PAY AS YOU PARK SMART PARKING SOLUTION: INTERNAL NAVIGATION SYSTEM FOR SRI LANKAN PARKING AREAS

Lahiru Vernon Ferreira

(IT18013924)

B.Sc. (Hons) Degree in Information Technology

Department of Computer Science and Software Engineering

Sri Lanka Institute of Information and Technology

Sri Lanka

October 2021

PAY AS YOU PARK SMART PARKING SOLUTION: INTERNAL NAVIGATION SYSTEM FOR SRI LANKAN PARKING AREAS

Lahiru Vernon Ferreira

(IT18013924)

Dissertation submitted in partial fulfillment of the requirement for Bachelor of Science in Information Technology specialization in Software Engineering

Department of Computer Science and Software Engineering

Sri Lanka Institute of Information and Technology

Sri Lanka

October 2021

DECLARATION OF CANDIDATE

“I declare that this is our own work, and this proposal does not incorporate without acknowledgement any material previously submitted for a degree or diploma in any other university or Institute of higher learning and to the best of our knowledge and belief it does not contain any material previously published or written by another person except where the acknowledgement is made in the text. Also, I hereby grant to Sri Lanka Institute of Information Technology, the nonexclusive right to reproduce and distribute my dissertation, in whole or in part in print, electronic or other medium. I retain the right to use this content in whole or part in future works (such as articles or books).”

Signature :

Date : 13/10/2021

DECLARATION OF SUPERVISORS

The above candidates are carrying out research for the undergraduate Dissertation under my supervision.

Name of supervisor: Ms. Nadeesa Pemadasa

Name of co-supervisor: Ms. Thamali Dissanayake

Signature of supervisor: Date:

Signature of co-supervisor: Date:

DEDICATION

This study is devoted to all the people who drive on the road on a daily basis and have various challenges in locating a better and widely available parking yard to park in. Parking is an essential activity in metropolitan areas. This is a challenge that people face daily.

My idea is to forecast a user's position when they enter a yard and make it easier for them to find an available parking slot. Given the difficulties that drivers confront, my goal is to implement smart internal parking navigation for better parking.

ACKNOWLEDGMENT

I would like to present our heartfelt gratitude for the immense support, guidance and motivation provided by our supervisor Ms. Nadeea Pemadasa (Assistant Lecturer - Faculty of Computing SLIIT) throughout the research and for her valuable feedback to make this component more effective and successful. Her guidance showed the way to get into competitions, work with industry experts, show talents and bring glory to SLIIT. Also, we offer our special gratitude for our co-supervisor Ms. Thamali Dissanayake (Lecturer - Faculty of Computing SLIIT), who guided me, and I would like to thank Dr. Dharshana Kasthurirathne (Professor- Faculty of Computing SLIIT), for mentoring us to sort out the research topic to the team. In addition to that my special thanks goes to Dr. Jagath Wickramarathne (Senior Lecturer - Faculty of Computing SLIIT) for suggesting technologies which I can use for an internal navigation. I would like to thank our parents for their patience, time and providing resources to acquire the needful and for other expenses. A special thanks to the fellow research members who worked alongside to make this research more effective and success.

ABSTRACT

Urban areas and dense communities are expanding, so the need for new applications to assist in planning and optimization is rising and although there are many both inside and outside parking areas in Sri Lanka, in present Sri Lanka, an urban area like Colombo faced a huge traffic jam due to inefficient vehicle parking, the number of vehicles used in the country by people, etc.

When focusing on vehicle parking in Sri Lanka, sometimes people are unable to find a free parking slot, and even if people find a slot, it is hard to find the exact way to the parking slot.

"Pay as You Park" is the proposed smart parking system and internal parking navigation is one of its' research components. Users can find the way to the reserved parking slot through a mobile application, and the system is based on BLE Beacons to find the way to a particular parking slot. Three Beacons will be used to track user's position(Triangulation).

By the implementation of this smart parking system, users can find the way not only to the parking area but also to the parking slot which is not reserved/free inside the parking area. Users only need a mobile device that has Bluetooth and good network connectivity. When a user enters to the parking area, the mobile application will guide the user to the available parking slot. "Pay as You Park" smart parking system can be utilized in day-to-day parking issues faced by people in the present days.

Keyword: Triangulation, BLE Beacon

TABLE OF CONTENT

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

DNN - Deep Neural Network

NN - Neural Network

RSSI - Received Signal Strength Indicator

ToA - Time of Arrival

TDoA - Time Difference of Arrival

1. INTRODUCTION

1.1. Background and Literature Survey

1.1.1 Background

People spend a lot part of their daily lives indoors. Locations such as universities, colleges, hospitals, working places, shopping malls and shopping centers, etc. Indoor positioning is a trivial problem, as satellite-based approaches (E.g. GPS) do not work properly within buildings. Wi-Fi, ZigBee, RFID and BLE are more for indoor positioning research, favored. Figure 1.1 shows the people who use and do not use an indoor parking area.

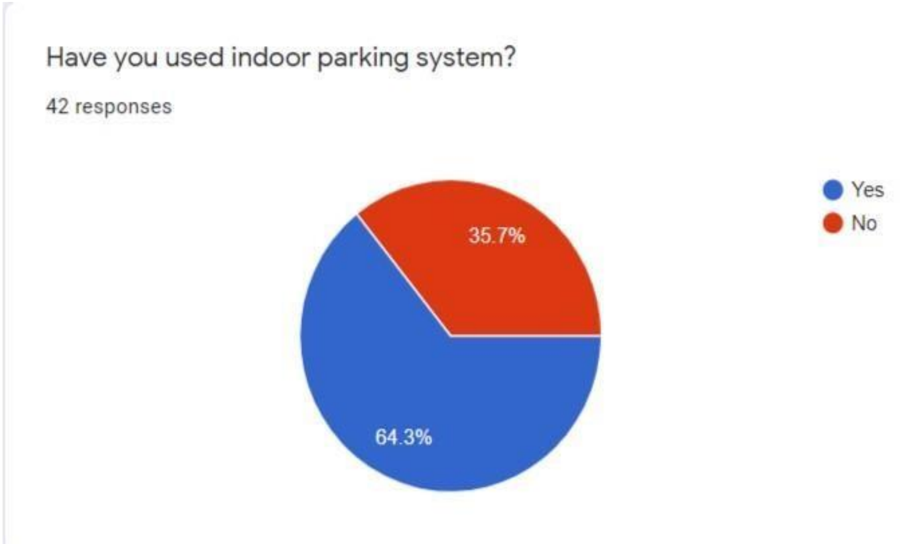


Figure 1.1. Percentage of people who use the indoor parking

1.1.2 Indoor Position Technology Comparison(Literature Review)

There are many kinds of indoor positioning in world-wide. Some of them are,

- Wi-fi based indoor positioning
- BLE Beacon based indoor positioning
- UWB based indoor positioning
- RFID based indoor positioning

