

PUBLIC BUS MONITORING SYSTEM

Harsh Kumar¹, Sourav Patra²

¹ Student, Computer Science and Engineering, SRM Ramapuram, Tamil Nadu, India

² Student, Computer Science and Engineering, SRM Ramapuram, Tamil Nadu, India

ABSTRACT

With the endless expansion of metropolitan cities, one of the major concerns of people is the means of transportation. All the major cities are spread across tens of kilometers. In such cases, going from one place to another is a hectic process. The government does a great job by providing people with options like local trains, metro trains and government buses. As the local people get used to these systems, the people who come from other cities have a hard time keeping a track of these services especially when there is a language gap. For example, the MTC services in Chennai, India, started with a fleet of 30 buses in 1947. Today the number of buses has increased to 3964, running at the same time all over Chennai and nearby areas. These buses have a distinct number, and the buses having a particular number runs to and fro, on a particular route. So, if a commuter has to go someplace, he/she has to know the nearest bus stand as well as the bus numbers which ply on the required route. Though the buses ply at regular time intervals from the initial bus stops, the time they reach to the final and intermediate bus stops is affected by factors like traffic, road conditions, weather etc. So, people never know how long it will take for the next bus to come and they end up wasting time for the wrong bus. This project aims at developing an interactive notification system about the location of public buses. The proposed system will allow people to know about the exact location of the buses by the help of GPS on mobile devices and Google Maps API. It also provides additional information like the route of the buses and bus stands it will stop at. The location is retrieved using a small android app using inbuilt android APIs and sent to an HTTP server where it is stored in a database. On the front end, the locations are fetched from the server and displayed on the map.

Vehicle Tracking: - Bus, GPS, Google Maps, location tracking

1. PROBLEM STATEMENT

With the endless expansion of metropolitan cities, one of the major concerns of people is the means of transportation. All the major cities are spread across tens of kilometers. In such cases, going from one place to another is a hectic process. The government does a great job by providing people with options like local trains, metro trains and government buses. As the local people get used to these systems, the people who come from other cities have a hard time keeping a track of these services especially when there is a language gap. For example, the MTC^[1] services in Chennai, India, started with a fleet of 30 buses in 1947. Today the number of buses has increased to 3964, running at the same time all over Chennai and nearby areas. These buses have a distinct number, and the buses having a particular number runs to and fro, on a particular route. So, if a commuter has to go someplace, he/she has to know the nearest bus stand as well as the bus numbers which ply on the required route. Though the buses ply at regular time intervals from the initial bus stops, the time they reach to the final and intermediate bus stops is affected by factors like traffic, road conditions, weather etc. So, people never know how long it will take for the next bus to come and they end up wasting time for the wrong bus.

2. OBJECTIVE

The project aims at creating a system for tracking and information retrieval about public buses in transit. The system will consist of an user interface which will show the location of buses in real time and their routes, A module in the

buses which will retrieve the geographical coordinates and send it to the server and a database on the server which will be used to store and fetch bus details in real time.

3. SPECIFICATION

The 'Public Bus Monitoring System' provides helpful information about Public Buses in a given geographical area. Also it stores tracking information in the database which keeps track of the Bus's location and informs the commuters about the same. To implement this, seamless connection between the buses and the front end is necessary. The users will not only be able to see the nearby buses, but also see their routes and destination. This will help people especially those who are new to the city. This will save a lot of time for people who can't afford their own vehicle or are visiting the city from brief time.

3.1 Purpose

Even though Public Bus Transport System is the most important form of urban and rural passenger transport in India we are not utilizing the advancement in technology in this area. There is no current system available to track the movement of public bus. The time of arrival and departure totally depends on various conditions and there is no way to notify the commuters about it. In the daily operation of bus transport systems, mainly that of buses, the movement of vehicles is affected by different uncertain conditions as the day progresses. It could be due to traffic congestion, unexpected delays or irregular vehicle-dispatching times. As a result, commuters are left stranded on road waiting for directions to reach their destination. Many commuters are late to work because they decide to wait for the bus instead of just simply using an alternative transport. Moreover, the issues with the above systems are quite a few. The main goal of the proposed work is to improve the Bus Transit system by adding the necessary additional features into the application, like accurate bus timings, correct bus numbers and moreover adding a GPS enabled device into the bus to track its movement. The location is then updated in the database and is shown onto the Google Map displayed in the user's device. This allows real-time location of the bus being shown to the user along with the estimated arrival time of their bus. Commuters can take advantage of these information and decide according on their mode of travel whether to take a bus or look for an alternative. They need not be stranded on road thinking about the same.

