PEOPLE'S DEMOCRATIC REPUBLIC OF ALGERIA
Ministry of Higher Education and Scientific Research

UNIVERSITY ECHAHID HAMMA LAKHDAR - EL OUED
FACULTY OF EXACT SCIENCES
Computer Science Department

End Of Study Memory
Presented For The Diploma Of

ACADEMIC LICENCE

Domain : Mathematics and Computer Science
Faculty : Exact Sciences
Specialty : Information Systems

Theme

Mobile application For Locating Buses at the University Park

Presentd by :

- **CHONGARA Haitham**
- **MERAGHNI Laid**
- **LARROUM Oualid**

Proposed and Supervised by :

Mr. SOLTANI Khaled

Sustained on ……/……. /2022 In front of the jury:

M. ……..…… MCA President

M. ……..…….…… MAA Rapporteur

Academic Year : 2021 - 2022

Acknowledgments :

We are pleased to retain these few lines as evidence of gratitude and deep appreciation to all those who have contributed , from near or far , to the achievement of this work and its results .

We thank Almighty God who has always been with us and helped us throughout our journey and then of course .

I would like to thank my supervisor Mr.Soltani Khaled , for his support, seriousness, availability and valuable advice and helping us throughout the development of this work. We would like to thank the jury members who were nice enough to examine and evaluate this thesis .

Dedication

In the Name of God, Most Gracious , Most Merciful
All the Pries is due to God alone , the Sustainer of all the worlds

This work is dedicated to:

To our dear parents ,

That there is no dedication that expresses what we owe them , for their charity , passion and support . Treasures of kindness, generosity and tenderness , in testimony to our deep love and gratitude " God save you "

For our family

To all my friends with whom I shared moments of joy and happiness .

To all my teachers for your supports , education and guidance throughout my educational journey .

Abstract

The objective of this project is to implement a mobile application that allows to determine the bus location in the park of the university . The location of the bus can be determined by a " Map " on which the location of the bus is indicated .
This application helps to solve the search process and determine the bus location .

Keywords : Bus - Mobile Application - Database

Résumé

L'objectif de ce projet est de mettre en œuvre une application mobile qui permet de déterminer l'emplacement de la bus dans le parc de l'université . L'emplacement de la bus peut être déterminé par " Google Maps " sur laquelle l'emplacement de l'autobus est indiqué .
Cette application aide à résoudre le processus de recherche et à déterminer l'emplacement du bus .

Mots Clés : Bus - Application Mobile - Base de Donées

الملخص

الهدف من هذا المشروع هو إنشاء و تطوير تطبيق للهاتف المحمول يجعل من الممكن تحديد موقع الحافلة في حظيرة الجامعة . يمكن تحديد موقع الحافلة من خلال خرائط قوقل عن طريقة علامة تظهر فالخريطة تبين موقعها الحالي .
يساعد هذا التطبيق في تسهيل عملية البحث وتسريعها وتحديد موقع الحافلة بكل سهولة .

كلمات مفتاحية : حافلة ـ تطبيق هاتف ـ قاعدة بيانات

.

Contents

List of Figures

General Introduction

In an era of evolution , humans dependence on his mobile phone has increased more than ever . Where most users spend their time moving from one app to another exchanging information and searching for solutions to problems they can face , with the growing popularity of mobile apps , the current trend is to design and develop phone apps according to specific plans or patterns that serve most users . In this context , we were asked to develop a mobile application for Smart Management to find bus Location at the university park of El oued . In fact , our application is an application to facilitate the possibility of:

- Easily find the bus location .
- Know the distance from students place to the bus place .

At last we found a solution for development , to inform the student about the necessary information for the bus by locating the bus and its distance , as we dealt in our project to explain some relevant concepts about our subject , then modeled using a programming language and eventually create a successful application .

Chapter 1

Introduction and Definitions

1.1 Introduction

In our first chapter we going to give some definitions of university park and present the problem that we going to solve it while at the end mention our application and its importance to the students and university .

1.2 The Problem

Everyday , Hamma Lakhder's students have a problems to find the location of the buses in the park Since the first moment of exiting the classroom , the student starts thinking about the bus location and if it's available or no , until he reaches to the exit gate of the University when he will have another problem which it is represented by searching for the bus that going to take him home but he always take a lots of time to find it because of:

1) Random and unorganized stops of buses in the park where we find the bus today at the beginning of the station and the next day in the middle of the station .

2) Changing the destination of some buses sometimes makes a problem , when you want to take a specified bus after a long and exhausting search process , you find it heading to a different destination than yours .

3) In the case of absence of transport managers , the student faces a problem that he won't find someone to guide him , in addition the absence of a schedule at the park that own the buses numbers and their destinations make the process more difficult .

4) Changing the buses in the case of a bus malfunction or an update of the transportation schedule , especially on exam days .

5) Sometimes , searching for the bus in late times leads to missing it . as there are only two minutes left until the buses start and the student start searching for your bus so you find it went from the park .

1.3 Proposed Solution

As mobile technologies evolve and emerge, we have decided to create an app for the students who use the bus to go to their municipalities, our app allows students to find out where the bus is in the park . By this application, the student will not have to look long for the bus and will not waste his time and effort .

1.4 Definition

1.4.1 Definition of the bus park of the Hamma Lakhder University

Hamma Lakhder University park is a place designated for buses to stop , for the purpose of picking up or disembarking students . The park contains 101 buses with its drivers dedicated to transport students to and from their municipalities . The schedule of transferring students takes place in two stages:

1. Cruising: At this stage , students are transported by buses from their municipalities of residence to the university , and it is in two shifts , the first at 7 AM the second one at 8:30 PM

2. Return: and at this stage , students are transported by buses from the university to their municipalities of residence,and it is in three shifts , the first at 1:00 PM , the second one at 4:30 PM the last one at 5:30 PM .

• **Notice:** The transfer schedule is changed during the examination period in the return stage , they add another shift at 11:30 AM . We enclose in this table the number of buses and their numbers assigned to each municipality :

Municipality	Number of buses	Bus numbering
Guemar	5	46,47,49,87,99
Ghamra	3	50,53,80
Taghzout	4	09,21,72,49
Reguiba	7	51,52,53,54,55,56,86
Kouinine	6	57,59,60,61,62,20
Ourmes	3	19,45,49
Oumih ouansa	4	68,69,71,78
Ouad Alanda	3	69,70,71
Nakhla + Ogla	6	58,63,64,66,67,81
Elbiyada	7	11,13,14,15,16,17,18,67,60
Eltrifaoui	2	44,84
Hassani abdelkarim	3	23,24,77
Elzgom	4	25,26,27,77
Sidi Oune	4	28,36,37,89
Eldjedida	5	35,38,73,79,89
Elmegrane	4	32,33,34,83
Debila	4	29,30,31,88
Hassi Khalifa	6	39,40,41,42,43,85
Elrebah	6	57,59,60,61,62,65
Downtown	13	01,02,03,04,05,06,07 ,08,74,75,76,90,91

1.4.2 The importance of the application

- The importance of the application for the university :
  - Facilitation and conduct of transport in a series manner .
  - Not receiving many complaints from students .
  - Publish any update in the table automatically via the application .

- The importance of the application for the students :
  - Time economy : The student can find out the location of the bus he wants to travel on board while in the library or accommodation , or anywhere inside the university , he does not have to move to the park to find the bus's location , when he gets out of the university , he goes straight to the location where the bus is parked .
  - Using time: The bus park is within 5 minutes of the nearest college , there are other colleges such as economics , rights and others located 10 minutes or more , and university stays are some of them further away .

  Here, the student is able to economize the time to use him for reading or to prepare part of his research , and he is relieved because he know the location of the bus in the park and the time of its departure .

  Avoiding crowds of students : The student's knowledge of the position of the bus in the park through this app will enable him to choose the path that is less overcrowded with students , especially during the peak periods of students going to the park .
  - Avoid the effects of bad weather : the weather in the area is very hot or has very hot winds and dust , so if the student knows the location of the bus in the park via this app he can get to the bus quickly which avoids exposure to these air factors for longer .
  - Understanding any changes in times or buses .

1.4.3 Definition of the application

It is a purposeful mobile application made to help university students in their transportation in the university park and the process of searching for buses .

- **The parts of the application :**

  1) Welcome Page : in this page we will find a welcoming statement and a description for the application . Also it's have a two buttons , the first for the login and the second for the sign up .

2) Sign Up / Login Page : like every application , our application have a page to sign up for new users and other page to sign in for who have signed up before .
3) The Main Page : in this page we will have three buttons named by **" Cruising Time "** and **" Return Time "** and **" Downtown Buses "** , these buttons will lead you to another page have the buses and it's time to start .
- This page also have a Search bar , this tool made to make the process of searching more easier . in this tool we can search by three methods which are :
Searching by times , searching by municipality or searching by bus numbering .
4) Cruising Time Page : in this page we will have the time of buses start from the student municipalities of residence to the university .
5) Return Time Page : in this page we will have the buses that start from the university to the student municipalities of residence .
- In the last two pages the students can have the choice to choose between time phases and then it will lead them to another page have a list of municipalities
6) List of municipalities Page : this page is a list view have the name of each municipalities in El oued with a subtitle have the bus numbering
- When the student choose the municipality , he will have the choice again to choose between buses .
7) Bus numbering Page : At this page , group of buses will appear to user and when he click on one of them it will Google Maps and show him the location .

