1. Introduction

One of the most relevant issues in cities is mobility. The current situation of traffic congestion in cities leads to several health problems derived from pollution, noise, and, obviously, the great stress in drivers who spend several hours in traffic to reach their destination [1]. Several developed cities have created smart mobility systems as new solutions to solve this problem. The implementation of systems of this kind involves the development of software and hardware infrastructure in the city. These developed countries have created passenger information systems for bus users that indicate the real-time bus location, and they have also developed several apps that provide this information directly to the smartphones of the bus users.

However, in many provinces or developing cities, there is a lack of solid human and physical infrastructure to implement intelligent transport projects. The main problem with the public transport system in Mexico is the lack of appropriate mechanisms to manage the large number of means of transportation, such as buses, metrobuses, or metro [2,3]. In the Mexican context, there are not enough public buses to cover the enormous demand for trips around the city every day. In addition, the buses are in a very bad condition. Therefore, it is very difficult to assume that buses have the needed infrastructure to publish their real-time location to their users. In Mexico, there are only a few examples of intelligent transportation systems that help users' mobility. In this scenario, intelligent mobility solutions must take into account the context of cities with limited infrastructure that have computational resources of the users that need to provide effective solutions.

Solutions are needed that use the common resources of passengers and bus drivers, such as smartphones. It is possible to take advantage of the several sensors of recent smartphones to determine location and manage remote communication with web servers. Moreover, cheap technological solutions are required to determine the available seats on each public bus.

2. Background and Related Work

2.1. Smart Cities and the Internet of Things

A smart city is an urban development that integrates Information and Communication Technologies and the Internet of Things (IoT) technology in a secure fashion to manage a city's assets. The smart city can be defined as a city that uses information and communication technologies to improve the critical infrastructure of the city and also to make the public services for citizens more efficient [**4**].

A Smart City uses technology to improve the quality of life and the accessibility of its inhabitants. Additionally, it ensures sustainable economic, social, and environmental development. A smart city allows citizens to interact with it in a multidisciplinary way and also permits the city to adapt to their needs in real time [**5**]. The concept of a Smart City requires the implementation of mechanisms to know the real time context of the services provided to the citizens. For example, in the case of a smart mobility system, such as the one proposed in this research work, the users need to know the location of buses in real time in order to improve the times used to reach a destination.

One of the technologies that has driven the development of smart cities is the Internet of Things. There are different definitions for the concept of the Internet of Things, however, one of the most accepted definitions is the one provided by the European Research Cluster on the Internet of Things (IEC). It defined the Internet of Things as a dynamic global network infrastructure with self-configuring capabilities based on standard and interoperable communication protocols where physical and virtual "things" have identities, physical attributes, and virtual personalities, use intelligent interfaces, are seamlessly integrated into the information network, and often communicate data associated with users and their environments. This enables objects to detect their context and to communicate with others in order to interact with the context.

3. The Proposed Smart Information System

Today, the public transport system in Mexico does not have the functionality to provide information to determine the arrival times of buses or to know if there are available seats on the next bus that is arriving to the user's bus stop. This information is very relevant in Mexican cities where millions of users a day must be mobilized. The proposed solution is based on the fact that the current public buses do not have any kind of sensor or mechanism to determine the location of the bus.

The approach uses the resources of the bus drivers' smartphones to collect context information about public transport. This information is used by the algorithms to determine the arrival time to the next bus stop. In addition, a small electronic device was developed to record the users getting on/off of the bus. The users of public transport can use the web application developed in this project to receive the information about the arrival time of the next bus coming to the bus stop where the user is located and also to know the number of available seats on the coming bus. **Figure 1** shows the conceptual architecture of the proposed system that predicts the arrival times of buses and also determines the available seats on that bus.