3.2 Project scope

Using GPS tracking^[2] will not only improve the efficiency of city bus operations, but also encourage more commuters to take advantage of their city's bus systems. People generally get frustrated waiting for a particular bus or are unable to determine which bus goes to which area. This will make it significantly easier for people to use the right bus and will also be well informed about its arrival.

3.3 User classes and characteristics

The general users of the system are regular commuters who need to travel around the city a lot. The system will help them travel faster without wasting time waiting for the buses. The application will help users to know the public transport system better. The system can also be used to keep a track of the buses and analyze their movement.

3.4 Operating environment

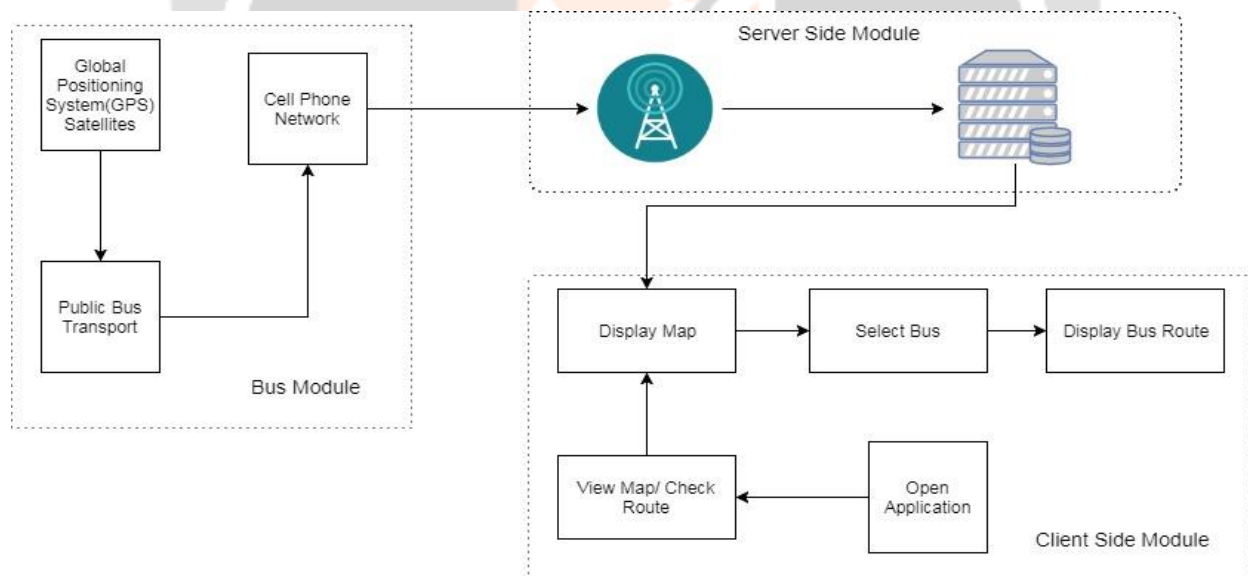
To determine the location of buses, the android app needs to run on a device on the bus. The app is developed using android studio, an integrated development environment developed by google to create android apps. It uses an API called location services API^[3] which requires a minimum of android 4.0 installed on the device and an integrated GPS sensor. The device also needs to have an active internet connection to communicate with the server. The front end consists of a Google map integrated web page which will display the real time locations of the buses and their route information. This webpage runs on any browser which supports JavaScript. The server contains files which will help in extracting and inserting bus locations into the server. These file needs to be hosted on a remote server. The database is a generic MySQL database running on the server and the database adapters are written in PHP. So, a suitable file hosting server which supports MySQL and PHP is chosen.