1.4.4 Purpose of the application

The main objective of the application is to solve the transport problem by enabling the student to locate the bus he/she will take in the park in front of the university. As well as to know the time of its departure , by choosing the number of the bus whose route is known beforehand , Any bus number that travels towards the area he intended , whether it's his residence , downtown , or any other location means the options offered by the various bus routes available .

1.5 Similar types of this application

1.5.1 DoubleMap Bus Tracking

DoubleMap is a real-time GPS bus tracking system .
DoubleMap allows riders to track the exact location of a bus in real time for your city .
Android app features :
- View buses in real time on the map .
- View specific routes and associated stops .
- View your current location if GPS is enabled .
- View current bus system related announcements such as delays and re-routing .
- Made specifically for Android devices, so it's smoother than the DoubleMap bus tracker mobile website .
The Application is on Google Play Store , was released on Feb 18 , 2012 , powered by DoubleMap .

Figure 1.1: DoubleMap Bus Tracking Application

1.5.2 Lynx Bus Tracker

LYNX Bus Tracker is a real-time GPS bus tracking app for Orlando's LYNX public transit system .

The Application is on Google Play Store , was released on Aug 31, 2017 , powered by DoubleMap .

Figure 1.2: Lynx Bus Tracker Application

1.6 Conclusion

We have presented in this chapter some definitions and have mentioned our problem which we will finally solve .
Next chapter we will talk about UML and show our diagrams .

Chapter 2

Modelling by UML

2.1 Introduction

In this chapter, we will recognize UML and attach the diagrams associated with our app .

2.2 UML

2.2.1 Definition

Unified Modeling language (UML) is a standardized modeling language enabling developers to specify , visualize , construct and document artifacts of a software system .
Thus , UML makes these artifacts scalable, secure and robust in execution.
UML is an important aspect involved in object-oriented software development , it uses graphic notation to create visual models of software systems . [7]

2.2.2 Origin

The goal of UML is to provide a standard notation that can be used by all object-oriented methods and to select and integrate the best elements of precursor notations. UML has been designed for a broad range of applications. Hence, it provides constructs for a broad range of systems and activities (e.g., distributed systems , analysis , system design and deployment) .

UML is a notation that resulted from the unification of OMT from :

1. **Object Modeling Technique OMT** [James Rumbaugh 1991] - was best for analysis and data intensive information systems.
2. **Booch** [Grady Booch 1994] - was excellent for design and implementation. Although the Booch method was strong, the notation was less well received (lots of cloud shapes dominated his models - not very tidy) .
3. **OOSE** (**Object-Oriented Software Engineering** [Ivar Jacobson 1992]) - featured a model known as Use Cases. Use Cases are a powerful technique for understanding the behavior of an entire system (an area where OO has traditionally been weak).

In 1994, James Rumbaugh, the creator of OMT, stunned the software world when he left General Electric and joined Grady Booch at Rational Corp. The partnership aimed to merge their ideas into a single, unified method (the working title for the process was indeed the " Unified Method ").
By 1995, the creator of OOSE, Ivar Jacobson, had also joined Rational, and his ideas (particularly the concept of " Use Cases ") were fed into the new Unified Method - now called the Unified Modelling Language1. The team of Rumbaugh, Booch, and Jacobson are affectionately known as the " Three Amigos " Other object-oriented notations have also influenced UML :
- Mellor and Shlaer [1998]
- Coad and Yourdon [1995]
- Wirfs-Brock [1990]
- Martin and Odell [1992]

UML also includes new concepts that were not present in other significant methods at the time , such as extension mechanisms and a constraint language .[8]

2.2.3 History

In 1996, the first Request for Proposal (RFP) was issued by the Object Management Group (OMG) provided the catalyst for these organizations to join forces around producing a joint RFP response. Rational established the UML Partners consortium with several organizations willing to dedicate resources to work toward a strong UML 1.0 definition. Those contributing most to the UML 1.0 definition included :

- Digital Equipment Corp
- HP
- i-Logix
- IntelliCorp
- IBM
- ICON Computing
- MCI Systemhouse
- Microsoft
- Oracle
- Rational Software
- TI
- Unisys

This collaboration produced UML 1.0, a modeling language that was well-defined, expressive, powerful, and generally applicable . This was submitted to the OMG in January 1997 as an initial RFP response .
In January 1997, IBM, ObjecTime, Platinum Technology, Ptech, Taskon, Reich Technologies and The software also submitted separate RFP responses to the OMG. These companies joined the UML partners to contribute their ideas, and together the partners produced the revised UML 1.1 response. The focus of the UML 1.1 release was to improve the clarity of the UML 1.0 semantics and incorporate contributions from the new partners. It was submitted to the OMG for their consideration and adopted in the fall of 1997.1 and enhanced from 1.1 to 1.5, and subsequently to UML 2.1 from 01 to 06 (now the UML current version is 2.5) [8]

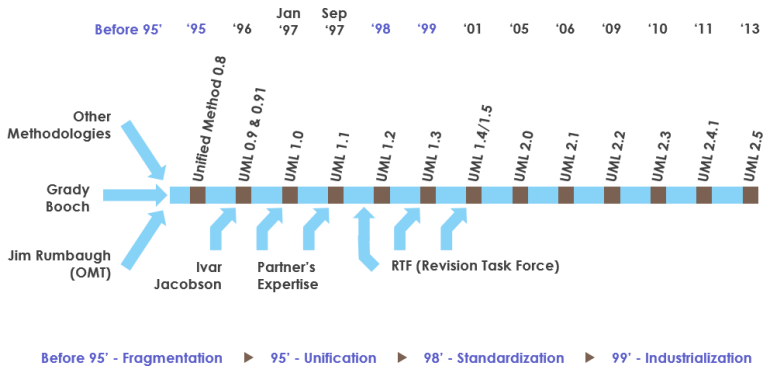

Figure 2.1: The History of UML
[8]

2.2.4 Diagrams Types

UML is linked with object-oriented design and analysis. UML uses elements and forms associations between them to form diagrams. Diagrams in UML can be broadly classified as :

A - Structural Diagrams : Capture static aspects or structure of a system , it is include :

Component Diagrams , Object Diagrams , Class Diagrams and Deployment Diagrams .

B - Behavior Diagrams : Capture dynamic aspects or behavior of the system . it is include :

Use Case Diagrams , State Diagrams , Activity Diagrams , Interaction Diagrams , Sequence Diagrams .

The image below shows the hierarchy of diagrams according to UML 2.2 [5]

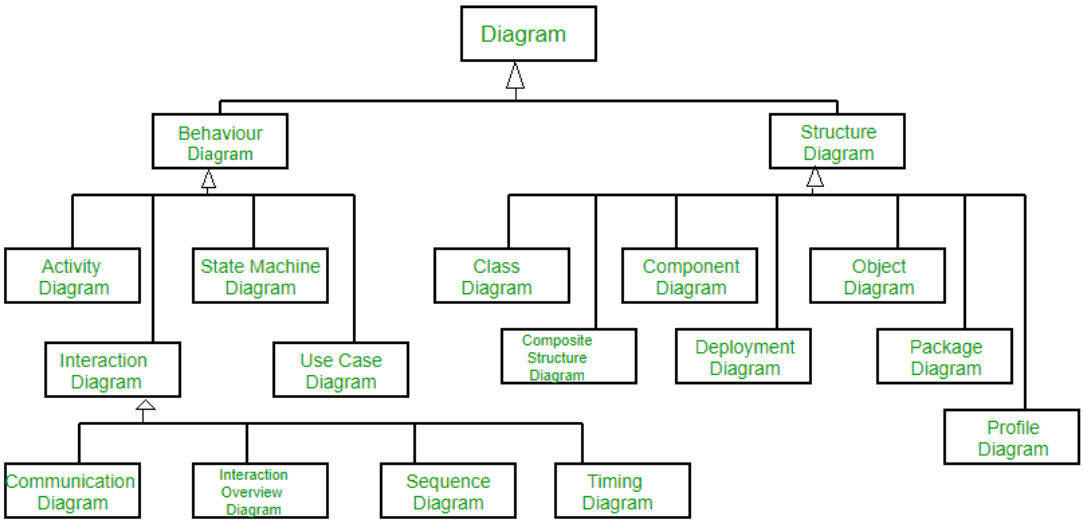

Figure 2.2: Diagrams Types of UML
[5]

2.3 Why UML

As the strategic value of software increases for many companies, the industry looks for Techniques to automate software production and improve quality and reduce cost and time-to-market . These techniques include component technology, visual programming, patterns, and frameworks. Businesses also seek methods to manage the complexity of systems as they increase in scope and scale .
In particular, they recognize the need to solve recurring architectural problems, such as physical distribution, concurrency, replication, security, load balancing, and fault tolerance. Additionally, while making some things more straightforward, the development of the World Wide Web has exacerbated these architectural problems. The Unified Modeling Language (UML) was designed to respond to these needs. The primary goals in the design of the UML summarize by Page-Jones in Fundamental Object-Oriented Design in UML is as follows :

1 - Provide users with a ready-to-use, expressive visual modeling language so they can develop and exchange meaningful models .
2 - Provide extensibility and specialization mechanisms to extend the core concepts.
3 - Be independent of particular programming languages and development processes.
4 - Provide a formal basis for understanding the modeling language .
5 - Encourage the growth of the OO tools market .
6 - Support higher-level development concepts such as collaborations , frameworks, patterns and components .
7 - Integrate best practices .