Following is the brief description of the technologies,

a. Wi-fi Based Indoor Positioning

Wi-Fi-based real-time indoor positioning systems, such as smartphones, tablets, and Wi-Fi tags, locate and track active Wi-Fi devices. Depending on the preconditions, the accuracy of Wi-Fi used for server-based indoor localization ranges from eight to fifteen meters. [4]

Advantages :
Capability to control all Wi-Fi-enabled devices, ability to track visitor behavior, wide range of devices (up to 150m).

Disadvantages :

The degree of accuracy offered by BLE or RFID, high latencies and randomized MAC address usage are difficult to achieve when the system is not connected to the Wi-Fi network.

b. BLE Beacon Based Indoor Positioning

Beacons are small wireless devices that use Bluetooth Low Energy, also known as Bluetooth Smart, to relay signals. They are relatively inexpensive, can run for up to five years on button cells. With Bluetooth 4.0.0, accuracy is usually less than eight meters. The new 5.1 version of the Bluetooth specification allows for path finding use cases and offers less than one meter of accuracy. Beacons are scalable and highly portable in all sorts of different formats.[4]

Advantages :
Flexibility and Efficiency of Cost

Disadvantages :
Signal dispersion attenuations inside houses, layout alteration instability and radio interference.

c. UWB Based Indoor Positioning

Ultra-wideband is a radio technology with a short range. The precision is less than 30 cm, which is slightly higher than when dealing with Wi-Fi or beacons. You can also reliably calculate height differences.[6]

Advantages:
High accuracy, low latency times, no interferences

Disadvantages:
Higher cost and battery life is less than beacon

d. RFID Based Indoor Positioning

RFID is a type of wireless communication that identifies objects through radio waves. Passive RFID technology only operates close to specialized RFID readers providing a 'point-in-time' role.[5]

Advantages:
Very high accuracy, no battery needed

Disadvantages:
Short range signal transmission, installation need significant planning and infrastructures can be expensive.

1.2 Research Gap

"Pay as You Park" is the system going to be implemented in the research project. In this system, users are able to find the way to the indoor parking slot, and the proposed indoor navigation technology is BLE beacon-based indoor positioning. At the heart of the indoor positioning market is Bluetooth. The technology itself is not new; since the 1990s, Bluetooth's functionality has been well-known. But it was the emergence of the BLE version of energy-saving Bluetooth that paved the way for many new application scenarios, making Bluetooth an industry standard available today on most devices.

1.2.1 Indoor Navigation Using Bluetooth Beacon

Since GPS does not run indoors, Bluetooth is a strong alternative, and the go-to option for indoor navigation applications is BLE beacons. Beacons can send out Bluetooth signals, but beacons are not able to transmit them. Beacons are relatively inexpensive, can run up to several years on button cells, and have a maximum range of 30 meters indoors (Ranges are different from Beacon series to series) but now in industry has discovered new beacons with hundred meters accuracy. A location accuracy of up to one meter can be achieved. Positioning happens on the user's mobile device in client-based solutions, ensuring optimal data security. There is a need for an app and Bluetooth must be enabled. Also possible is a server-based solution (assets or individual tracking) using beacons.

Several beacons are needed for positioning in client-based applications. Without much effort, beacons can be glued or screwed to ceilings or brick walls and can be easily incorporated into any environment.

Indoor Positioning Service	Technology Name	Accuracy	Advantages	Disadvantages	Cost (Installation/Unit)
Radar	Wifi	2-3m	Low Price/existing infrastructure	Low accuracy/complex system	Low/Low
BLE Beacon	**Bluetooth**	**0.5-3m Present there are beacons with high accuracy (100m)**	**Smartphone user unit/ease of access**	**Need many beacons for better precision**	**High/Low**
Ubisense	UWB	15cm	Very precise/very robust	Expensive Unit and installation	High/High
LANDMARC	RFID	2m	Very cheap user unit	Location delay for 7.5s	High/Low
Lock8	Ultrasound	3cm	Smart phone user unit/	Require new infrastructure for every unit	Low/Low

			precise		

Table 1.1 Technology Comparison Table

1.2.2 Ranged Based Schemas

There are two systems based on range. Two of these are distance estimation and position estimation. Attempts have been made in the literature to reduce distance error and enhance location accuracy. The centroid localization approach proposed by employs beacon coordination to estimate the location of unidentified mobile placement using the centroid formula, but the position accuracy is relatively low with this algorithm. Weighted Centroid Localization Method uses weight to estimate location as a factor. The AWBCL algorithm, which is based on the WCL method, has improved spot accuracy, but location error remains high. [5] RSSI-based and ToA/TDoA-based distance estimation algorithms are both available. The following are the advantages and disadvantages of the distance estimate methods. Pros and Cons of RSSI Based :

Pros: * Low Complexity

* No Time Sync

* Low Power

Cons: * Low Accuracy

Pros and Cons of ToA/TDoA Based:

Pros: * High Accuracy

Cons: * High Complexity

* Time Sync

* High Power

2. OBJECTIVES

2.1 Main Objective

The major goal of this "Pay as You Park" Smart Parking solution Thesis is to give an optimal internal navigation system by tracking users to securely park their vehicle effectively by tracking the user. Users can save time by using the proposed system.

Because smart solutions play an important part in society, when users become accustomed to this system, the difficulty in parking their vehicle will be lessened. This fact illustrates unequivocally that drivers can avoid wasting crucial time.

2.2 Specific Objectives

- Develop a user-friendly mobile application that supports cross-platform navigation to illustrate interior parking yard navigation. Provide a rich user experience through the application.
- Identify the user's position and show it on the internal parking yard floor map.
- Implement an algorithm to get users position by giving distance between the user and three beacons as given inputs. The method uses here is called triangulation.