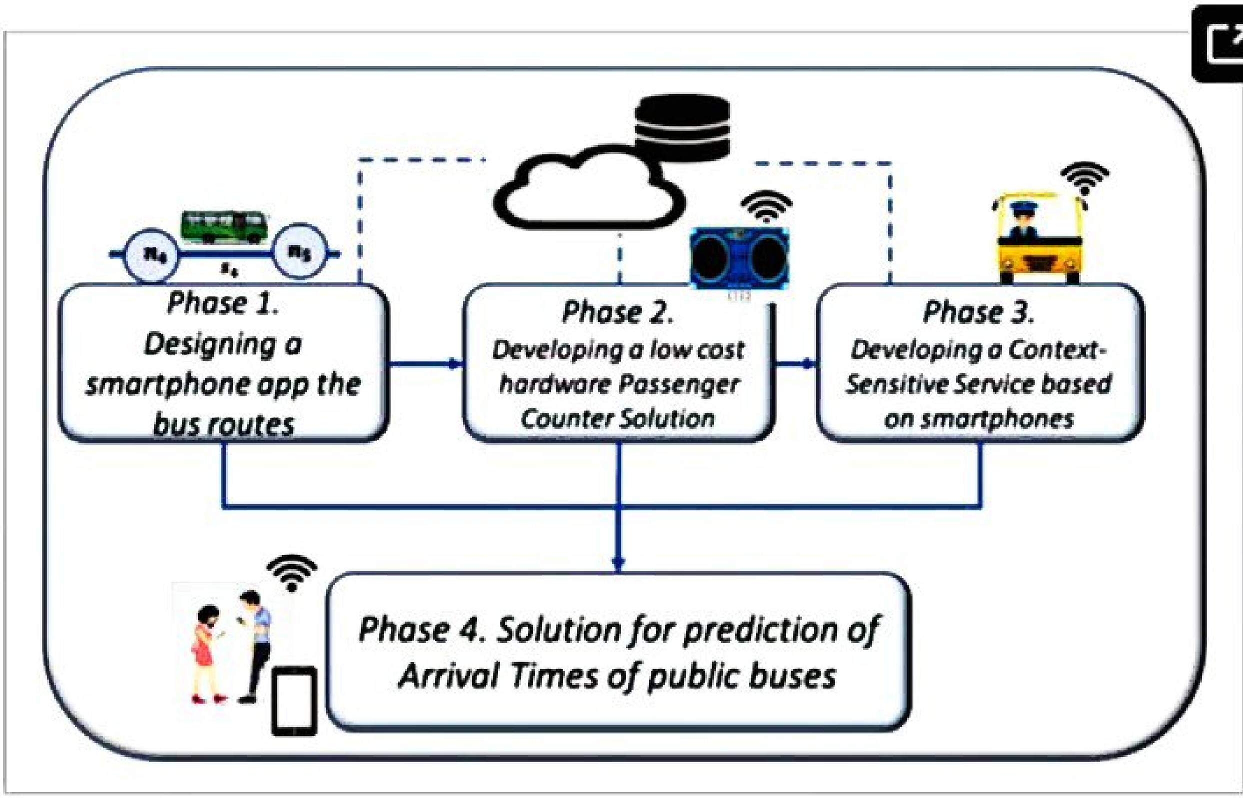


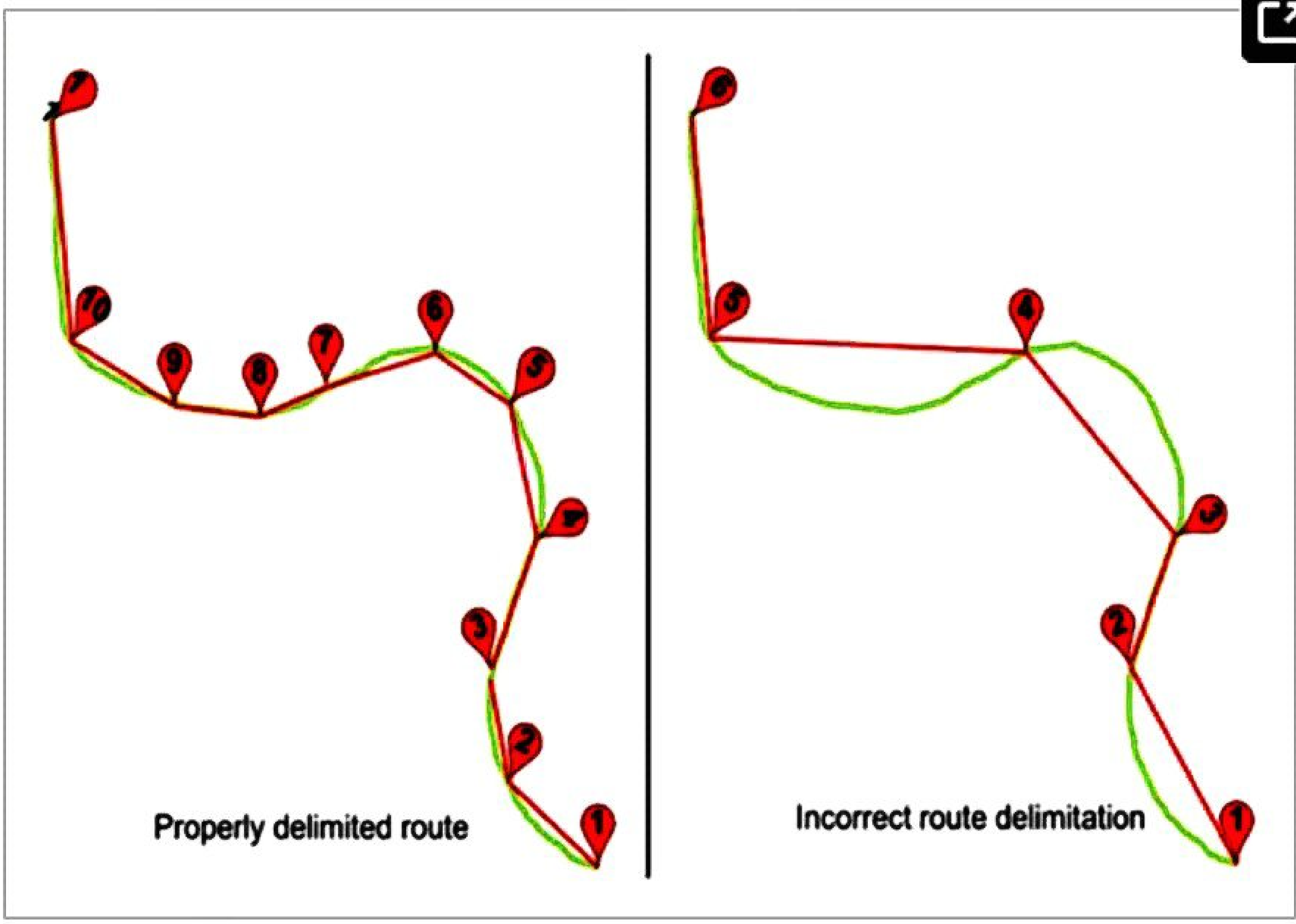
Figure 2. An example of an asymmetrical bus route.