2.4 Diagrams

2.4.1 Use Case Diagram

is a graphical depiction of a user's possible interactions with a system .

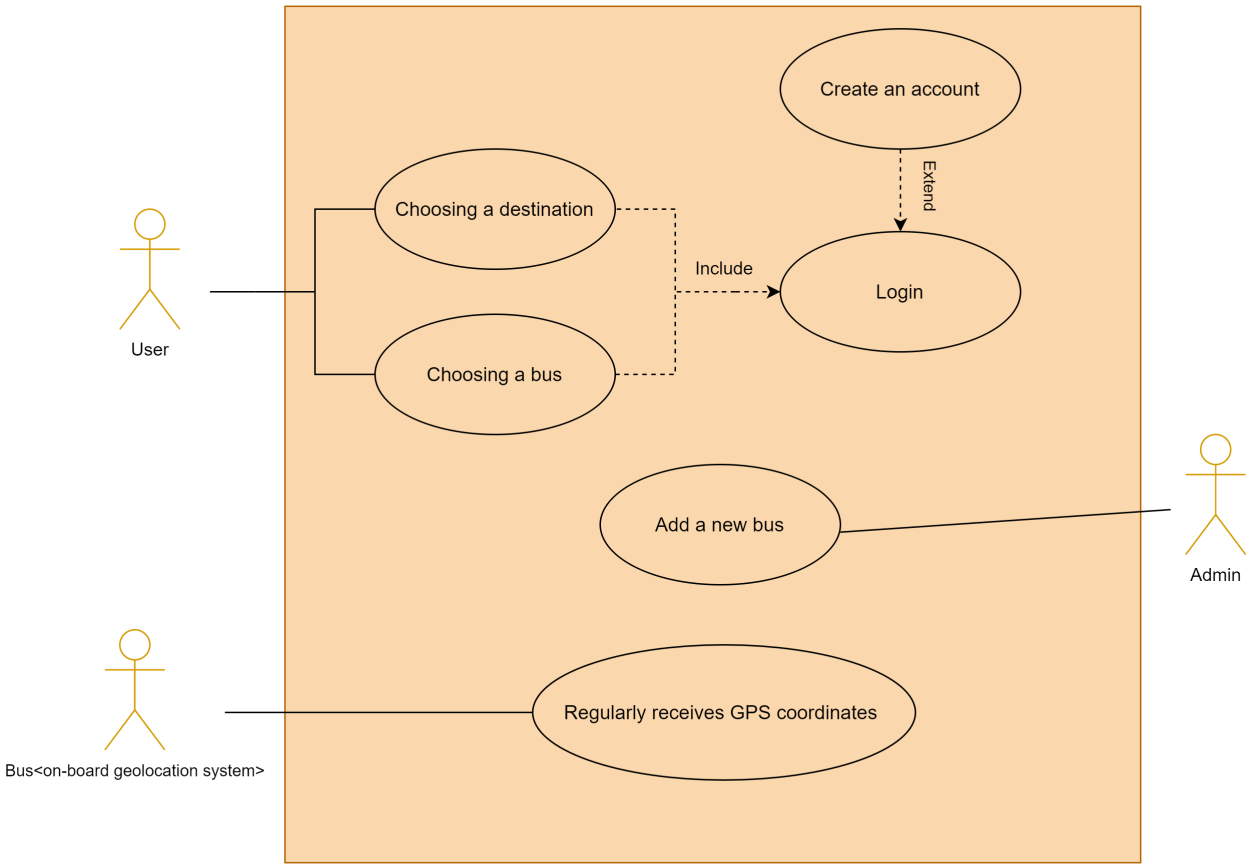

Figure 2.3: Use Case Diagram

2.4.2 Sequence Diagram

A sequence diagram is a type of interaction diagram because it describes how and in what order a group of objects works together .