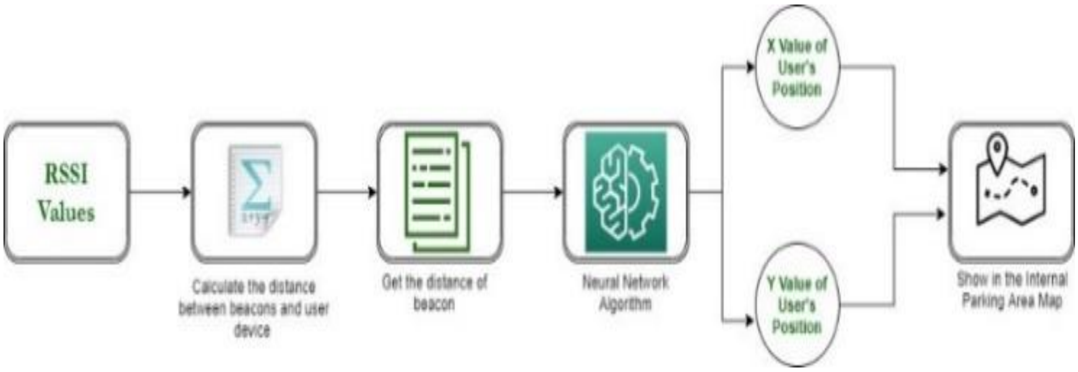
3. METHODOLOGY

3.1 Methodology of the System

According to present systems in Sri Lanka, vehicles waste time trying to discover the exact free slot inside a parking lot after entering one. When users enter the parking lot mobile application, they must select the name of the parking area. The system then displays a map of the parking space as well as the available free slots within the parking area. The database details for free parking slots are being retrieved, and those open parking slots will be displayed to users on the map.

Although a user's position may be traced using technology such as GPS when they are outside, it is difficult or impossible to track a user's position precisely when they are inside a building or underground due to obstructions such as walls, ceilings, and so on. [1] Beacons are employed in this manner by analyzing many indoor positioning systems. Beacons are less expensive than other technologies, according to the literature assessment, and they are more accurate. [2] Three beacons are used to get the users' positions using RSSI information from the beacons and the Neural Network Machine Learning Algorithm. In this method, the distance between the mobile and each beacon is calculated by using the BLE RSSI values of each beacon and passing it to the trained model. Following that, the model predicts the user's X and Y values.

After obtaining the users' X and Y values, these values are displayed on the map. The user can then view his or her current location and navigate to the free parking space



indicated on the map. Assume a building has two stories; how do users determine which floor they are on? Yes, that would be a cause for concern. A few beacons can be utilized to evade detection. Beacons have unique IDs, and the IDs of those beacons can be used to identify the floors. [3]Internal Navigation diagram structure shown in Figure 3.1.

Distance calculation between the user's mobile and the three beacons will be calculated using RSSI signal strength and equation for distance calculation is below.

$$\mathbf{RSSI = 10*N \log_{10} d + A}$$

RSSI - Received Signal Strength Indicator (dBM)