3.5 Design and implementation constraints

The system uses a phone which runs on android to track the vehicles. Only an android phone can be used for retrieving the location. It needs to run an android version 4.0 or higher. It needs to have a consistent internet connection and a power source to avoid shutdown due to power loss. On the other hand, the front end runs on a web browser which can support JavaScript. An active internet connection is necessary to communicate with the server. To render the map in the web browser, google maps API is used hence, an API key is needed to be generated to use this API. The server is hosted on a third party server. A proper server space needs to be employed, which supports MySQL server and PHP to feed and extract data from the server.

4. SYSTEM DESIGN

The proposed system involves several elements, such as the android device that transmits the location of the buses, a database to store the transmitted data in a relational way, and the user interface that displays the location of the buses to the user. The android device calculates the latitude and longitude of its position and transmits it via cellular network. The database is hosted on web that stores the location information. The user interface consists of a map that is embedded using the Google Maps API, which is a messenger that requests and tells the system what to do and returns the response back. It also displays the geolocation of the user as well as the location of the buses. The buses are displayed on the map using map markers. Each marker is put using the latitude and longitude of the respective bus. On a single click on the bus marker the route of the bus is displayed using the Google Direction Service and distances are measured using the Distance Matrix Service. The location of the buses is updated every 5 seconds. This allows the commuters to view the real-time location of the buses all the time.



5. MODULES

The proposed system is a complex project where a lot of different things happen at different places. So, to make things simple, the project is divided in modules so the processing can managed on a smaller scale. The four modules are:

- **Bus Module** This module is responsible for determining the location of the buses and sending it to the server. This module is the most important module for any live tracking system.

- **User Interface** This module is responsible for creating a user friendly console for commuters which will provide the details about the buses to the user. This module is the face of the project to which the users will interact.
- **Client Server Communication** This module acts as the middle man between other modules allowing the modules to communicate each other through it. It also acts as the remote location where the data is stored and easily accessible for other modules.
- **Route Info** This module provides the route information of the buses according to their bus number

5.1 Bus module

To track buses in real time, their exact location needs to be determined. To do this a module needs to be developed, which is used to determine the geographical coordinates. This module is also needed to be connected to the internet to send data from the bus to the server. The best way to do this is to use mobile devices as almost all mobile phones today use global positioning system and are also connected to the internet using cellular networks. To get the location, Location Services API of Android repository is used. Android gives applications access to the location services supported by the device through classes in the `android.location` package. The central component of the location framework is the [LocationManager](#) system service, which provides APIs to determine location and bearing of the underlying device. First an instance of `LocationManager` is requested from the system by calling `getSystemService (context.LOCATION_SERVICE)`. This function returns a handle to a new `LocationManager` instance. Now the application will be able to register/unregister for periodic updates of the bus's current location from a location provider. A location provider is a medium by which the current location is determined. The location can be extracted from the gps sensor as well as it can also be requested from the network provider which is way more accurate than gps. For making this work, the application needs to ask `ACCESS_FINE_LOCATION` permission from the user. Whenever, location of buses change, the new coordinates need to be updated on the server. The data is sent to the server using cellular network. Volley is an HTTP library that makes networking for android app easier and faster. Its source code is available on GitHub. Volley offers many features like automatic scheduling of network requests, multiple concurrent network connections. At high level, volley is used by creating a `RequestQueue` passing it `Request` objects. The `RequestQueue` is a first in first out queue which manages worker threads for running the network operations, reading from and writing to the caches, and parsing responses. A `Request` object contains network requests to be processed on a particular url. In this context, the bus number and its coordinates are sent to the server as a request, which is processed by the database adapter to store in the database. To use Volley, `android.permission.INTERNET` permission needs to be added to the app's manifest. The target device may be the driver's phone or the ticket collector's phone or a standalone device kept in the bus somewhere. The target device needs to be running on Android 4.0 or higher as location services API is compatible with android 4.0 and higher. The device also needs to be connected to internet to communicate with the server.