Therefore, to design a specific route, two routes must be created with their corresponding bus stops for each route direction. The routes have been designed as directed graphs, where nodes are used to represent bus stops and edges are used to represent the distances between one bus stop and another. This module uses the API of Google maps and extends it for the definition of each bus stop for a specific route in the city.

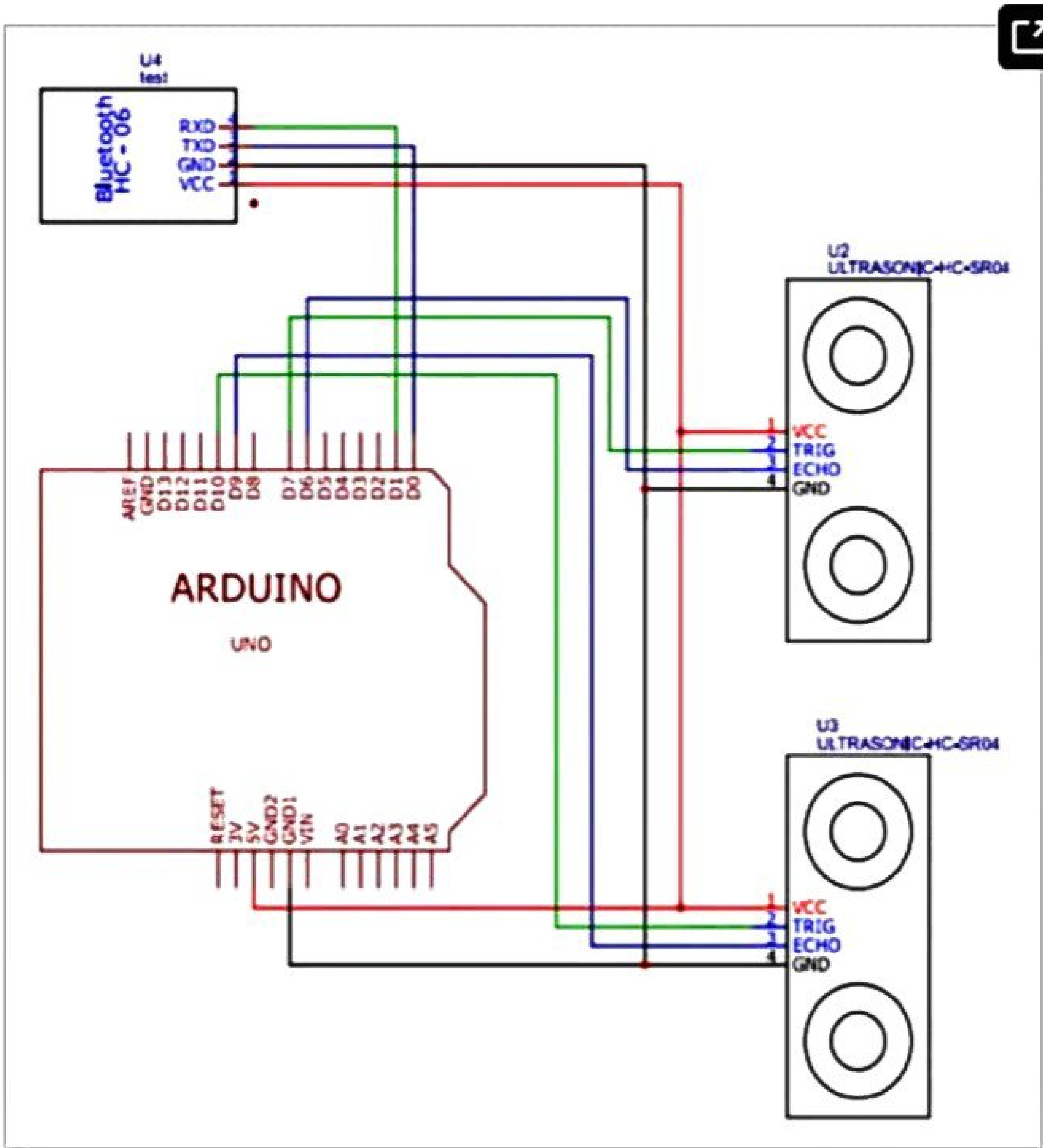
In the first step, the GPS position of each one of the bus stops need to be captured. This is done as follows:

- The proposed application is used to show the map of the Google Maps app on a Smartphone. The application makes use of the “Support-MapFragment” class. The geographical location where the users want to design a route is indicated on the map presented in the smartphone application. The specific method “moveCamera” is used to trace the path of the route.

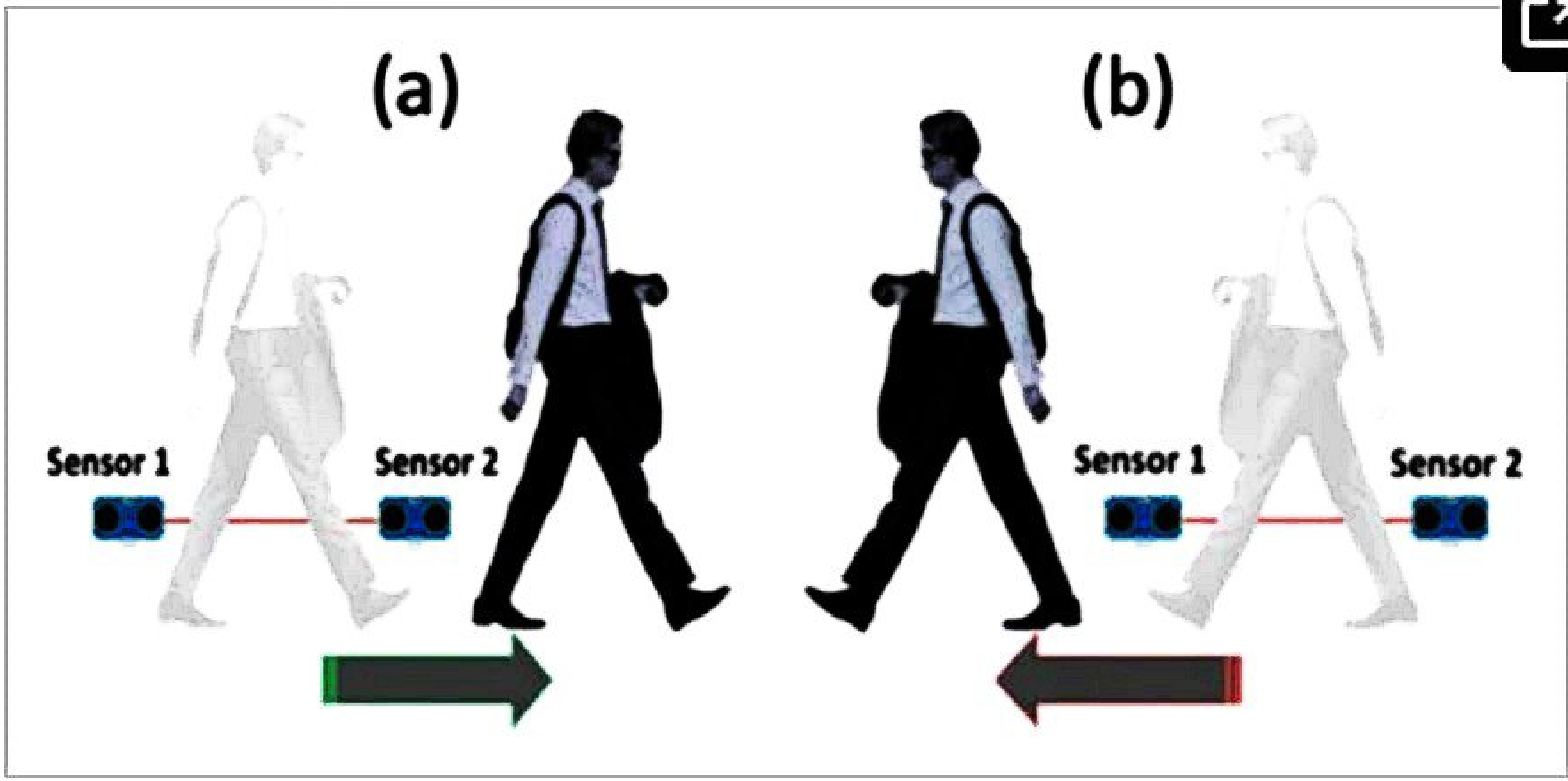
The accuracy of the solution depends on the design of a well-defined route. In most cases, a route in the city is not completely straight and it is likely to have several curves. Therefore, care needs to be taken to define the straight and curved sections of the route. To do this, when designing the route, additional GPS points need to be assigned on the route in order to make a better match between the graphical route designed in the software app and the real bus route in the city. **Figure 3** shows an example of a well-designed route and an example of a poorly-designed route.