Figure 3.1. The structure of user

N - Path Loss Exponent

D - The Distance from Transmitter

A - The Reference Value, 1m away

User position Calculation, Triangulation method shown in figure 3.2

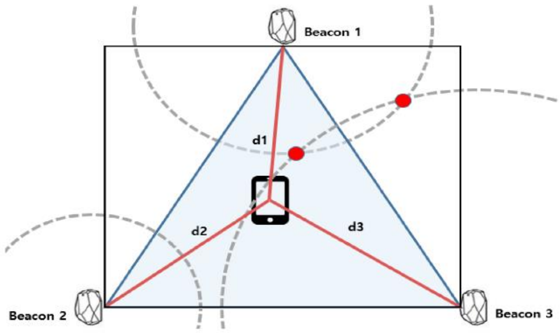


Figure 3.2 Beacon Triangulation

Workflow of the component

the workflow structure of the component is shown in figure 3.3

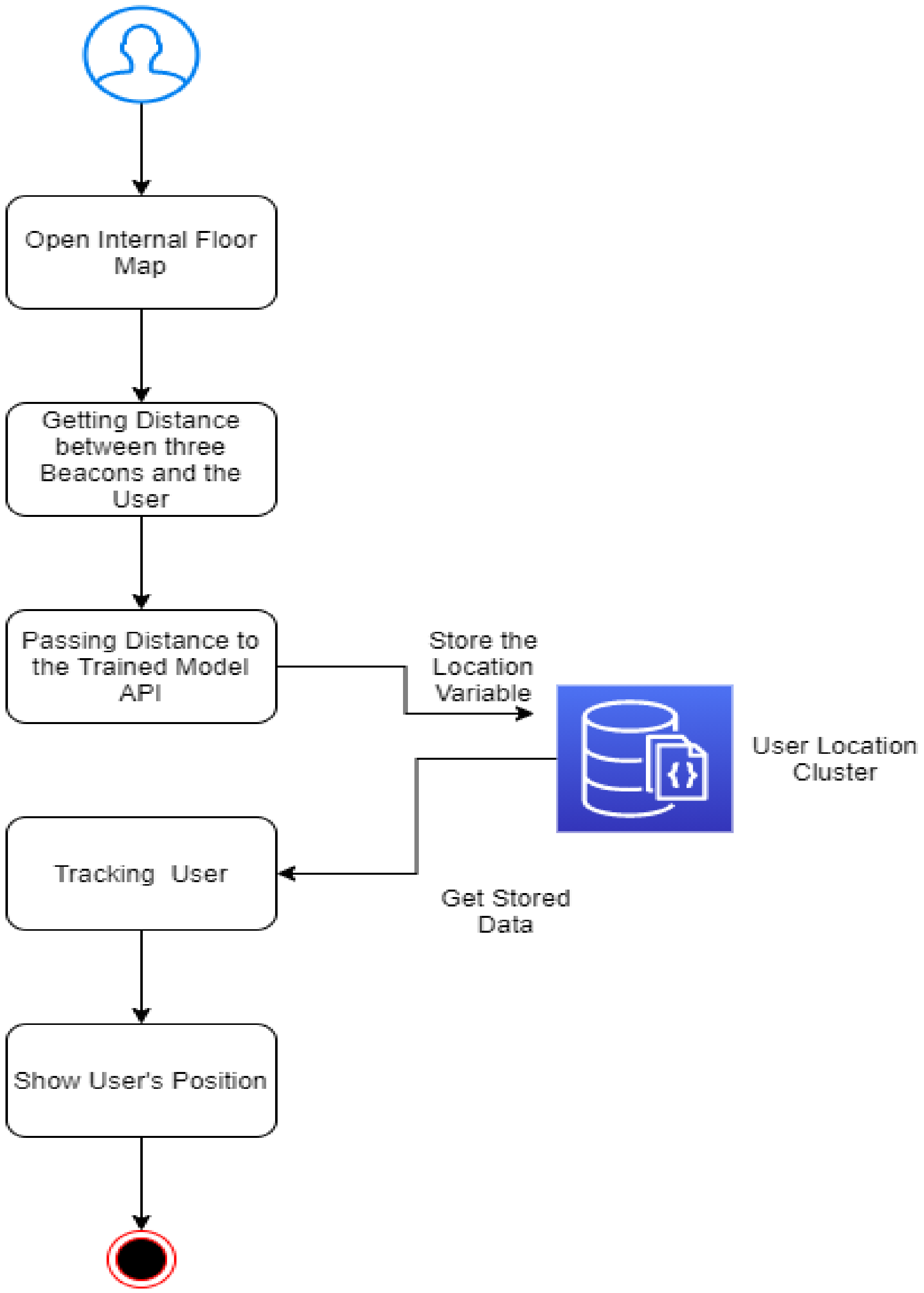


Figure 3.3 Workflow

System Architecture of the Whole System

Figure 3.4 System Architecture

The above diagram (figure 3.4) shows the full system diagram of the introduced "Pay As You Park" , the smart urban parking solution for Sri Lanka.

3.2 Commercialization of the Product

Parking is a highly desired sector in large centres, and the majority of users face parking challenges on a daily basis. With the implementation of interior parking yard navigation, users can identify vacant slots without wasting critical time, resulting in a better parking experience. The parking yards may have a high demand for the mobile application, and the application market may improve as a result of the product's efficiency.

4. TESTING & IMPLEMENTATION

4.1 Implementation

There are two applications for the Pay as You Park smart parking solution system. One is an online application for parking lot owners. We used flutter to create a mobile application for the end consumers. It is compatible with both the Android and iOS operating systems. I also used a mobile application to visualize the suggestions for my research component. To access the mobile application, users must have a mobile device with an active internet connection.

4.1.1 Mobile Application Implementation

Flutter, a widely used cross-platform mobile application development framework, is utilized to create the mobile application. It also enables widget-based development, which allows for the creation of reusable components. Flutter features a feature called hot reload that helps with development by saving time. We don't have to rebuild the entire application every time we make a minor change. With the hot reload feature, Flutter will swiftly reload the app.

Mongo DB is utilized for the application's database because of its high availability and correctness. It also includes an object-based database, which is useful for mobile platforms. When compared to other cloud services, Mongo provides more freedom. To perform operations with Mongo DB, we used a node.js application for API level development. The API is used by node.js to perform database operations.

In order to visualize the internal parking floor map, I have used an image passing to the flutter application and there are two colors(Figure 4.1) of pin markers in this application to identify the available slots inside the parking area. Availability will be getting from the Mongo DB as a Boolean value and color will be changed accordingly.

