastructure Engineering, United Kingdom, Volume 19, 2004, pg 364-376

- [2] Wei-Hua Lin, Jian Zeng, "An Experimental Study On Real Time Bus Arrival Time Prediction With GPS Data", Journal of Center for Transportation Research and Department of Civil and Environmental Engineering, Virginia

- [3] Jens Biesterfeld, Elyes Ennigrou, Klaus Jobmann, "Neural Networks for Location Prediction in Mobile Networks", Journal of Institute far Allgemeine Nachrichtentechnik, University Hannover, Germany

- [4] Lajos KISGYORGY, Laurence R.RILETT, "Travel Time Prediction By Advanced Neural Network", Journal of Budapest University of Technology and Economics, Texas, 10 October 2001

- [5] N.Davey, S.P.Hunt, R.J.Frank, "Time Series Prediction and Neural Networks", Journal of Unviersity of Hertfordshire, Hatfield, United Kingdom.

APPENDIX

SOURCE CODE

1. Login Page

```
import React, { useState } from 'react';
import { ActivityIndicator } from "react-native";
import styled from 'styled-components/native';
import Colors from '../constants/colors';

interface Props {
  setUserLogin: any
}

const LoginScreen = ({ setUserLogin }: Props) => {
  const [loading, setLoading] = useState<boolean>(false);
  const _logInUser = () => {
    setLoading(true);
    setTimeout(() => {
      setUserLogin(true);
    }, 1000);
  }
  return (
```

```

<>
<Screen>
  <LoginText>
    Intelligent Bus Notification App
  </LoginText>
  {
    loading ? <ActivityIndicator size={50} color={Colors.blue} /> :
    <SignInButton onPress={_logInUser} activeOpacity={0.3} >
      <SignInText>Go to Search</SignInText>
    </SignInButton>
  }
</Screen>
</>
);
};

```

```

const Screen = styled.View`
  flex: 1;
  paddingHorizontal : 32px;
  paddingBottom: 64px;
  justifyContent : flex-end;
`

```

```

const LoginText = styled.Text`
  fontSize : 20px;
  marginBottom : 480px;
  marginLeft : 16px;
  fontWeight : bold;
`

```

```

const SignInButton = styled.TouchableOpacity`
  borderRadius : 8px;
  height : 40px;
  justifyContent : center;
  alignItems : center;
  backgroundColor : ${Colors.blue};
`

```

```
`  
const SignInText = styled.Text`  
fontWeight : bold;  
fontSize : 18px;  
color : white;  
`
```

```
export default LoginScreen;
```

2. Home Page

```
import { BabelFileModulesMetadata } from '@babel/core';  
import React, { useState } from 'react';  
import { ActivityIndicator } from 'react-native';  
import { View } from 'react-native';  
import Icons from "react-native-vector-icons/MaterialIcons";  
import styled from 'styled-components/native';  
import Colors from '../constants/colors';  
import { Bus } from '../models/bus.model';  
import searchAvailability from './search.service';
```

```
interface Props {  
  navigation: any  
};
```

```
const HomeScreen = ({ navigation }: Props) => {  
  const [source, setSource] = useState("");  
  // const [destination, setDistination] = useState("");  
  const [nobus, setNoBus] = useState(false);  
  const [buslist, setbuslist] = useState<Array<Bus>>([]);  
  const [isLoading, setLoading] = useState(false);
```

```
  const searchBusRoute = () => {
```

```

setNoBus(false);
setLoading(true);
const buses: Array<any> = searchAvailability(source);
setTimeout(() => {
  setLoading(false);
  if (buses.length == 0) {
    setNoBus(true);
    return;
  }
  setbuslist(buses);
}, 3000)
}

```

```

const getFormattedEta = (dateFuture: any) => {
  const dateNow: any = new Date();
  const seconds: number = Math.floor((dateFuture - (dateNow)) / 1000);
  let minutes = Math.floor(seconds / 60);
  let hours = Math.floor(minutes / 60);
  const days = Math.floor(hours / 24);
  hours = hours - (days * 24);
  minutes = minutes - (days * 24 * 60) - (hours * 60);
  return minutes;
}