The design of the ultrasonic counter was made taking into account the characteristics of commercial products on the market to register the number of people on a bus. Our passenger counter is composed of the following elements: (i) two ultrasonic sensors, which are responsible for detecting a passenger boarding or getting off of the bus; (ii) a Bluetooth module which is responsible for the communication with the smartphone of the bus driver; (iii) and an Arduino-based device which is responsible for processing the data coming from the ultrasonic device. It is also responsible for sending the "getting on/off" events to the smartphone of the bus driver through Bluetooth. The ultrasonic sensors were aligned side by side with a separation of 15 cm. The sensors were configured to measure a maximum distance of 80 cm, which is equivalent to the width of the door of the bus. All of the information captured from the sensors is processed using an Arduino device. **Figure 5** presents the circuit diagram of the ultrasonic counter device.



The USoniCont device follows the following process detailed to register when a person boards a bus (**Figure 6**): if a passenger walks near Sensor 1, a waiting circle is created where a counter is waiting for that user to walk near Sensor 2.



The context information is the following: the GPS position of the monitoring bus, the hour and date when data were sent, the ID of the monitored bus, the ID of the bus driver, etc. This context information is obtained from the smartphone of the bus driver as well as from the ultrasonic circuit counting passengers. **Figure 7** shows the context information that is produced by a change in the location of the bus.

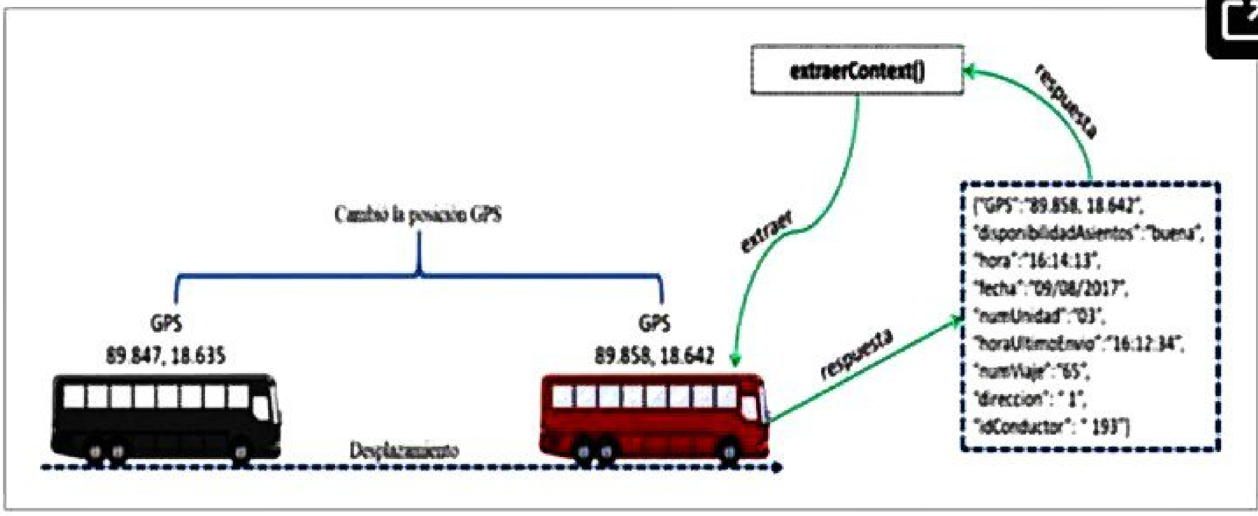


Figure 7. Context data produced by a bus on the route.

(d) Sending the context information to the cloud. The data that is periodically collected from the location sensor of the driver smartphone is used to determine the current/previous bus location. The context information that is produced by the smartphone of the driver and by the passenger counter needs to be sent to a Web Server. To do this, a web service has been created that is responsible for periodically sending the context information to a Web server using the JSON format. The original data obtained from sensors are transformed to a JSON object. This JSON object is continuously sent over the Internet (using the Smartphone resources) to the server. The context data are sent every 20 s. The experimentation demonstrates that a shorter time can generate duplicated data (e.g., the GPS position) and a longer time can result in a lack of data of the bus position, making it difficult to estimate the arrival time.