5.2 User Interface

The user interface of the system consists of a map which shows the location of all the buses in transit. The map is rendered using Google Maps API. Google Maps is a web mapping service developed by Google. It offers satellite imagery, street maps, 360° panoramic views of streets (Street View), real-time traffic conditions (Google Traffic), and route planning for traveling by foot, car, bicycle (in beta), or public transportation. Google maps is the most widely used map service in the world. It's the first app which comes in mind when someone thinks of navigation. It is free to use for developers to make systems related to navigation. Most common of these are online cab booking services like Uber and Ola, which provides the locations of nearby cabs on a map. This map is integrated by using Google Maps API. Google launched the Google Maps API in June 2005 to allow developers to integrate Google Maps into their websites. It is a free service, and currently does not contain ads, but Google states in their terms of use that they reserve the right to display ads in the future. By using the Google Maps API, it is possible to embed Google Maps site into an external website, on to which site specific data can be overlaid. The Google Maps API is free for commercial use, provided that the site on which it is being used is publicly accessible and does not charge for access, and is not generating more than 25,000 map accesses a day. Implementing maps in a website is very easy. Google maps does an authentication using an API key. Before rendering the map, the API checks for a valid

API key which can be obtained from the official website. This key is used to monitor the number of API calls by a particular user. When the map is successfully rendered on the page, the location of buses needs to be displayed on it. The location of the buses are stored in a database on the server. The official guides suggest to extract the coordinates from an XML document to make it fast and easily manageable. The location are extracted from the server using AJAX and setTimeout functions of JavaScript. So, every few seconds the data is extracted from the database and the XML page is requested. The response is then parsed to extract the coordinates. These coordinates are then used to see the exact location of the buses. It also includes the list of all the buses available nearby having a filter of distance between the current location of the user and the nearby buses. The user is able to select any bus and see its location as well its route to better plan his/her journey. The user interface is a web app which runs on any web browser. It requires a device which is connected to the internet and need to have a gps sensor so that the current location can be determined. The webpage is made using simple HTML and CSS. JavaScript is used to render the map and to extract the location information.

5.3 Server

The server acts as a medium between the user interface and the bus module. The bus module provides the location of the buses and the user interface displays it on the map. The server and database act as a middle man between them. It receives the data from the bus module and stores in the database for the user interface to fetch to display to the user. Instead of communicating directly with each other, the other two modules interact with the server. This helps in managing the bandwidth and the user interface becomes smoother. A public third party server space provider is used. In this case Awardspace.com is chosen. It provides limited free server and database space of up to one giga bytes. It supports php as well as MySQL. It provides full access to the database through personel 'phpmyadmin' page for managing the database. It consists of a database adapter which receives the data from bus module and stores it in the database. It is php page which receives the information as a post request and inserts it in the database. PHP is a server-side scripting language designed primarily for web development but also used as a general-purpose programming language. Originally created by Rasmus Lerdorf^[4] in 1994, the PHP reference implementation is now produced by The PHP Development Team. PHP originally stood for Personal Home Page, but it now stands for the recursive acronym PHP Hypertext Preprocessor. PHP code may be embedded into HTML or HTML5 markup, or it can be used in combination with various web template systems, web content management systems and web frameworks. PHP code is usually processed by a PHP interpreter implemented as a module in the web server or as a Common Gateway Interface (CGI) executable. The web server software combines the results of the interpreted and executed PHP code, which may be any type of data, including images, with the generated web page. It is connected to the database using function mysqli_connect(). This method returns a connection object which is then used to connect to the database. The database is a simple MySQL database running on the server. MySQL is an open-source relational database management system (RDBMS). When the adapter page receives the data, it sends the data to database in respective rows. The server also includes a page which generates location of the buses as an XML document. For the user interface. Whenever this page is requested, it creates an XML document in response. The response of this page is parsed for XML elements to fetch the locations. This page creates a new node for every bus in the database and stores their locations inside these nodes. The speed of the server depends on many factors like number of requests per minute, The size of the database. Hence a proper server space is required for the deployment of the system. Before the full scale deployment, the server can be transferred to a bigger space or it can be moved to the cloud.