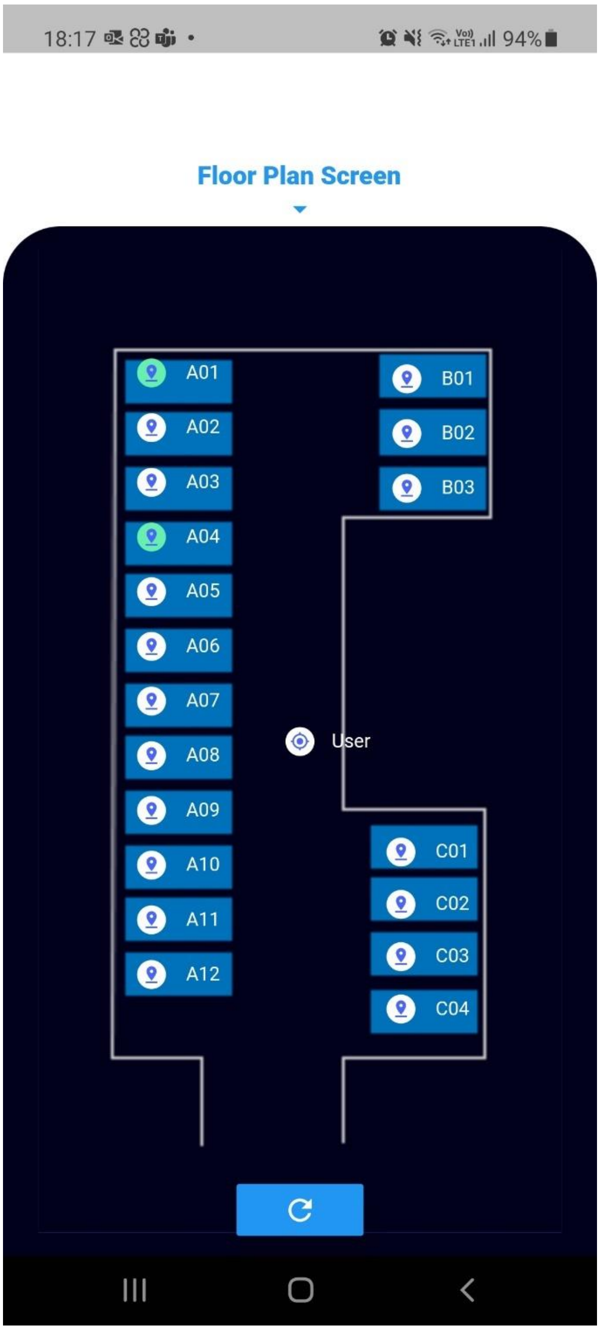


Figure 4.1 Availability View UI

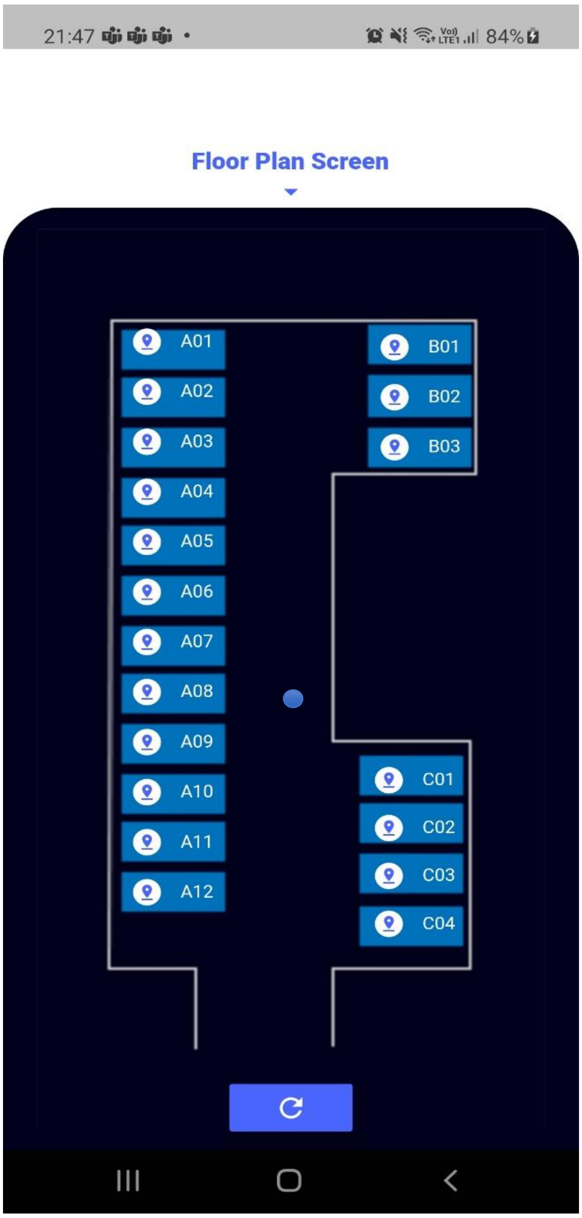


Figure 4.2. Mobile Application UI

Figure 4.2 is the floor map of the internal parking area with the slots. The code is shown in Figure 4.3 and 4.4

```
import 'package:custom_zoomable_floorplan/core/viewmodels/floorplan_model.dart';
import 'package:custom_zoomable_floorplan/view/widgets/appbar_widget.dart';
import 'package:custom_zoomable_floorplan/view/widgets/gridview_widget.dart';
import 'package:custom_zoomable_floorplan/view/widgets/overlay_widget.dart';
import 'package:custom_zoomable_floorplan/view/widgets/raw_gesture_detector_widget.dart';
import 'package:custom_zoomable_floorplan/view/widgets/reset_button_widget.dart';
import 'package:flutter/material.dart';
import 'package:provider/provider.dart';

class FloorPlanScreen extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    final model = Provider.of<FloorPlanModel>(context);

    return Scaffold(
      backgroundColor: Colors.white,
      appBar: PreferredSize(
        preferredSize: Size.fromHeight(120.0),
        child: AppBarWidget(),
      ), // PreferredSize
      body: ClipRRect(
        borderRadius: BorderRadius.only(
          topLeft: Radius.circular(40),
          topRight: Radius.circular(40),
        ), // BorderRadius.only
        child: Container(
          color: Color(0xff02011c), //02011c
          child: Center(
            child: Stack(
              alignment: Alignment.center,
              children: <Widget>[
                RawGestureDetectorWidget(
                  child: GridViewWidget(),
                ), // RawGestureDetectorWidget
                model.hasTouched ? ResetButtonWidget() : OverlayWidget()
              ], // <Widget>[]
            ), // Stack
          ), // Center
        ), // Container
      ), // ClipRRect
    ); // Scaffold
  }
}
```

Figure 4.3. Code for Indoor Map and Marker

Floor map handles form the code in Figure 4.3

```dart
import 'package:custom_zoomable_floorplan/core/models/location_model.dart';
import 'package:custom_zoomable_floorplan/core/models/models.dart';
import 'package:custom_zoomable_floorplan/core/viewmodels/floorplan_model.dart';
import 'package:custom_zoomable_floorplan/view/shared/global.dart';
import 'package:flutter/cupertino.dart';
import 'package:flutter/material.dart';
import 'package:provider/provider.dart';

class GridViewWidget extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    final Size size = MediaQuery.of(context).size;
    final model = Provider.of<FloorPlanModel>(context);
    return GridView.builder(
      gridDelegate: SliverGridDelegateWithFixedCrossAxisCount(
        //crossAxisSpacing: 2.0,
        //mainAxisSpacing: 20.0,
        crossAxisCount: 1,
      ), // SliverGridDelegateWithFixedCrossAxisCount
      itemCount: 1,
      shrinkWrap: true,
      itemBuilder: (context, index) {
        int currentTile = index + 1;
        int locationTile = 1;
        List<Slot> tileSlots =
            model.slots.where((item) => item.tile == currentTile).toList();

        List<Location> titleLocation = model.locations
            .where((item) => item.tileLocation == locationTile)
            .toList();

        return Stack(
          alignment: Alignment.center,
          children: <Widget>[
            Container(
              color: Global.blue,
              child: Image.asset('assets/images/floorPlan.png'),
            ), // Container
            Stack(
              children: List.generate(
                tileSlots.length,
                (idx) {
                  return Transform.translate(
                    offset: Offset(
                      size.width * tileSlots[idx].position[0],
                      size.width * tileSlots[idx].position[1],
```

```
            ), // Offset
            child: Stack(
              alignment: Alignment.center,
              children: <Widget>[
                CircleAvatar(
                  backgroundColor: tileSlots[idx].status
                      ? Colors.greenAccent
                      : Colors.white,
                  radius: 5.0,
                  child: Center(
                    child: Icon(
                      Icons.pin_drop,
                      color: Global.blue,
                      size: 7,
                    ), // Icon
                  ), // Center
                ), // CircleAvatar
                Transform(
                  transform: Matrix4.identity()..translate(18.0),
                  child: Text(
                    tileSlots[idx].name,
                    style: TextStyle(
                      fontSize: 6.0,
                      color: Colors.white,
                    ), // TextStyle
                  ), // Text
                ), // Transform
              ], // <Widget>[]
            ), // Stack
          ); // Transform.translate
        },
      ), // List.generate
    ), // Stack
    Stack(
      children: List.generate(
        titleLocation.length,
        (idx) {
          return Transform.translate(
            offset: Offset(
              size.width * titleLocation[idx].positionLocation[0],
              size.width * titleLocation[idx].positionLocation[1],
            ), // Offset
            child: Stack(
              alignment: Alignment.center,
              children: <Widget>[
```

```
                        CircleAvatar(
                          backgroundColor: Colors.white,
                          radius: 5.0,
                          child: Center(
                            child: Icon(
                              Icons.my_location_rounded,
                              color: Global.blue,
                              size: 7,
                            ), // Icon
                          ), // Center
                        ), // CircleAvatar
                        Transform(
                          transform: Matrix4.identity()..translate(18.0),
                          child: Text(
                            titleLocation[idx].nameLocation,
                            style: TextStyle(
                              fontSize: 6.0,
                              color: Colors.white,
                            ), // TextStyle
                          ), // Text
                        ), // Transform
                      ], // <Widget>[]
                    ), // Stack
                  ); // Transform.translate
                },
              ), // List.generate
            ) // Stack
          ], // <Widget>[]
        ); // Stack
      },
    ); // GridView.builder
  }
}
```