- ManageDrivers: This module is responsible for registering, modifying, deleting and showing the information of the bus drivers. This module enables the concessionaires to establish the passenger capacity of the transport units. The interface of the system enables the definition of specific drivers for a route and the definition of the capacity of the public buses that compose the designed route.

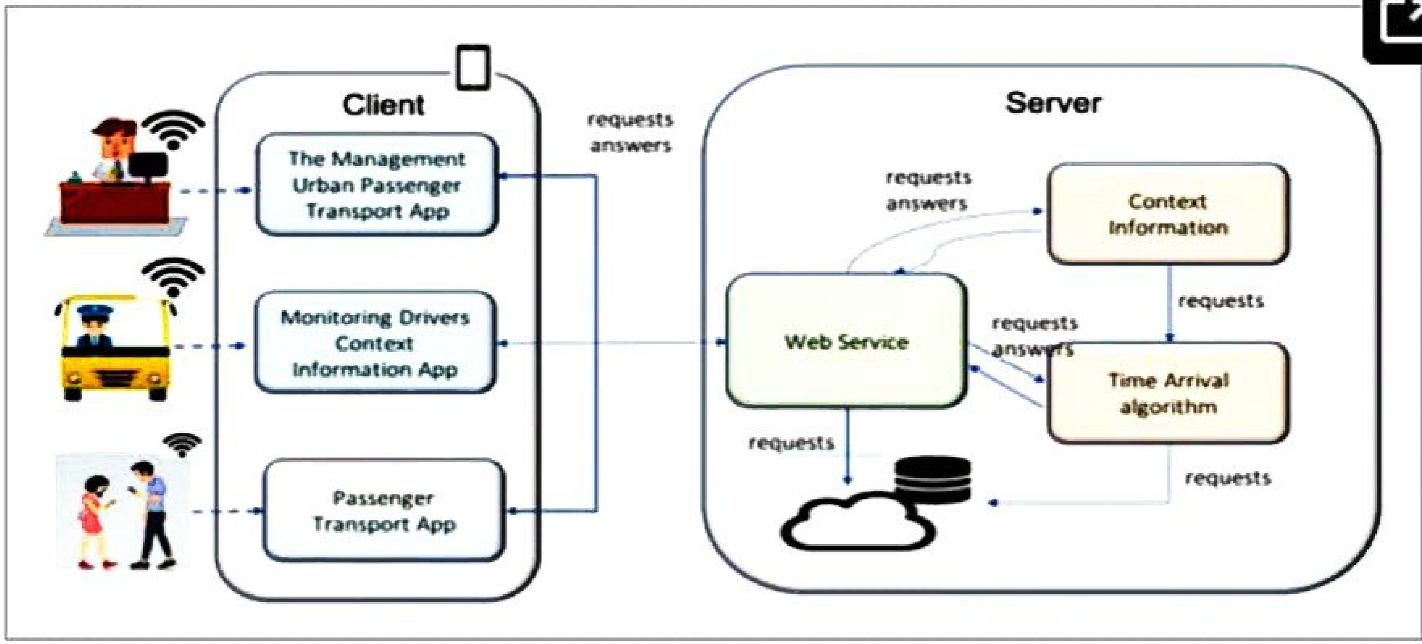
Here's an example of a Python program for public transport optimization:

```python
# Import necessary libraries
import pandas as pd

# Read the public transport data
public_transport_data = pd.read_csv('public_transport_data.csv')

# Perform optimization calculations and logic here

# Print the optimized solution
print("Optimized public transport routes:")
print(optimized_routes)
```