5.4 Route info

The most important part of the system is to provide information about the buses especially their routes. The implementation of routes is the most challenging part of the system as there is no reliable source of this information. The buses work in a traditional way, based on drivers experience and conventional routes. There isn't a proper data source other than details of the bus number and their respective bus stops where it goes which is available for the general public to see on MTC's official website. So, the data of the buses are extracted from the website and stored in a separate table which will contain the first and last bus stops of each bus. Google maps API contains a sub repository called Directions API which allows developers to plot a directions map between two points on the map. It contains a feature of changing the mode of transportation which also contains public transport as an option i.e. the path a public bus would take to go from one point to the other. When a bus is selected, according to the bus number, the first and last bus stop's location is retrieved from the database. These two points are used to plot a directions map from the first stop to the last. All the intermediate bus stops are also highlighted so the commuters will be able

to know on which stops a particular bus will go. The route of the buses is updated seamlessly every time a different bus is selected.

6. SYSTEM IMPLEMENTATION

The platform for the system is an android device which sends the location of the buses and the user interface which displays the locations on the map. The android device used runs on android 7.0. It has an inbuilt GPS sensor and a sim slot. This sim slot carries a sim card. This connection has an active internet connection. The app is installed on the system and the phone is placed on the bus with the app running. On the server, the server is made active so that it receive and send data to and from the bus module and user interface respectively. The user computer is turned on and the chrome is opened. The user computer is a laptop running windows operating system and has chrome browser installed. The address of the server is entered in the address bar. The browser loads the index page which contains the map and the locations of the buses.

6.1 Overview of the platform

The android device used as the bus module is a Moto G5 plus running on android 7.0 Nougat operating system having 4 gigabytes of RAM and 16 gigabytes of internal storage. The app is installed on it through a USB cable in ADB mode from android studio. The app installed is developed on Android studio 2.3 which is available for free on the google developer's website. The android studio is installed on a HP laptop with 8 gigabytes of RAM and 1 TB hard disk running on windows 10 operating system. The server is created on 'www.awardspace.com', a free website hosting website which provides limited amount server space and a small database for free. The user interface was tested on the same computer to test the system.

6.2 Simulation parameters

The android device used runs on android 7.0. It has an inbuilt GPS sensor and a sim slot. This sim slot carries a sim card. This connection has an active internet connection. The app is installed on the system and the phone is placed on the bus with the app running. On the server, the server is made active so that it receive and send data to and from the bus module and user interface respectively. The user computer is turned on and the chrome is opened. The user computer is a laptop running windows operating system and has chrome browser installed. The address of the server is entered in the address bar. The browser loads the index page which contains the map and the locations of the buses. The android device used as the bus module is a Moto G5 plus running on android 7.0 Nougat operating system having 4 gigabytes of RAM and 16 gigabytes of internal storage. The app is installed on it through a USB cable in ADB mode from android studio. The app installed is developed on Android studio 2.3 which is available for free on the google developer's website. The android studio is installed on a HP laptop with 8 gigabytes of RAM and 1 TB hard disk running on windows 10 operating system. The server is created on 'www.awardspace.com', a free website hosting website which provides limited amount server space and a small database for free. The user interface was tested on the same computer to test the system.

7.1 CONCLUSION

Hence, the proposed system gives a reliable notification service for public buses which will reduce the time wasted by people standing on the bus stop, waiting for buses. This time can be utilized in doing some other productive work. The tools used are free to use and available for the user, hence, the system can be deployed free of cost. The system can also be used to keep a track of the buses and determining problems associated with the current system to improve user experiences. The system is easy to implement and doesn't require any special, new hardware to be installed. It can be readily deployed without any extra cost. The location determination requires an android device which is abundantly available. The app developed is small and can run in the background without affecting the device's performance. The server space used is provided freely on a limited amount by the provider, 'www.awardspace.com' which supports PHP and MySQL required by the project. The user interface is a simple and user friendly which makes it easily understandable by the regular user.

6. REFERENCES

- [1]. The Metropolitan Transport Corporation (formerly known as Pallavan Transport Corporation) sometimes known as the MTC, is the agency that operates the public bus service in Chennai, India.
- [2]. The Global Positioning System (GPS), originally Navstar GPS, is a space-based radio navigation system owned by the United States government and operated by the United States Air Force
- [3]. In computer programming, an application programming interface (API) is a set of subroutine definitions, protocols, and tools for building application software.
- [4]. Rasmus Lerdorf (born 22 November 1968) is a Danish-Canadian[1] programmer. He created the PHP scripting language.

