

# Bluetooth Based Airport Enquiry System Using Python

Mainak Samanta, Ankur Bhattacharya, Koushik Ghosh, Ritidrita Dhara

Department of Electronics and Communication Engineering  
Supreme Knowledge Foundation Group of Institution

**Abstract-** Here we are proposing a Bluetooth based autonomous system which has features, such as; check route from one point to another within a premise, providing rate charts of the nearby coffee shops etc. We have used **Python** language to design and implement the protocols of this system due to its easy syntax and rapid prototyping features. The methodology section describes the implementation in detail and result section show that the proposed system works as per our expectation. We hope that the concept provided in this report will aid the developers' community in near future.

**Index Terms-** system description, Bluetooth, dictionary, results

## I. INTRODUCTION

The system is primarily based on Bluetooth communication with python for the software design.

**Python** is a widely used general-purpose, high-level programming language. Its design philosophy emphasizes code readability, and its syntax allows programmers to express concepts in fewer lines of code than would be possible in languages such as C. The language provides constructs intended to enable clear programs on both a small and large scale.

**Bluetooth** is a wireless technology standard for exchanging data over short distances (using short-wavelength radio transmissions in the ISM band from 2400–2480 MHz) from fixed and mobile devices, creating personal area networks (PANs) with high levels of security.

A master Bluetooth device can communicate with a maximum of seven devices in a piconet (an ad-hoc computer network using Bluetooth technology), though not all devices reach this maximum. The devices can switch roles, by agreement, and the slave can become the master (for example, a headset initiating a connection to a phone will necessarily begin as master, as initiator of the connection; but may subsequently prefer to be slave). The Bluetooth Core Specification provides for the connection of two or more piconets to form a scatternet, in which certain devices simultaneously play the master role in one piconet and the slave role in another.

## II. SYSTEM DESCRIPTION

This system is based in airport. Airport is a huge premise and it is often quiet hard for boarders to find places such as terminals, coffee shops, counters, washrooms etc and moreover their directions. Though indications through direction boards are available, but one has to search for it. According to our project, one can find their desired destination within the airport premise using preinstalled helpdesk or an enquiry booth through which one can access this system. This system consists of a displayed window with few buttons like "Terminal", "Coffeeshop", "Routemap", etc. Function of each button is as follows:

- **Button "Terminal":** Function of this button is to find the terminals and to show the list of all terminals available in that airport.
- **Button "Coffeeshop":** If clients click on this button then he/she will see the list of nearest coffee shops and below this list there is a button named "Ratechart". "Ratechart" button provides name of the available products in a particular coffee shop and also their price.
- **Button "Washroom":** It shows list of nearest washrooms.
- **Button "Medical":** It shows list of nearest medical outlets and free check-up centre's.

- **Button “Routemap”:** It is the most important button of this system. If you click this button, an entry box is opened which wants your destination name. After writing destination name if you click “ok” button, it provides you a route map or graph of your path.

More buttons can be created as per requirement.

### III. DATA STRUCTURES

This system consists of six data base files. These files are used as a dictionary in the system. Therefore data’s are stored in the file as a form of dictionary. Tuples are also used in this system, within a dictionary to show the valid information. These are frequently used to represent simple data structures. If any data of those files will change Server’s needs to update these files. All data based files are described below:

- **Terminal:** This file is content name of all the terminals. The file is shown below:  
**{Mac id 1: [Terinal1, Terminal4, Terminal3], Mac id 2: [Terminal2, Terminal5 Terminal6]}**  
Here the term ‘Mac id’ denotes Mac id of a particular helpdesk. Mac id 1 is Mac id of helpdesk 1. As we know helpdesk is connected with all sectors through Bluetooth communication medium. So, a particular helpdesk can find its nearest sectors. Thus, in this file, “[Terinal1, Terminal4, and Terminal3]” is the list which shows nearest terminals of helpdesk 1. By this way all data in the terminal file are stored. This file is used in the server’s system terminal function.
- **Coffee shop:** Name of all the coffee shops is stored in this file. The file shows below  
**{Mac id 1: [café coffee day, coffee5, coffee6], Mac id 2: [coffee2, coffee4, coffee zone]}**  
Structure of this file is same as structure of terminal file. Only difference is that it shows nearby coffee shops of a particular helpdesk.
- **Washroom:** It consists of all the washrooms. Airport is a large premises therefore sometimes it is difficult to find washrooms here. Thus in this system, name of the washrooms are stored in the washroom file. In the server’s system washroom function uses this file. This is also a data based file and the structure of the file is same as above two files.  
**{Mac id 1: ["washroom1","washroom3","washroom5"], "mac2": ["washroom4","washroom2"] ...}**
- **Place:** Place is also a data base file but the structure of this file is little bit different with refer to above three files. This file provides coordinate of all sectors/zone in the form of dictionary. The file shows below  
**{Mac id 1:[20,53],.....,Terminal1:[12,21],Terminal2:[]......,Coffee2:[45,43],.....}**  
In this dictionary all sectors or zones are saved with their coordinates. Coordinates are stored in the form of list.
- **Rate:** It is a file used in the rate chart function. This file contains price and available product of all coffee shops. The structure of the file shown below  
**{café coffee day:["coffee -60”, “cappuccino -150”, “biscuit -20”], coffee2:["coffee-55”, ....],..}**

### IV. ALGORITHM

1. The client sends a request to the server, by clicking one of the buttons.
2. The server then verifies the request and prints the corresponding list to the client.
3. On viewing the list provided, the user re-sends further request for detailed information, such as products available, rate chart, etc. by simply typing one, in the list.
4. If the request is regarding ‘coffee shop’, it shows the list of available products, rates, etc.
5. If the request is regarding ‘terminal’, it shows the list of available terminal in the airport.
6. If the request is regarding ‘medicine’, it shows the list of available outlet and free checkup booths.
7. If the request is regarding ‘washroom’, it shows the list of nearest washrooms.

8. Now if the request is 'route map', then it exhibits the direction of the destination from that particular help desk.

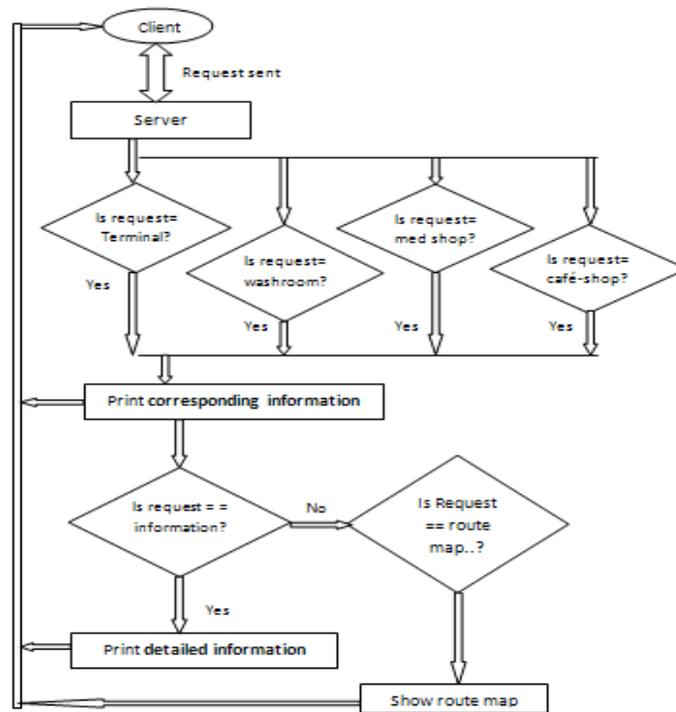


Figure 1: Flowchart demonstration of the system

## V. RESULT AND OUTCOME

In this section we will describe working methods of the software step by step with help of screenshots.

At first the window in figure 2, will appear in the display of 'helpdesk'. A user can click any of them for their enquiry.