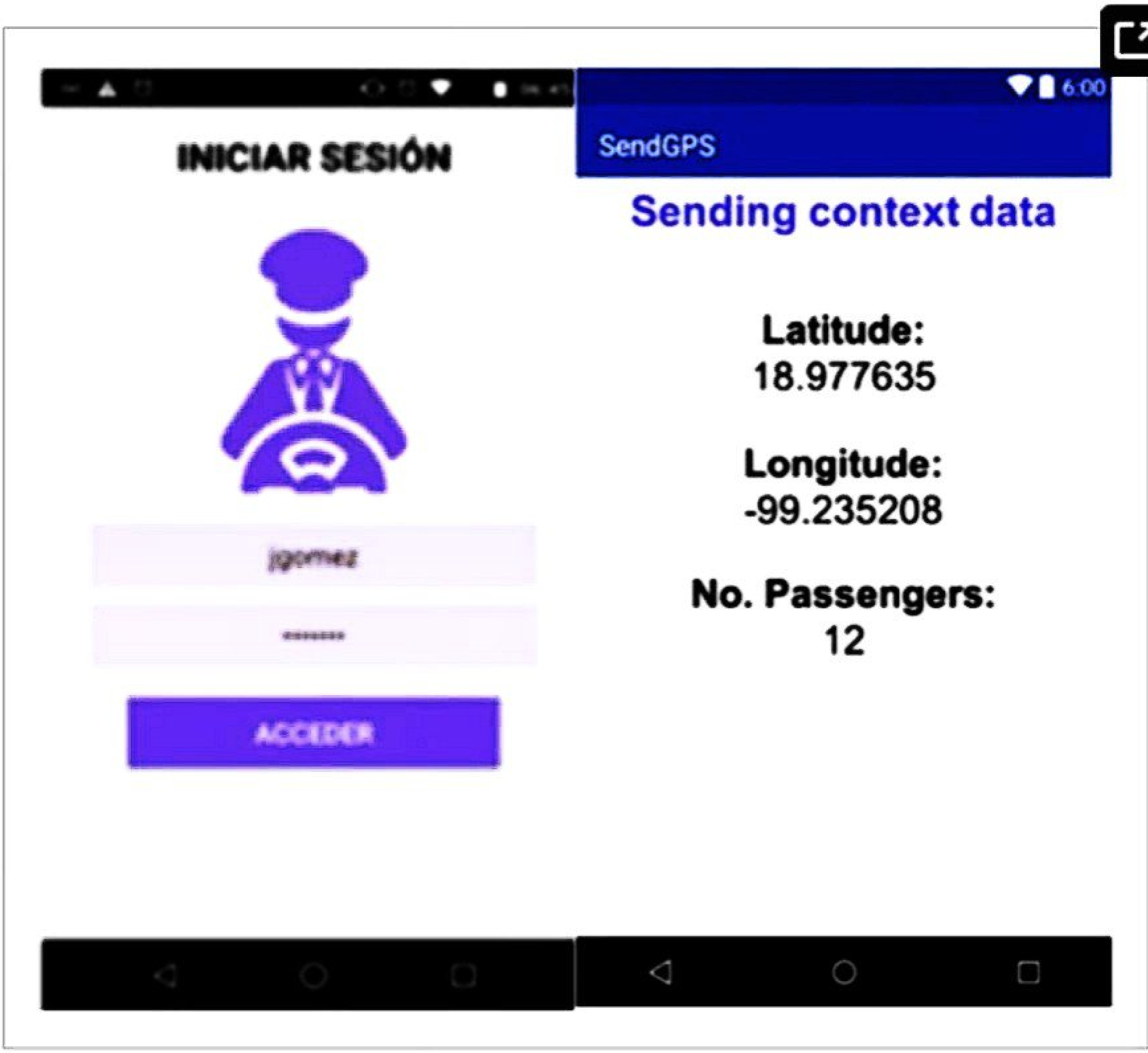
In this example, you would need to replace `public_transport_data.csv` with the actual file path or data source containing the public transport data. The program would then perform the necessary calculations and logic to optimize the routes and print out the optimized solution.

4. Implementation of the Smart Information System

This section presents a description of the architecture of the smart information system for mobility. The selected architecture is the Client-Server, where the mobile apps send their requests to the server and the server performs the processing and sends the answer to the apps. **Figure 11** presents the proposed architecture of the Smart Information System. From the client-side point of view, the architecture integrates three applications that compose our solution: The Management Urban Passenger Transport App, Monitoring Drivers Context Information App, and Passenger Transport App.