Figure 4.4 Code for Internal Floor Map Marker and User Current Location Icon

When I keep reading the research papers I found that the RSSI value which transmit from a beacon is not accuracy as it is expected So to make it more accurate I have used Kalman filter. Following is the code for the Kalman Filter(figure 4.5)

```
class KalmanFilter {
  KalmanFilter(double processNoise, double sensorNoise, double estimatedError,
      double initialValue) {
    q = processNoise;
    r = sensorNoise;
    p = estimatedError;
    x = initialValue;

    print("Kalman Filter initialised");
  }

  /* Kalman filter variables */
  static const double TRAINING_PREDICTION_LIMIT = 500;
  double q; //process noise covariance
  double r; //measurement noise covariance
  double x; //value
  double p; //estimation error covariance
  double k; //kalman gain
  double predictionCycles = 0;

  double getFilteredValue(double measurement) {
    // prediction phase
    p = p + q;

    // measurement update
    k = p / (p + r);
    x = x + k * (measurement - x);
    p = (1 - k) * p;

    return x;
  }
}
```

Figure 4.5 Code of the Kalman Filter

```
import 'dart:math';

class AndroidBeaconLibraryModel {

  getCalculatedDistance(double rssi, int txAt1Meter) {
    print(txAt1Meter);
    var ratio = rssi * (1.0 / (txAt1Meter + 55));
    if(ratio < 1.0) {
      return pow(ratio, 10);
    } else {
      return (1.21112) * pow(ratio, 7.560861) + 0.251;
    }
  }
}
```

Figure 4.6 Code of Distance Calculation

Above figure 4.6 shows the distance calculation to beacons by using RSSI values.

4.1.2 User's Position Prediction

User prediction model is trained by using Neural Network and below I have briefly described about the NN Machine Learning algorithm. An indoor positioning system (IPS) is a network of sensors used to find people or items in places where GPS and other satellite technologies are ineffective or fail totally, such as multistory buildings, airports, alleys, parking garages, and underground areas.

Indoor positioning is provided by a wide range of techniques and devices, ranging from reconfigured devices already in use, such as mobile phones, WiFi and Bluetooth, cameras, and clocks, to purpose-built installations with relays and beacons placed strategically throughout a defined boundary. In IPS networks, lights, radio waves, magnetic fields, auditory signals, and behavioral analytics are all used.

Neural Network

A neural network is a set of algorithms that attempts to recognize underlying relationships in a batch of data using a technique similar to how the human brain works. In this context, neural networks are systems of neurons that might be organic or artificial in nature. Because neural networks can adapt to changing input, they can produce the best possible results without having to rethink the output criteria. The neural network concept, which has its roots in artificial intelligence, is quickly gaining traction in the creation of trading systems. Graphical structure of NN is shown in figure 4.7.

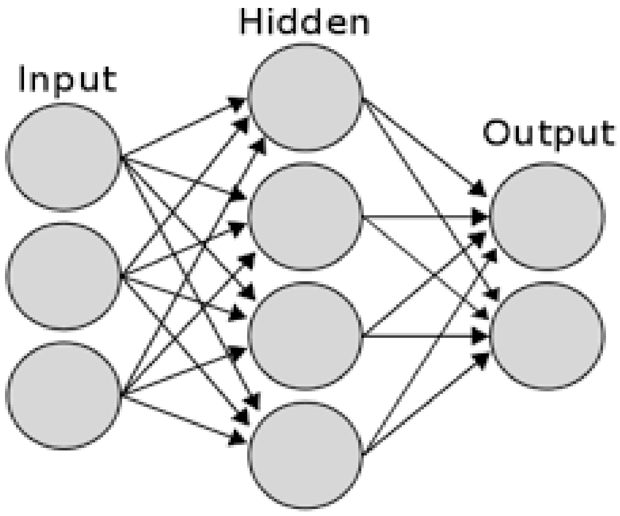


Figure 4.7 Neural Network Structure

Model of getting user's position is trained by using Neural Network Machine Learning algorithm and predicting value X and value Y is shown in figure 4.8 and figure 4.9.

Model for Predict Value X,

```
import pandas as pd

dataset=pd.read_csv('beacon_readings.csv').values

data=dataset[:,0:3]

#to predictt X
target=dataset[:,3]
from keras.utils import np_utils


categorized_target=np_utils.to_categorical(target)

from sklearn.model_selection import train_test_split

train_data,test_data,train_target,test_target=train_test_split
(data,categorized_target,test_size=0.1)

from keras.models import Sequential
from keras.layers import Dense

model=Sequential()
model.add(Dense(128,input_dim=3,activation='relu'))
model.add(Dense(32,activation='relu'))
model.add(Dense(166,activation='softmax'))

model.compile(loss='categorical_crossentropy',optimizer='adaGr
ad',metrics=['accuracy'])


model.fit(train_data,train_target,epochs=200,validation_split=
0.3)

import numpy as np

predicted_target=model.predict(test_data)

print(np.argmax(predicted_target,axis=1))
```

```python
print(np.argmax(test_target,axis=1))

from sklearn.metrics import accuracy_score

accuracy=accuracy_score(np.argmax(test_target,axis=1),np.argma
x(predicted_target,axis=1))
print(accuracy)

model.save_weights('x_weights.h5')  #to save only weights

model.save('x.model')  #to save the whole model
```