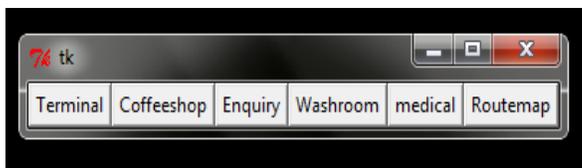


Figure 2: display window at helpdesk



Figure 3: screen, when terminal button is clicked

Suppose a user requested for terminals. Then it will display the list of terminals nearby to that helpdesk as in figure 3. It indicates that there are three nearby terminals from this helpdesk, namely, "terminal1", "terminal2" and "terminal3".

A user may also ask for 'coffee shops'. Then name of the nearest coffee shops from that helpdesk will be shown. A button will appear also as 'rate chart'. If a user types the name of a particular coffeshop (as an example 'cafe2' as shown in figure 4) in the entry box then corresponding rate chart will be appeared in a different window.

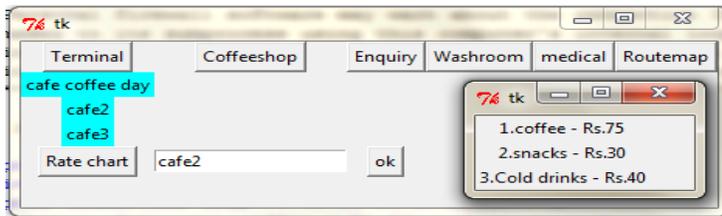


Figure 4: screen, displaying the rate chart of a coffee shop

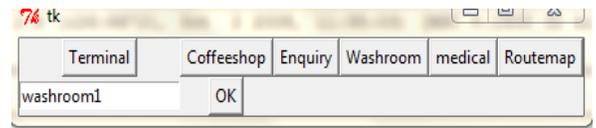


Figure 5: after clicking 'routemap' button

This software can display also route direction of a particular place from the helpdesk.

Like as below there will be 'route map' button. If clicked an entry box will get opened as shown in figure 5. Then, if the user writes the name of a particular place in the entry box and press 'ok' button, direction will be shown.

In this case the "washroom1" is in the north-east direction from that particular helpdesk. Similarly directions of other destinations can also be routed by this application.

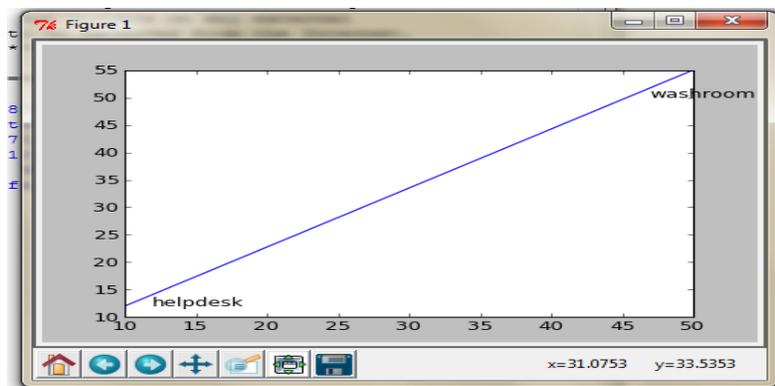


Figure 6: representing the route direction of 'washroom1'

## VI. CONCLUSION

The detail description and the concept of implementation of the proposed system are described in the methodology section. We hope that this system will be of great use in a large premise. We also hope that this system will aid the developers' community in future. This system lacks the security features. We will take security measures in order to prevent outside attacks and unauthenticated usage. Moreover the route demonstration would have been better using SQL database. Further implementation can be done based on shopping malls, stadiums, banks, railways stations, etc.

## AUTHORS

**First Author** – Mainak Samanta, Department of Electronics and Communication Engineering, Supreme Knowledge Foundation Group of Institution , mainaksamanta007@gmail.com.

**Second Author** – Ankur Bhattacharya, Department of Electronics and Communication Engineering, Supreme Knowledge Foundation Group of Institution

**Third Author** – Koushik Ghosh, Department of Electronics and Communication Engineering, Supreme Knowledge Foundation Group of Institution, koushik.ghosh91@gmail.com

**Fourth author** – Ritidrita Dhara, Department of Electronics and Communication Engineering, Supreme Knowledge Foundation Group of Institution

**Correspondence Author** – Mainak Samanta, mainaksamanta007@gmail.com, samantamainak@yahoo.com.