```

```

const handleChange = (e: any) => {
  setSource(e);
}

```

```

return (
<>
<HScreen>
<SearchBoxContainer>
<SearchByBusNUmber>
  Enter bus number
</SearchByBusNUmber>
<FromInput
  value={source}
  onChangeText={handleChange}
  placeholder={"Search by bus number"} />

```

```

        /* <TextInput
            value={destination}
            onChange={(e: any) => setDistination(e.target.value)}
            placeholder={"To : Destination location"} /> */
<SearchButton onPress={searchBusRoute} >
<View style={{
    display: "flex",
    flexDirection: "row",
    alignItems: "center"
}} >
<SearchText >Search</SearchText>
<Icons name="search" size={18} color="#FFFFFF" />
</View>
</SearchButton>
<AvailableBuses>
    {buslist.length > 0 ? "Available buses" : ""}
</AvailableBuses>
</SearchBoxContainer>
{
    isloading ? <NoBusContainer>
<ActivityIndicator size={50} color={Colors.blue} />
</NoBusContainer> :

        nobus ? (
<NoBusContainer>
<NoBusText>Sorry, No buses available for the selected bus
number</NoBusText>
</NoBusContainer>
        ) : buslist.length === 0 ? (
<NoBusContainer>
<NoBusText>Search by bus number by check availability</NoBusText>
</NoBusContainer>
        ) : (
<BusListContainer>
            {
                buslist.map((bus, index) => (
<BusTile key={bus.currentStop + index}>
<View style={{
                    display: 'flex',

```

```

                flexDirection: "row",
                alignItems: "center",
                justifyContent: "space-between"
            }} >
<View>
<Row style={{
                alignItems: "center"
            }} >
<BusNumber>
                {bus.number}
</BusNumber>
<Icons name="location-on" size={16} color={Colors.blue} />
<CurrentLocation>
                {bus.currentStop}
</CurrentLocation>
</Row>
<ETA>
                {bus.eta + " Minutes"}
</ETA>
</View>
<Row>
<IconPressable onPress={() => {
                navigation.navigate("Stops", { busstats: bus })
            }} >
<Icons name="navigate-next" size={24} color={Colors.blue} />
</IconPressable>
<IconPressable >
<Icons name="notifications-none" size={24} color={Colors.blue} />
</IconPressable>
</Row>
</View>
</BusTile>
        ))
    }
</BusListContainer>
)
}

</HScreen >

```

```
</>  
);  
};
```

```
const SearchBoxContainer = styled.View`  
  paddingHorizontal : 24px;  
`
```

```
const HScreen = styled.View`  
  flex : 1  
  backgroundColor : #FFFFFF;  
`
```

```
const FromInput = styled.TextInput`  
  marginBottom : 12px;  
  height : 48px;  
  fontSize : 18px;  
  fontWeight : bold;  
  paddingHorizontal : 8px;  
  borderColor: transparent;  
  backgroundColor : ${Colors.smoke};  
`
```

```
const SearchByBusNUmber = styled.Text`  
  fontSize : 20px;  
  fontWeight : bold;  
  lineHeight : 32px;  
  marginVertical : 8px;  
`
```

```
// const ToInput = styled.TextInput`  
// marginVertical : 8px;  
// height : 48px;  
// fontSize : 18px;  
// paddingHorizontal : 8px;  
// fontWeight : bold;  
// borderColor: transparent;  
// backgroundColor : ${Colors.smoke};  
// `
```

```
const SearchButton = styled.TouchableOpacity`
marginVertical : 12px;
borderRadius : 8px;
height : 40px;
justifyContent : center;
alignItems : center;
backgroundColor : ${Colors.blue};
`
```

```
const SearchText = styled.Text`
fontWeight : bold;
fontSize : 18px;
color : #FFFFFFF;
marginRight : 8px;
`
```

```
const AvailableBuses = styled.Text`
fontSize : 18px;
fontWeight : 700;
paddingVertical : 8px;
`
```

```
const BusListContainer = styled.View`
flex : 1;
paddingTop : 8px;
backgroundColor : ${Colors.smoke};
paddingHorizontal : 24px;
`
```

```
const BusTile = styled.View`
height : 80px;
backgroundColor : #FFFFFFF;
borderRadius : 8px;
marginVertical : 8px;
paddingVertical : 16px;
paddingHorizontal : 16px;
`
```

```
const BusNumber = styled.Text`
fontSize : 20px;
paddingRight: 8px;
fontWeight : bold;
`
```

```
const CurrenLocation = styled.Text`
fontSize : 16px;
fontWeight : bold;
`
```

```
const ETA = styled.Text`
fontSize : 16px;
fontWeight : bold;
opacity : 0.6;
`
```

```
const Row = styled.View`
display : flex;
flexDirection : row;
`
```

```
const IconPressable = styled.Pressable`
paddingHorizontal : 4px;
`
```

```
const NoBusContainer = styled.View`
flex:1;
display:flex;
flexDirection:row;
justifyContent:center;
alignItems:center;
paddingHorizontal : 32px;
`
```

```
const NoBusText = styled.Text`
fontSize : 20px;
fontWeight : bold;
`
```

```
export default HomeScreen;
```