Figure 4.8 Trained Model Code of Value X

Model for Predict Value Y

```python
import pandas as pd

dataset=pd.read_csv('beacon_readings.csv').values

data=dataset[:,0:3]

#to predictt Y
target=dataset[:,4]
from keras.utils import np_utils

categorized_target=np_utils.to_categorical(target)

from sklearn.model_selection import train_test_split

train_data,test_data,train_target,test_target=train_test_split
(data,categorized_target,test_size=0.1)

from keras.models import Sequential
from keras.layers import Dense
```

```python
model=Sequential()

model.add(Dense(64,input_dim=3,activation='relu'))

model.add(Dense(32,activation='relu'))

model.add(Dense(224,activation='softmax'))

model.compile(loss='categorical_crossentropy',optimizer='adaGrad',metrics=['accuracy'])

model.fit(train_data,train_target,epochs=200,validation_split=0.1)


import numpy as np

predicted_target=model.predict(test_data)

print(np.argmax(predicted_target,axis=1))

print(np.argmax(test_target,axis=1))

from sklearn.metrics import accuracy_score

accuracy=accuracy_score(np.argmax(test_target,axis=1),np.argmax(predicted_target,axis=1))

print(accuracy)


model.save_weights('y_weights.h5')  #to save only weights


model.save('y.model')  #to save the whole model
```

Figure 4.9 Code for Trained Model for Value Y

4.1.3 API Implementation Code

The Python trained model is used to implement the API. The Technology is used to implement the API is flask. Following figure 4.10 shows the implemented API

```python
import numpy as np
from keras.models import load_model
from flask import Flask, request, jsonify

app = Flask(__name__)
model_x = load_model('x.model')
model_y = load_model('y.model')


def give_predictions(input_array):
    prediction = {}
    # get x value
    predicted_x =
np.argmax(model_x.predict(np.array(input_array).reshape(1, 3, )))

    # get y value
    predicted_y =
np.argmax(model_y.predict(np.array(input_array).reshape(1, 3, )))

    prediction['x'] = predicted_x
    prediction['y'] = predicted_y

    return prediction

    # print(give_predictions([0.877462, 0.768608, 1.457214]))


@app.route('/api', methods=['GET'])
def position():
    dict = {}
    user_input_a = str(request.args['beacon_a'])
    user_input_b = str(request.args['beacon_b'])
    user_input_c = str(request.args['beacon_c'])

    print("inputs: " + user_input_a, user_input_b, user_input_c)
```

```python
    value_float_a = float(user_input_a)
    value_float_b = float(user_input_b)
    value_float_c = float(user_input_c)

    beacon_array = [value_float_a, value_float_b, value_float_c]
    print(beacon_array)
    print("####################################################")
    prediction = give_predictions(beacon_array)

    print(prediction)
    print("*****************************************************")
    dict['output'] = prediction
    print(dict)
    return [dict]


if __name__ == '__main__':
    app.run(debug=False, threaded=False)
```

Figure 4.10 Flask API Code

4.2 Testing

Testing is a required step in the software development process. Because parking is such an important duty, the application's accuracy and efficiency must be tested. In the application, I performed both individual unit testing and integration testing. We offered our application to only a restricted number of users who used our application to test the location tracking inside the parking lot because we care about the privacy of the research.

Following table demonstrate summary of the testing which I have done

Function Process	Issues Yes/No
API testing	No
Location Tracking	No
Integrated Testing	No

Table 4.1 Testing Issue Clarification

5. RESULT & DISCUSSION

5.1 Results

My research component is internal navigation inside the parking area. To do this, I discovered the characteristics that have an impact on parking lot proposals. According to the research, I was able to improve UI/UX and data accuracy.

1. Simple user interface to increase user experience

As a consequence of my research, I was able to create an algorithm that can determine a user's location using three beacons.

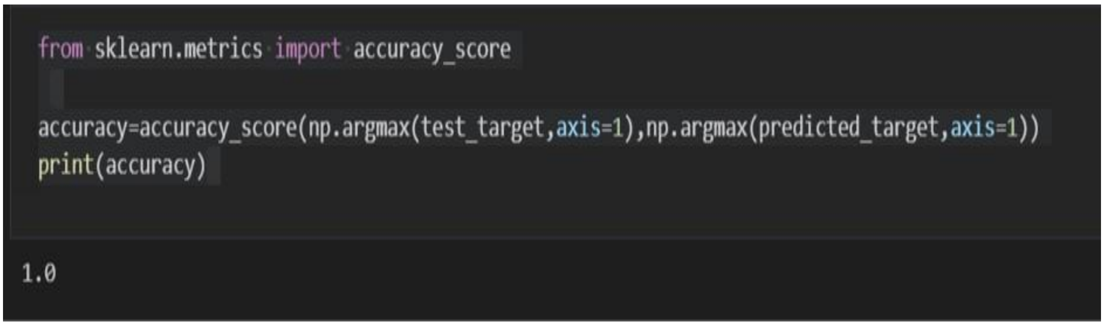
1. Mobile Application is simple to use
2. Accurate user tracking

When it comes to obtaining the user's positions, Many algorithms, including DNN, NN, and XGBoost, were used by the authors. Because of its great accuracy, the authors employed NN to train the model. When calculating a user's actual distance, the DNN and XGBoost algorithms have low inaccuracy. The authors trained the models using roughly 300 frames from a dataset, and the trained model's accuracy of getting a user's X value is 84 percent, while its accuracy of getting a user's Y value is 100 percent. Table 5.1 displays the accuracy.

Model Name	Training Accuracy of X	Training Accuracy of Y
Neural Network model	100%	90%

Table 5.1 Model Accuracy

Following figure 5.1, 5.2 show the accuracy of value X and Y.