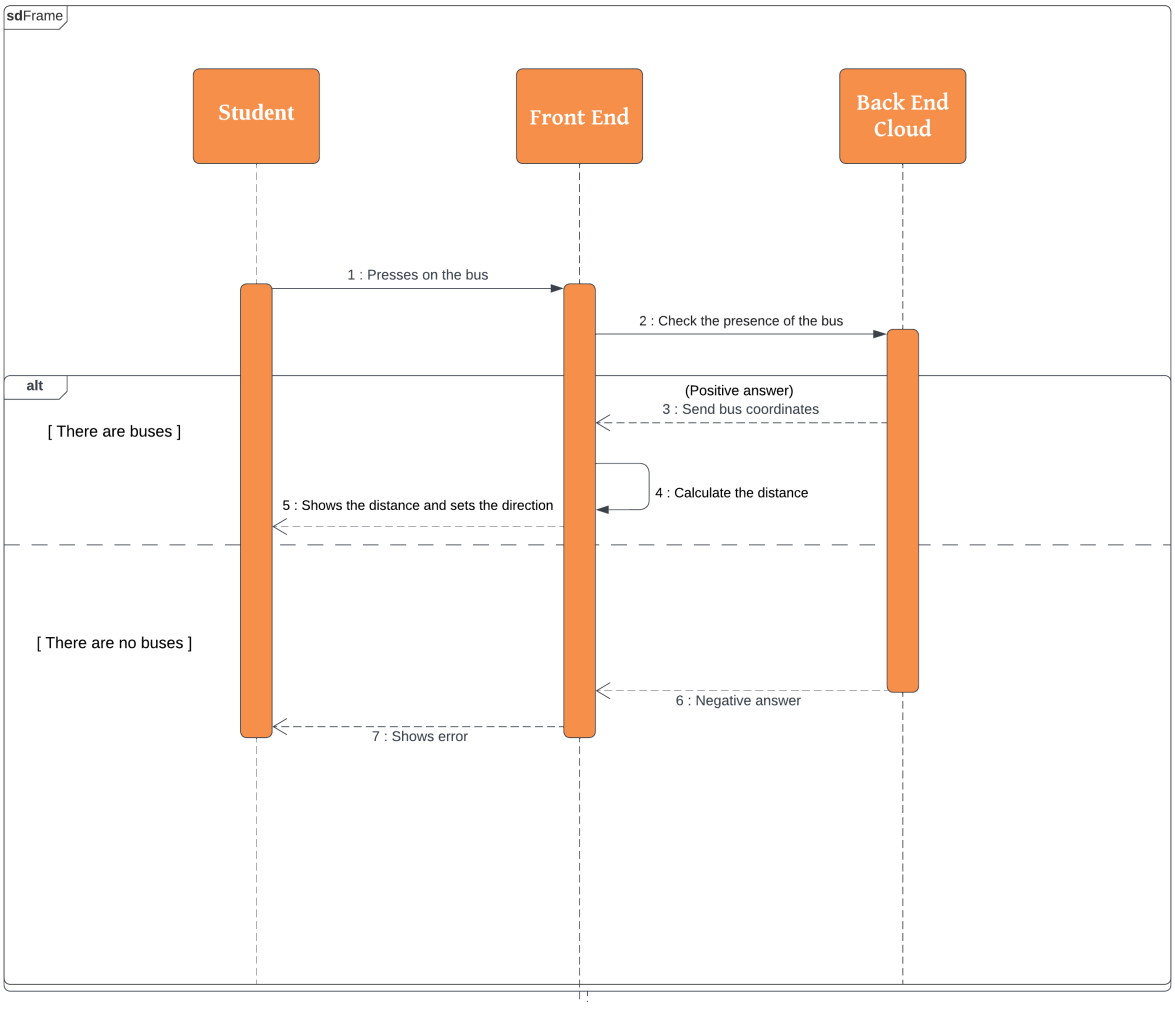

Figure 2.4: Sequence Diagram of the search on the bus

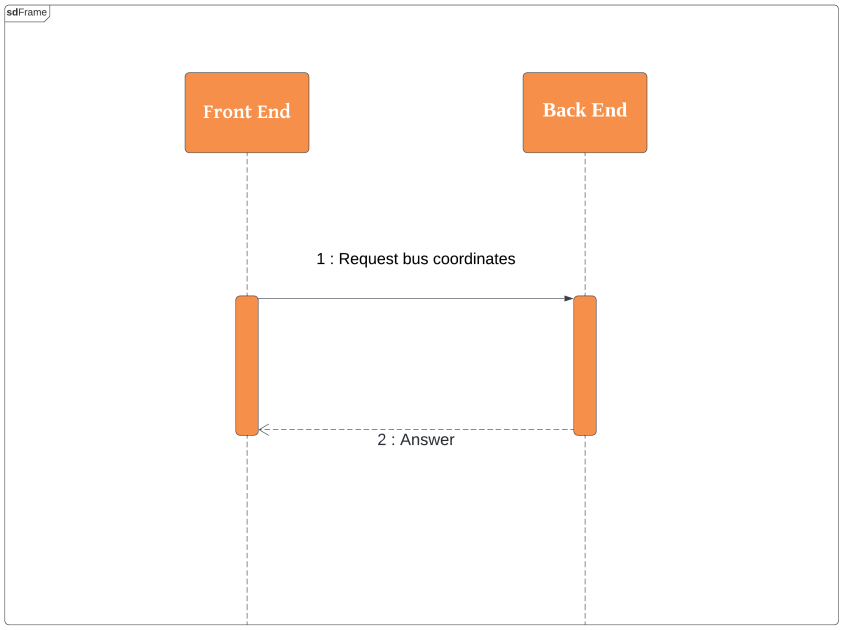

Figure 2.5: Sequence Diagram of the search on the bus

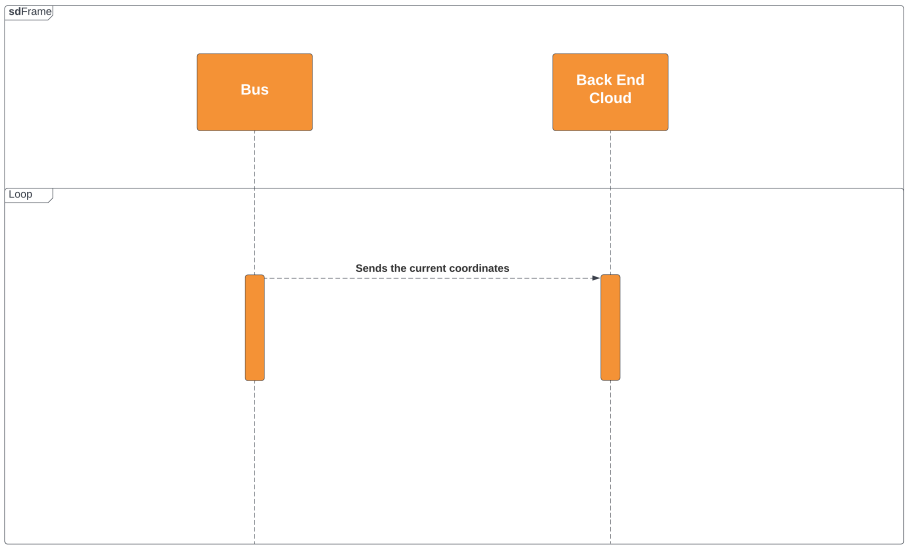

Figure 2.6: Sequence Diagram of sending request to the system

2.4.3 Class Diagram

A class diagram in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's: classes , attributes , operations and the relationships among objects .

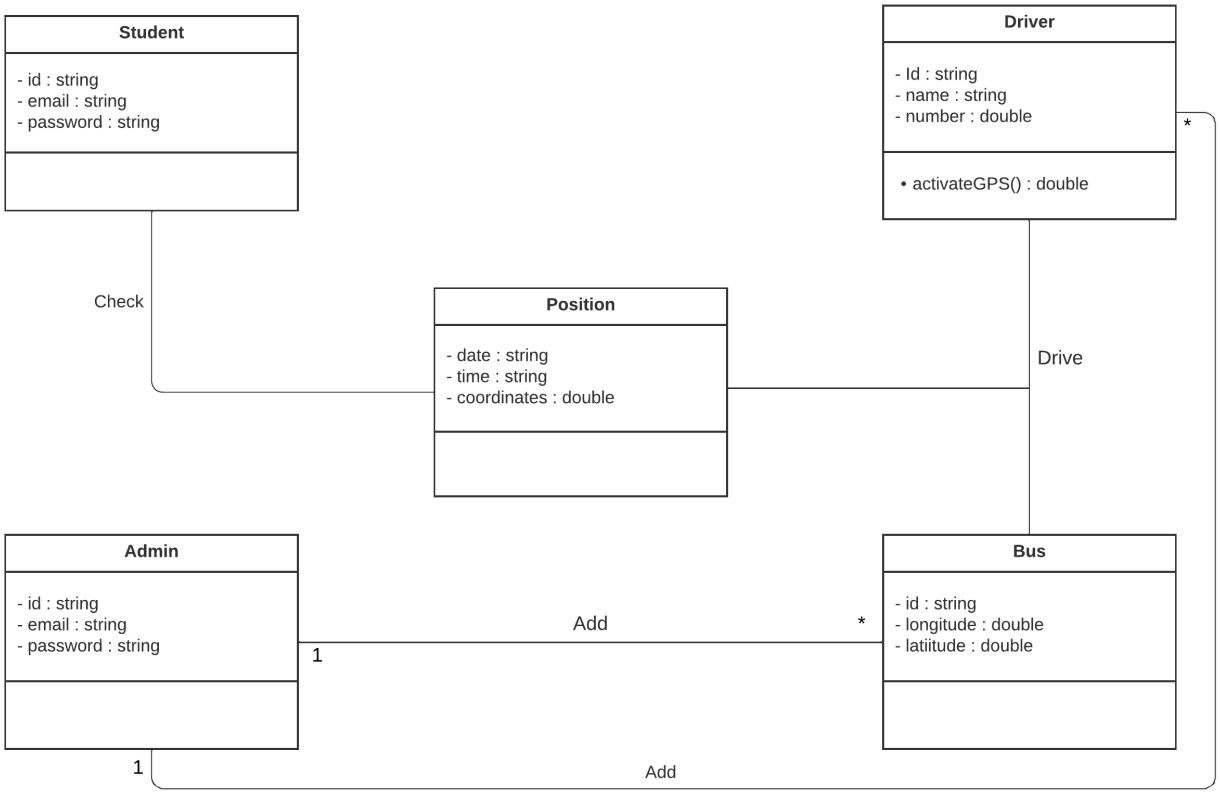

Figure 2.7: Class Diagram

2.5 Conclusion

In this chapter we presented the design phase of our project containing the UML diagrams which helped us to describe the way the system runs in detail in order to facilitate the implementation

In the next chapter , we begin the final phase of our project which is the phase of Design and Implementation .

Chapter 3

Design and Implementation

3.1 Introduction

In this chapter we will first introduce the hardware and software development environment , and then describe the implementation stage based on the display of some user , driver , admin interfaces .

3.2 Working Environment

3.2.1 Software environment

3.2.1.1 The Language

Dart : is a programming language designed for client development , such as for the web and mobile apps . It is developed by Google and can also be used to build server and desktop applications .
Dart It is an object-oriented , class-based , garbage-collected language with C-style syntax .
Dart can compile to either native code or JavaScript . [1]

Figure 3.1: Dart Logo

3.2.1.2 Tools

- **Visual Studio Code :**
  Visual Studio Code (famously known as VS Code) is a free open source text editor by Microsoft .
  VS Code is available for Windows , Linux , and macOS . Although the editor is relatively lightweight , it includes some powerful features that have made VS Code one of the most popular development environment tools in recent times .
  The VS Code user interface allows for a lot of interaction compared to other text editors. To simplify user experience , VS Code is divided into five main regions:
  - The activity bar
  - The side bar
  - Editor groups
  - The panel
  - The status bar [9]

Figure 3.2: Visual Studio Code Logo

- **Flutter :**
  Flutter is an open-source UI software development kit created by Google. It is used to develop cross platform applications for Android , iOS , Linux , macOS , Windows , Google Fuchsia , and the web from a single codebase . Was released in May 2017.[2]

Figure 3.3: Visual Studio Code Logo

- **Firebase :**
  Firebase is a Backend-as-a-Service (Baas) . It provides developers with a variety of tools and services to help them develop quality apps , grow their user base , and earn profit . It is built on Google's infrastructure .
  Firebase is categorized as a NoSQL database program, which stores data in JSON-like documents . [10]

Figure 3.4: Firebase Logo

- **Google Maps :**
  Google Maps is a web mapping platform and consumer application offered by Google. It offers satellite imagery , aerial photography , street maps , 360° interactive panoramic views of streets (Street View) , real-time traffic conditions , and route planning for traveling by foot , car, bike , air (in beta) and public transportation.[3]