3. Stops Page

```
import React from 'react';
import { Text, View, FlatList } from 'react-native';
import Icons from "react-native-vector-icons/MaterialIcons";
import styled from 'styled-components/native';
import Colors from '../constants/colors';

const StopsScreen = ({ route }) => {

  const { source, destination, number, stops } = route.params.busstats;
  // console.log(route.params);

  return <SScreen>
    <TopContainer>
      <BusNumer>
        {number}
      </BusNumer>
      <Row>
        <SrcDestNames>
          {source}
        </SrcDestNames>
        <Icons name="arrow-right-alt" size={36} color="#FFFFFF" />
        <SrcDestNames>
          {destination}
        </SrcDestNames>
      </Row>
    </TopContainer>
    <BottomContainer>
      <ButStop>
        Bus stops
      </ButStop>
      <FlatList
        data={stops}
        renderItem={({ item }) => <StopsContainer key={item.name}
crossed={item.crossed} >
```

```

        {item.crossed ? <Icons size={18} name="radio-button-checked"
color={Colors.green} />
        : <Icons size={18} name="radio-button-unchecked"
color={Colors.green} />}
<Time>{item.crossed ? "Crossed" : item.eta}</Time>
<StopName>{item.name}</StopName>
</StopsContainer>
    />
</BottomContainer>
</SSScreen>
}

```

```

const SSScreen = styled.View`
flex : 1;
backgroundColor : ${Colors.blue};
`

```

```

const TopContainer = styled.View`
height : 150px;
display : flex;
flexDirection : column;
alignItems : center;
justifyContent: center;
`

```

```

const BusNumer = styled.Text`
fontSize : 24px;
fontWeight : bold;
color : #FFFFFF;
paddingBottom : 16px;
`

```

```

const Row = styled.View`
display : flex;
flexDirection : row;
`

```

```

const SrcDestNames = styled.Text`
paddingHorizontal : 8px;
fontSize : 24px;
`

```

```
fontWeight : bold;  
color : #FFFFFF;  
`
```

```
const BottomContainer = styled.View`  
flex : 1;  
flexDirection : column;  
alignItems : center;  
backgroundColor : #FFFFFF;  
borderTopLeftRadius : 50px;  
borderTopRightRadius : 50px;  
paddingHorizontal : 24px;  
paddingTop : 8px;  
`
```

```
const ButStop = styled.Text`  
fontSize : 18px;  
fontWeight : bold;  
textAlign : center;  
color : #000000;  
marginBottom : 8px;  
`
```

```
const StopsContainer = styled.View`  
display : flex;  
flexDirection : row;  
paddingVertical : 16px;  
opacity : ${props => props.crossed ? 0.4 : 1}  
`
```

```
const StopName = styled.Text`  
fontSize : 18px;  
paddingHorizontal : 16px;  
`
```

```
const Time = styled.Text`  
fontSize : 18px;  
paddingHorizontal : 16px;  
`
```

```
export default StopsScreen;
```