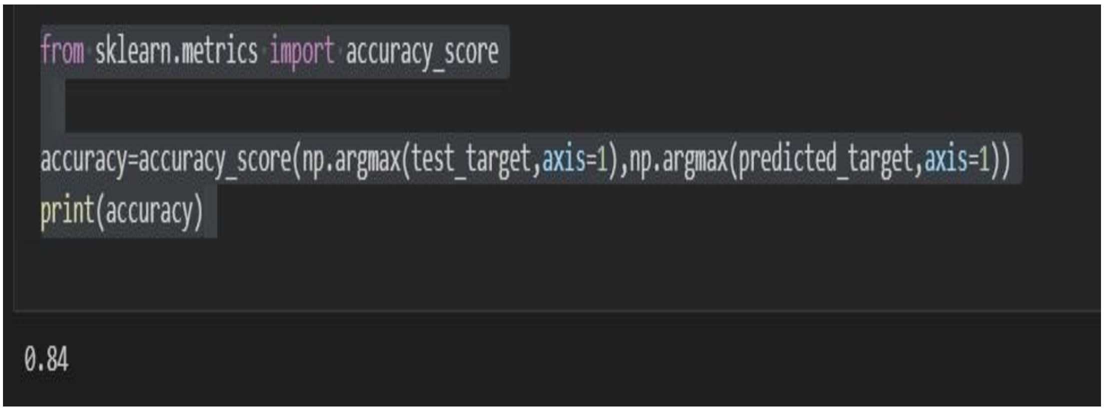


```
from sklearn.metrics import accuracy_score

accuracy=accuracy_score(np.argmax(test_target,axis=1),np.argmax(predicted_target,axis=1))
print(accuracy)

1.0
```

Figure 5.1 Accuracy of Value Y



```
from sklearn.metrics import accuracy_score

accuracy=accuracy_score(np.argmax(test_target,axis=1),np.argmax(predicted_target,axis=1))
print(accuracy)

0.84
```

Figure 5.2 Accuracy of Value X

5.2 Research Findings

Our study is based on a mobile application, which we built with flutter so that both Android and iOS users may use it. My goal is to present customers with ideal parking suggestions so that they can park their automobiles in a safe and secure manner. The research undertaken in collaboration with specialists and end users enables us to derive the following conclusions.

The following items, as well as research, are employed in research solutions to build better real-time mobile applications.

- A more user-friendly UI/UX style for the application
- Show Available Parking Slots Clearly

5.3 Discussion

My primary goal during my research was to give better interior parking navigation for users who encountered various types of challenges in their regular routine when entering a space. There are aspects that have an impact on internal parking navigation, and as part of my research, I was thinking about developing a solution that will reduce the impact on those parking navigations. In addition, I was focused on developing real-time mobile applications that provide end users with a superior UI and UX.

When we consider mobile level applications, our team focuses on cross platform mobile application development, allowing us to avoid native development. It was added as a significant benefit to the research. To get the user's position I have used Neural Network Machine Learning algorithm.

In addition, we chose node.js as the server application from which we obtained the endpoints for our mobile application. Node.js API application written in Express.js that connects to Mongo DB and performs database operations.

6. SUMMARY OF THE STUDENT CONTRIBUTION

- Identifying appropriate algorithms and the correct method for achieving the prediction

  When considering getting the user’s positions Authors have been used many algorithms such as DNN, NN, and XGBoost algorithms. Authors have used NN to train the model due to its high accuracy. DNN and XGBoost algorithms are low inaccuracy when getting the actual distance of a user. By using about 300 frames from a dataset, authors have been trained the models.

- Attractive UI designing and implementation

  The improved UI and UX we supplied can entice the majority of people to use the application. So, in order to provide users with a better experience, I created the mobile application section for my component while keeping the functionality's simplicity and performance in mind.

- Result validation and testing

  It is more crucial to test an application before releasing it to the public. In this case, I ran unit and integrated testing on my application's level and functionalities.

7. CONCLUSION

Consequently, it is clear from the estimates obtained that the BLE technology is the most optimum way for indoor navigation. The best combination of efficiency, ease of implementation and price for all technologies is given by Bluetooth Low Energy. To construct a navigation system in indoor, BLE is rightly a priority technology today. As the future work, internal parking of two or more stories will be done.

Users can obtain the ideal internal parking system to do their parking in an effective and efficient manner by using the mobile application to perform parking. They can use the application by accessing it with a mobile device that runs Android or iOS. Furthermore, the application's availability will be good, and users will be able to save time by utilizing it. Most people are having problems since they have various issues when they enter a parking lot, and I am addressing those concerns in my solution.

To create a better experience for our consumers, we used the most recent technology stack and focused on the performance of our application. To create the API service, I utilized the express.js framework, node.js, and MongoDB, and it was highly available and efficient.

The adoption of a ‘Pay as you Park' smart parking solution can lessen the challenges that users have with finding an available parking slot inside a parking area, and the parking process can be enhanced, which benefits the environment and society.

REFERENCE

[1] Improve Indoor Positioning Accuracy Using Filtered RSSI and Beacon Weight Approach in iBeacon Network Laial Alsmadi Faculty of Engineering and IT University of Technology Sydney Sydney, Australia Xiaoying Kong Faculty of Engineering and IT University of Technology Sydney Sydney, Australia Kumbesan Sandrasegaran Faculty of Engineering and IT University of Technology Sydney Sydney, Australia.

[2] Smart Parking System Based on Bluetooth Low Energy Beacons with Particle Filtering Andrew Mackey, Student Member, IEEE, Petros Spachos, Senior Member, IEEE, and Konstantinos N. Plataniotis, Fellow, IEEE

[3] Bluetooth-based Indoor Navigation Mobile System Adam Satan Institute of Information Science University of Miskolc Miskolc, Hungary satan@iit.uni miskolc.hu. Satan, A. (2018). Bluetooth-based indoor navigation mobile system. 2018 19th International Carpathian Control Conference (ICCC)

[4] Batistic, L., & Tomic, M. (2018). Overview of indoor positioning system technologies. 2018 41st International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO).

[5]Intelligent Indoor Parking Győző Gódor, Árpád Huszák* and Károly Farkas†, *† Inter-University Centre for Telecommunications and Informatics, Debrecen, Hungary *Department of Networked Systems and Services, Budapest University of Technology and Economics, Budapest, Hungary Email: {godorgy, huszak, farkask}@hit.bme.hu*

[6]Improve Indoor Positioning Accuracy Using Filtered RSSI and Beacon Weight Approach in iBeacon Network Laial Alsmadi Faculty of Engineering and IT University of Technology Sydney Sydney, Australia Xiaoying Kong Faculty of Engineering and IT University of Technology Sydney Sydney, Australia Kumbesan Sandrasegaran Faculty of Engineering and IT University of Technology Sydney Sydney, Australia

APPENDIX

Have you used indoor parking system?

Yes No

Is Indoor/Internal Parking Navigation being difficult?

Yes No

Do you like to have an indoor/internal navigation system in Sri Lanka?

Yes No