Figure 3.5: Google Maps Logo

- **Firestore :** Cloud Firestore is a flexible , scalable database for mobile , web , and server development from Firebase and Google Cloud . Like Firebase Realtime Database , it keeps your data in sync across client apps through realtime listeners and offers offline support for mobile and web so you can build responsive apps that work regardless of network latency or Internet connectivity .
  Cloud Firestore also offers seamless integration with other Firebase and Google Cloud products , including Cloud Functions . [4]

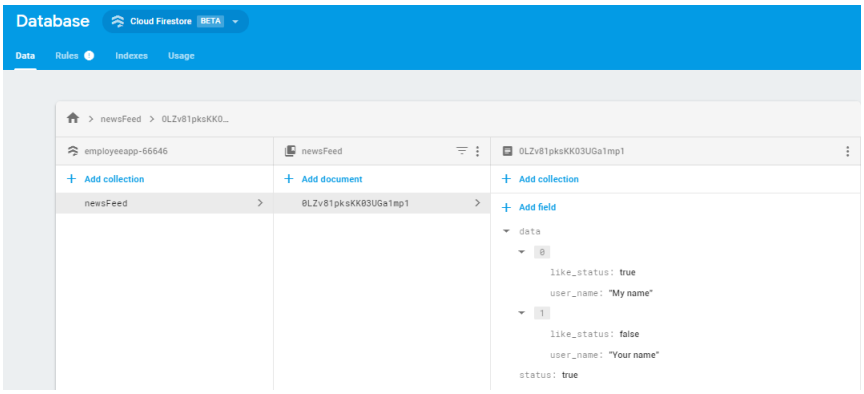

Figure 3.6: Firestore Data Base

3.2.2 Testing environment

- **Emulator :** Ld Player it is a Android simulator that uses virtual simulation technology and can help people experience Android OS on their powerful computers. It is one of the best and lightest Android simulation software for Windows OS . [6]

Figure 3.7: Ld Player Emulator

3.3 Application Description

In this part we will present the interfaces of our application with necessary explanations

3.3.1 User Interface

When the user runs the application , the **Welcoming Interface** will show first . It contains a description and two buttons , one for the Sign Up and the other for Login . If he choose the Sign Up button (Figure 3.9) , it will lead him to another interface to register to the app .
If he choose the Log in button (Figure 3.10) , it will lead him to another interface to login to his account in the app .

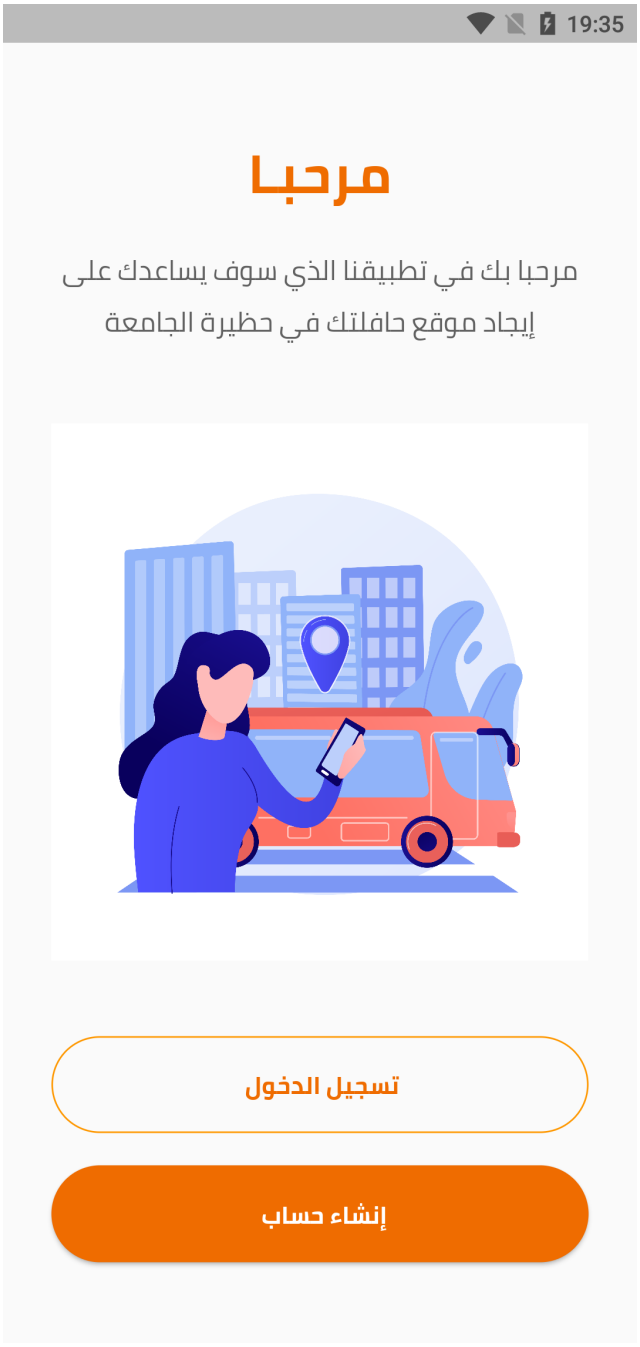

Figure 3.8: Welcoming Interface

The register in the application contains :
Student card number , email and password .
When he/she complete registering , the student must Log in to continue .

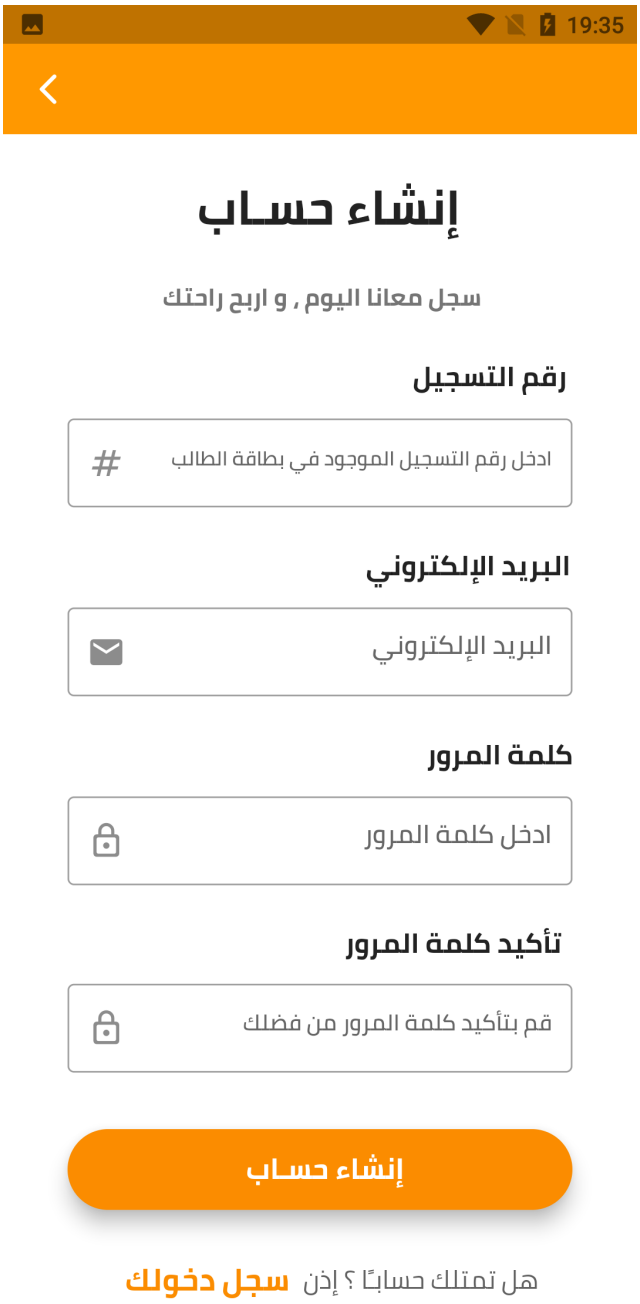

Figure 3.9: Sign Up Interface

When the student press the button to Sign Up , it send to execute the source code below to add new user to the Firebase Cloud (Figure 3.11) .

```
child: MaterialButton(
  minWidth: double.infinity,
  height: 50,
  onPressed: () {
    Map<String, dynamic> map = {
      "cardEtudiant":
          cardEtudiant.text,
      "email": email.text,
      "password": password.text,
    };
    FirebaseFirestore.instance
        .collection("ETUDIANT")
        .add(map);
    Navigator.of(context)
        .pushNamed('login');
```