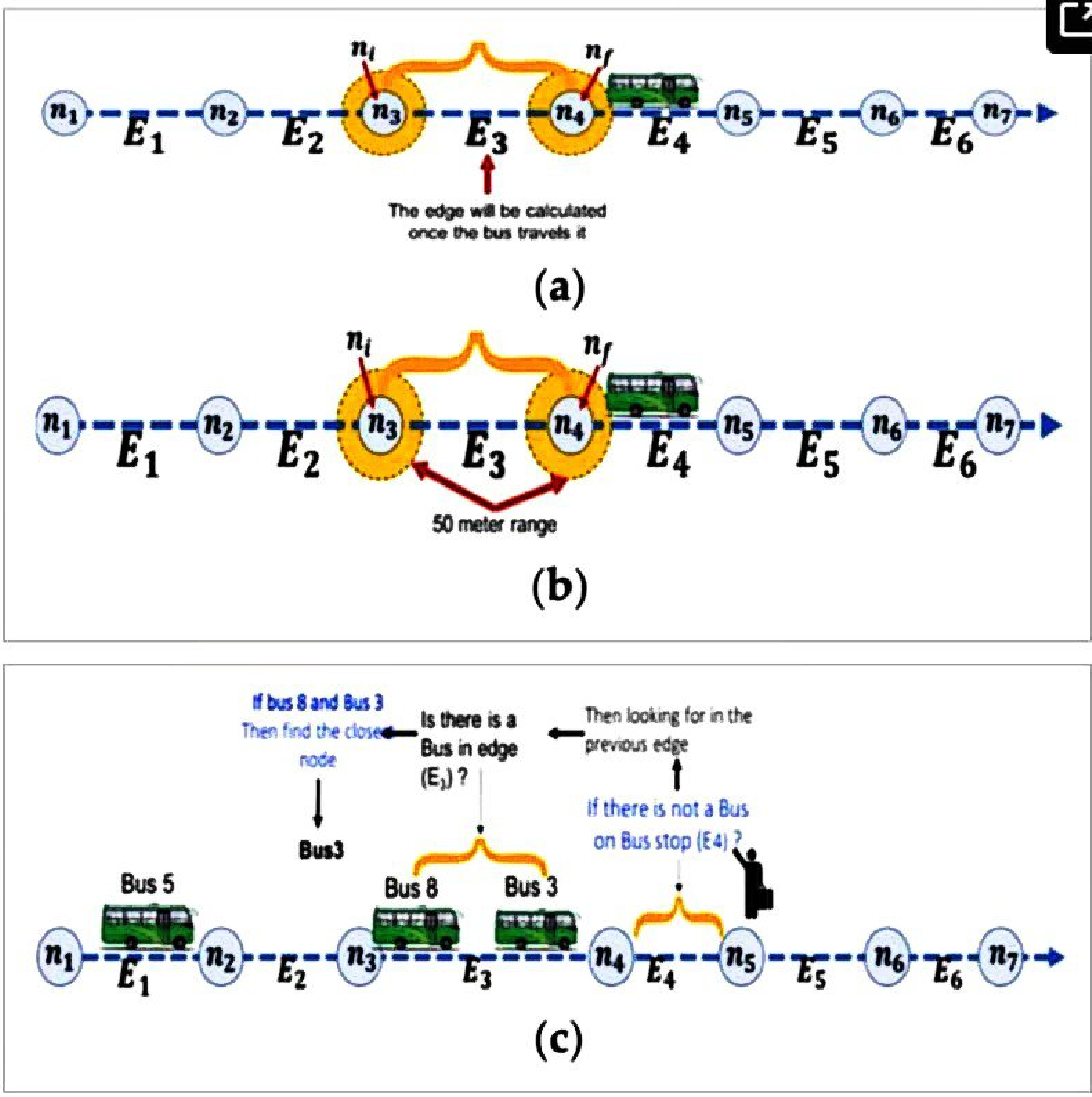


This software application has been developed to monitor, collect, and send the context information of each bus that circulates on the route to the Web Server and to store this information in a database. To do this, the bus driver needs to be logged into the app. This module periodically communicates the following data to the Web Server: the GPS position of the bus, the seat availability, the bus unit number and the route on which the bus is circulating. **Figure 15** shows the login interface for the bus driver, the interface for bus location data, and the information of the passenger counter.



- The value obtained (tEc) is averaged with the value of the database (tEdb). The greater weight should be given to the new value (because it presents the current traffic conditions) and lower weight should be given to the value stored in the database. The value of tEdb is calculated using Formula (2). The values 0.6 and 0.4 that modify the formula are used depending on the current of traffic conditions which increase or decrease the arrival time.

$$\mathrm{tEdb} = (\mathrm{tEc} \times 0.6) + (\mathrm{tEdb} \times 0.4) \tag{2}$$



The system architecture is composed of four main phases: 1. The Route Design phase to design a specific route in the city and to indicate its corresponding bus stops. 2. The Passenger Counter Solution Development phase to propose a small device to manage the users getting on/off of the bus. 3. The Context-Sensitive Service development phase to obtain, preprocess, and send the information about Urban Passenger Transport. 4. The Prediction of Arrival Time Solution phase to provide a software solution that enables the user to visualize the predicted arrival time of buses and also the seats available on that bus. The context information produced by the bus driver's smartphones and the ultrasonic counter device is sent to the cloud using an IoT approach, where devices capture information and send it to the cloud for further analysis.

(A) Phase 1: Design of Public Bus Routes

The first phase of the proposed methodology consisted of the design of the routes of the public transport buses. The objective is to define the routes and the bus stops of the public transport agencies that need to be monitored using the proposed system. This is done using GPS location points on the route. It is important to point out that only a small number of cities in Mexico have these bus routes specified in the GTFS (General Transit Feed Specification), which indicates the routes of public transport, the bus stops, and the frequency of buses at each bus stop. In that situation, an alternative method needs to be used to indicate the route and the corresponding bus stops.