Figure 3.10: Source Code Sign up

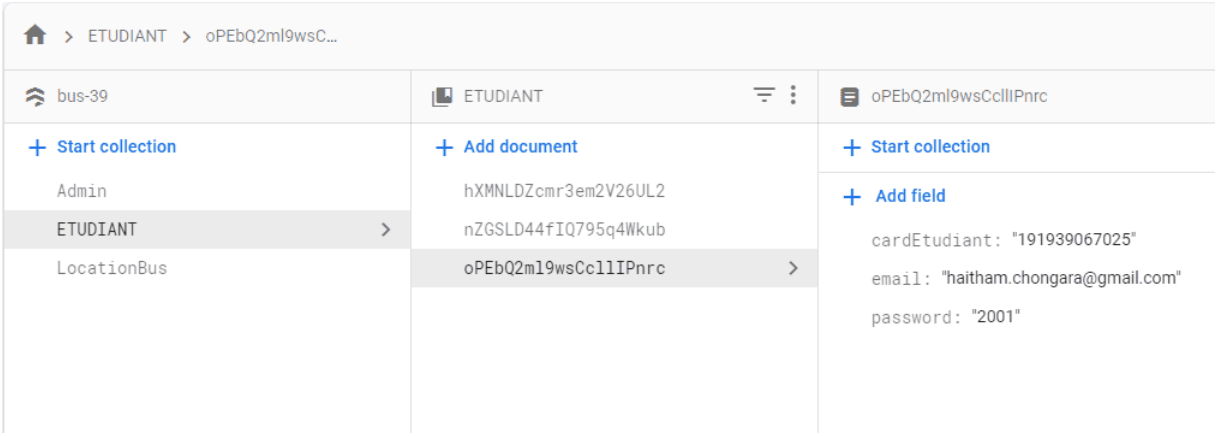

Figure 3.11: Firebase Cloud Sign up

After registration , the student login to his account .
When he/she press the button of Log in , the app start locating the student .

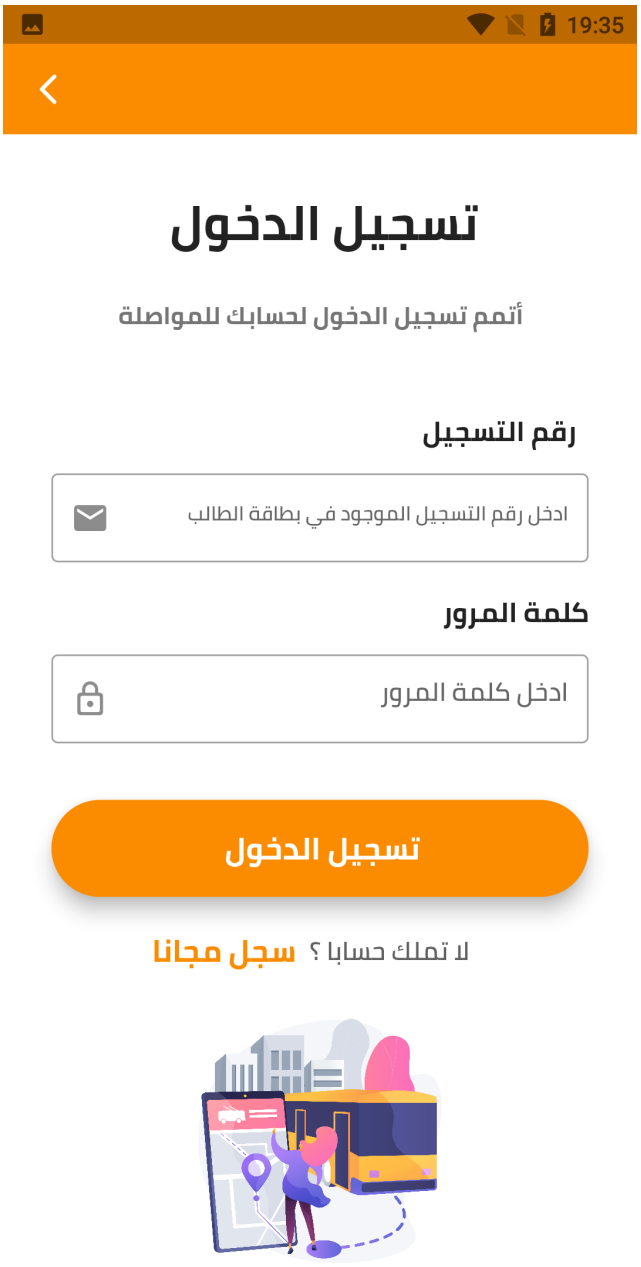

Figure 3.12: Log in Interface

After Log in , it will appear to the student the Home Page which contain buttons to search for his/her bus

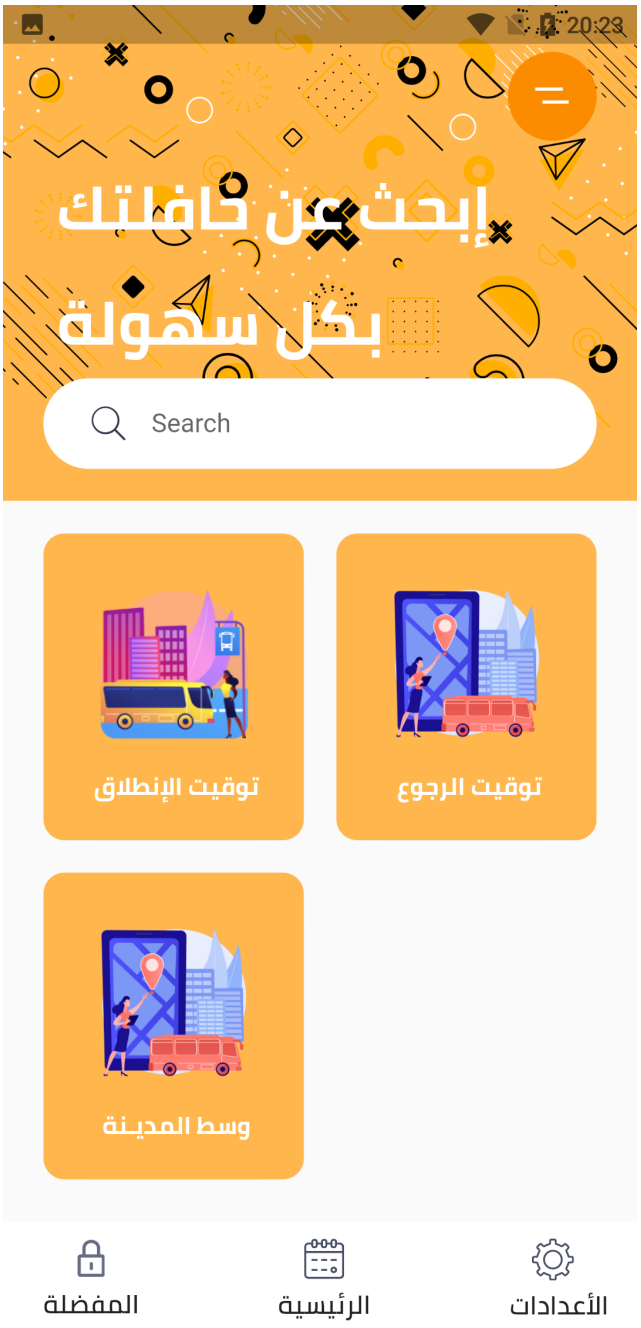

Figure 3.13: Home Page

The objective of our application is in Return Time , so when the student press on the Return time button it will show him another page contains a list of municipalities that the user going to choose one of them .

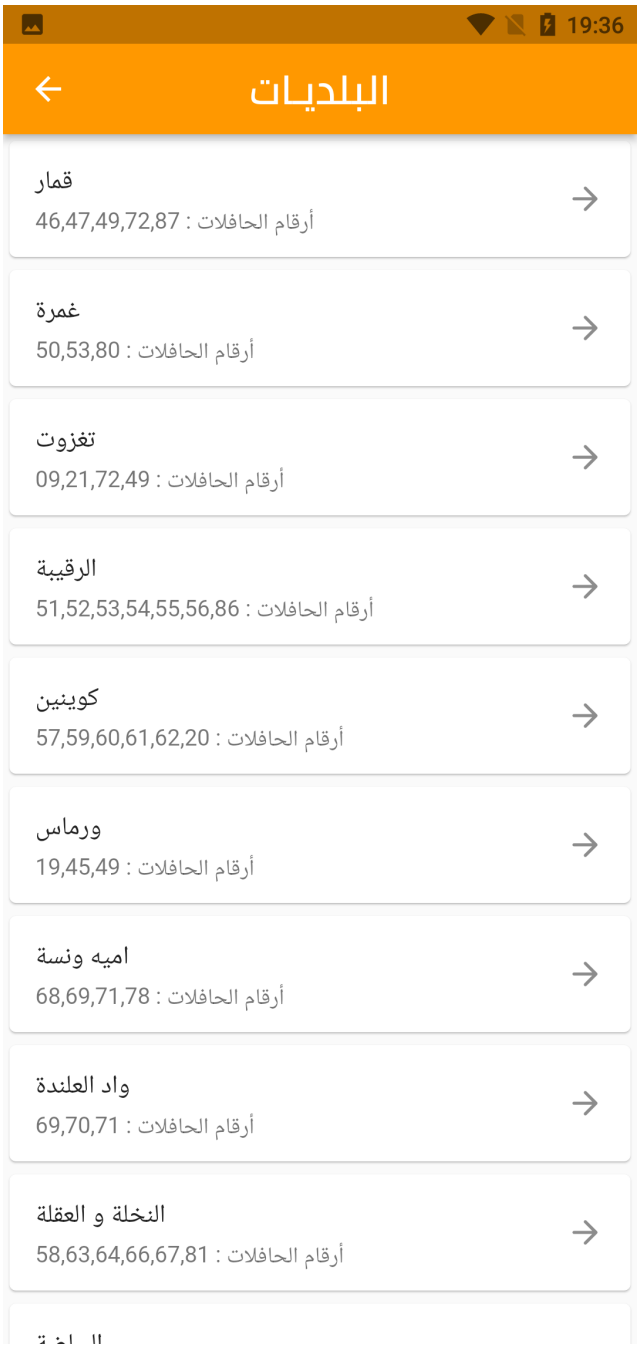

Figure 3.14: Municipalities List Page

When the student choose a municipalities , it will a screen have the bus numbering of this municipalities and the student must choose the one that going to take it . When he choose the bus it will take it to the map to determine the location .

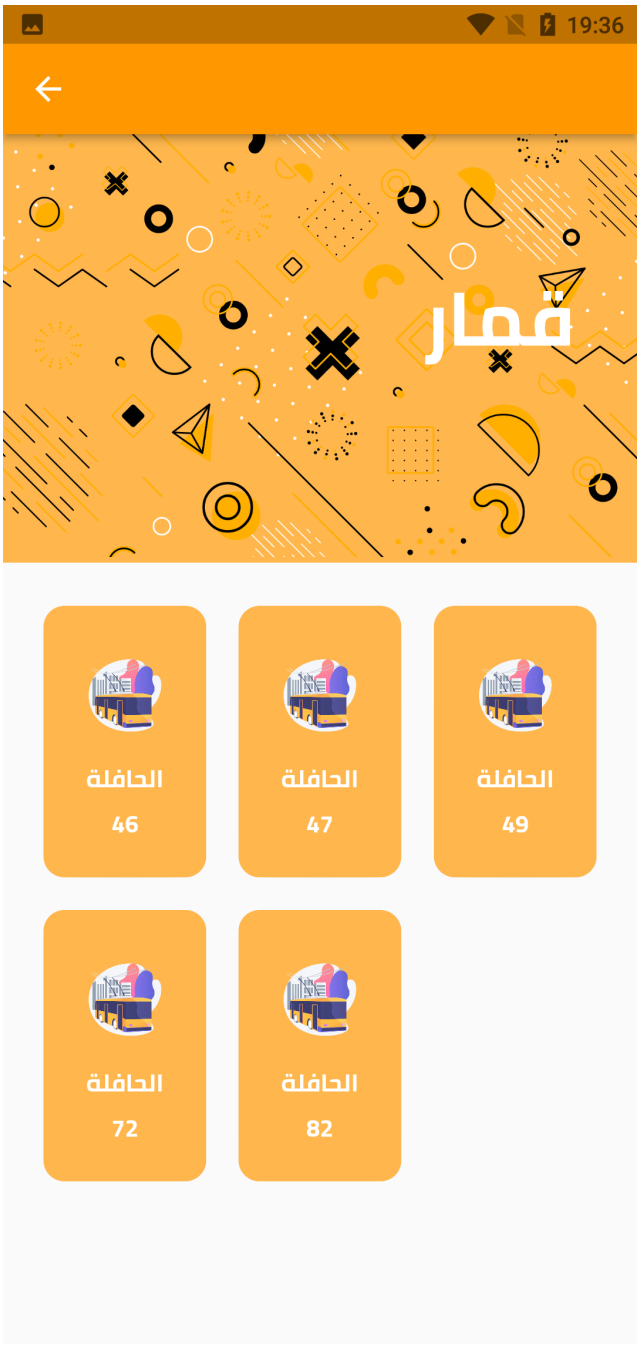

Figure 3.15: Bus Numbering Page of a Municipality

After the student press the bus numbering button , the map page will appear to him and show the user location and bus location .

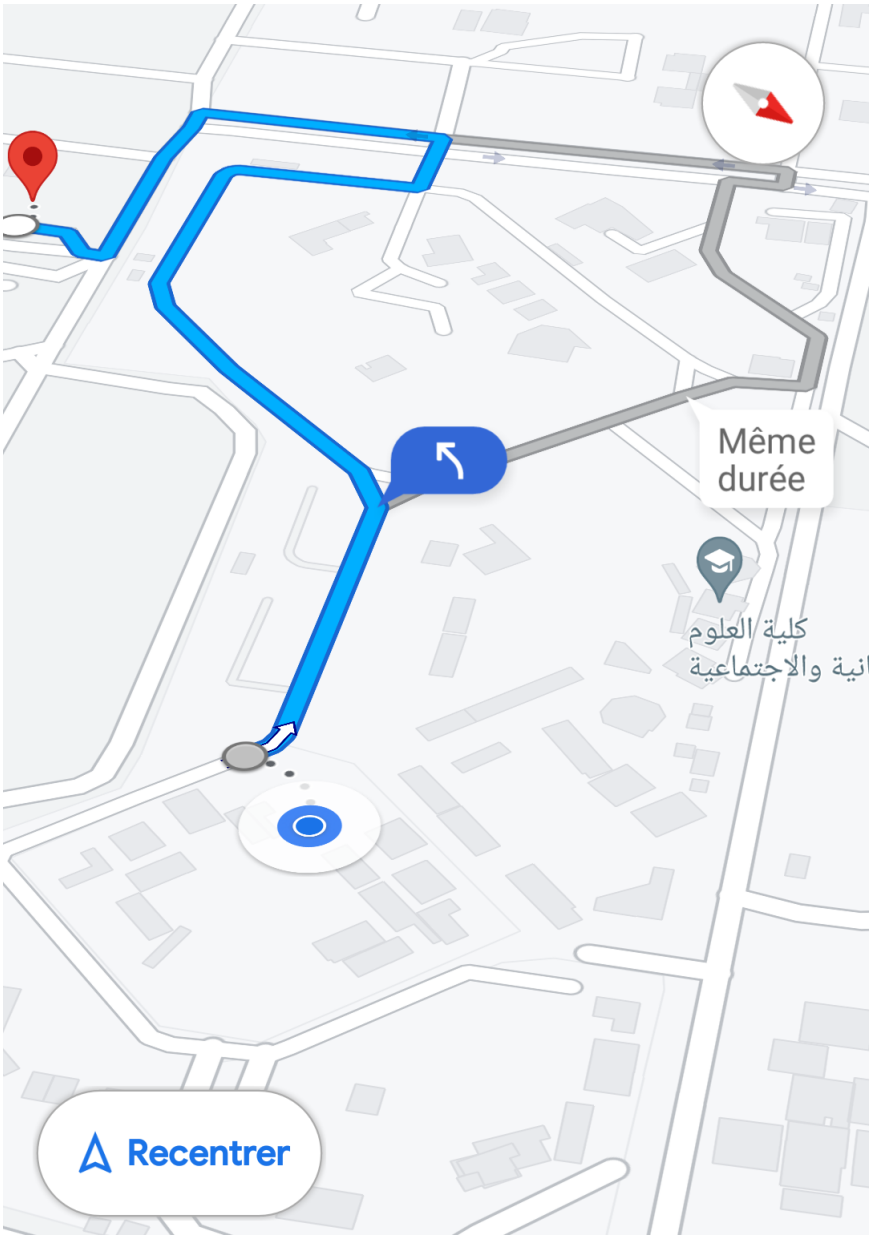

Figure 3.16: Google Maps

3.3.2 Driver Interface

In the other side , the driver of each bus have different program from students program .
The Driver application have buttons to start and stop the GPS , which means when he enable the GPS the app starts sending the current coordinates of the bus and if he stops it the coordinates will never be sent , and this is the main function to find the bus location .

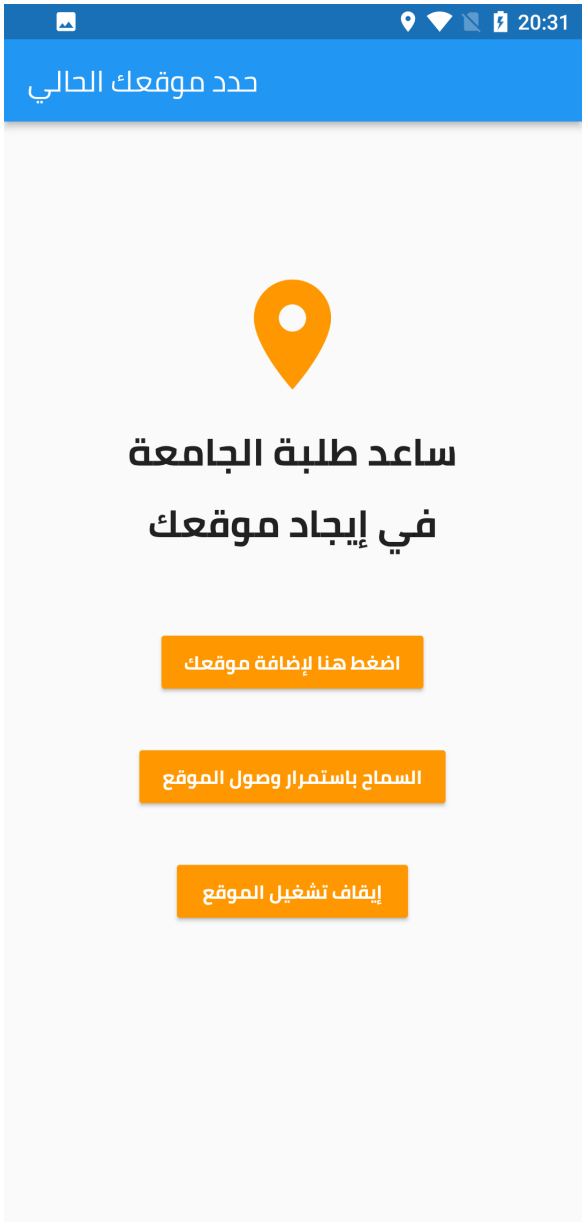

Figure 3.17: The Driver Application

The source code of the driver application :

```
_getLocation() async {
  try {
    final loc.LocationData _locationResult = await location.getLocation();
    await FirebaseFirestore.instance
        .collection('Location')
        .doc('Latlon')
        .set({
      'latitude': _locationResult.latitude,
      'longitude': _locationResult.longitude,
    }, SetOptions(merge: true));
  } catch (e) {
    print(e);
  }
}

Future<void> _listenLocation() async {
  _locationSubscription = location.onLocationChanged.handleError((onError) {
    print(onError);
    _locationSubscription?.cancel();
    setState(() {
      _locationSubscription = null;
    });
  }).listen((loc.LocationData currentlocation) async {
    await FirebaseFirestore.instance
        .collection('LocationBus')
        .doc('Latlon')
        .set({
      'latitude': currentlocation.latitude,
      'longitude': currentlocation.longitude,
    }, SetOptions(merge: true));
  });
}

_stopListening() {
  _locationSubscription?.cancel();
  setState(() {
    _locationSubscription = null;
  });
}
```

Figure 3.18: Source Code Driver Application

And this it is the Firestore Cloud , when the driver enable GPS , the coordinates are stored in LocationBus Document .

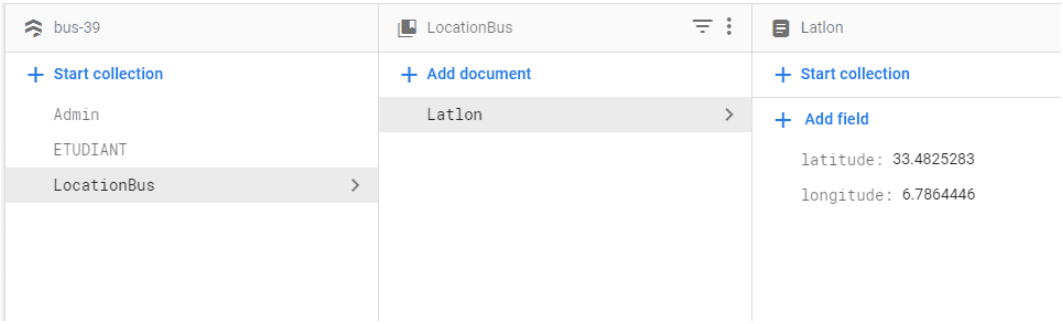

Figure 3.19: Source Code Driver Application

3.3.3 Admin Interface

The Admin can add new bus from a custom application for him .
The first screen is a welcoming screen have a button to start .

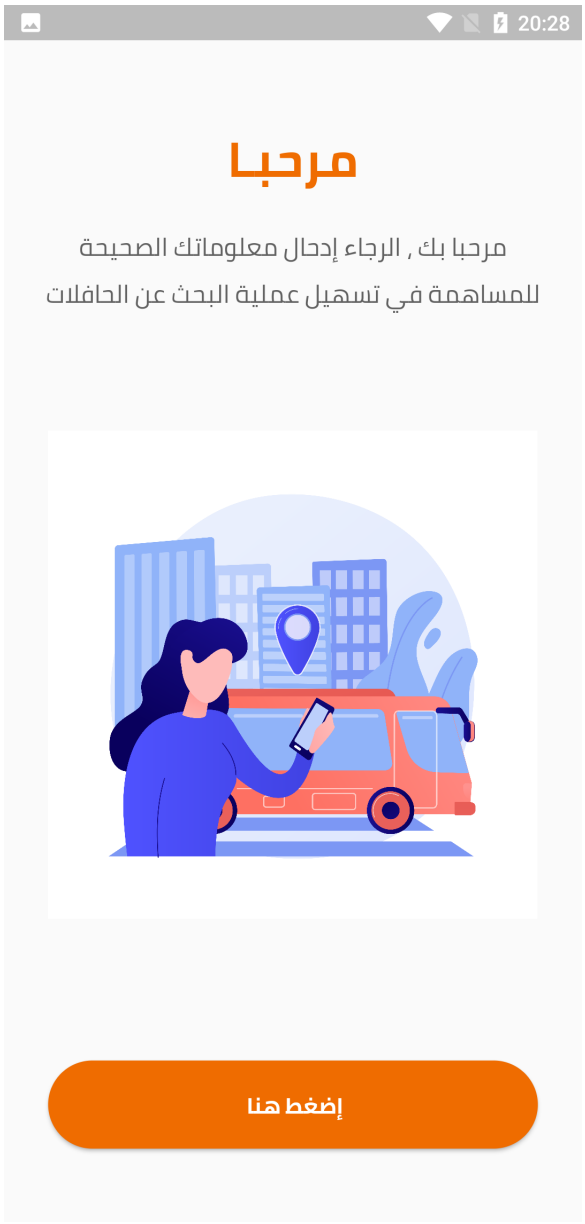

Figure 3.20: The Admin Application

To insert a new bus , the admin must enter the bus number and its destination and the phone number of the driver .

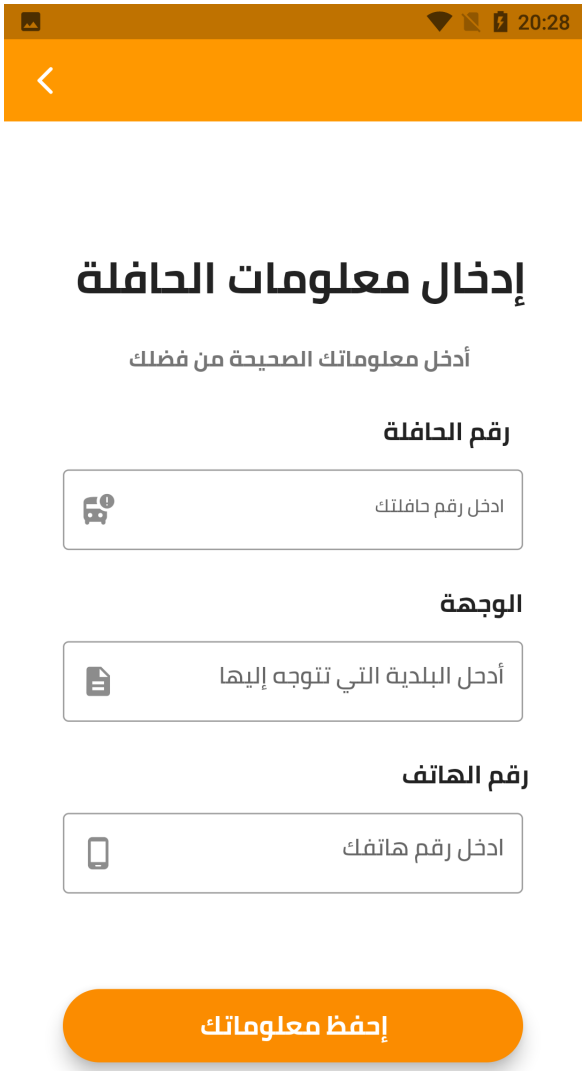

Figure 3.21: Insert a new bus

When the admin press the button to save the information , the app execute this code for adding a new bus in Firestore as indicated in (Figure 3.22) .

```
height: 50,
onPressed: () {
  Map<String, dynamic> map = {
    "busNumber": busNumber.text,
    "destination": destination.text,
    "phone": phone.text,
  };
  FirebaseFirestore.instance
      .collection("Bus")
      .add(map);
  Navigator.of(context)
      .pushNamed('login');
},
color: Colors.orange[600],
elevation: 10,
shape: RoundedRectangleBorder(
  borderRadius:
      BorderRadius.circular(50),
), // RoundedRectangleBorder
child: Text("إحفظ معلوماتك",
    style: TextStyle(
```

Figure 3.22: Source Code Adding Bus

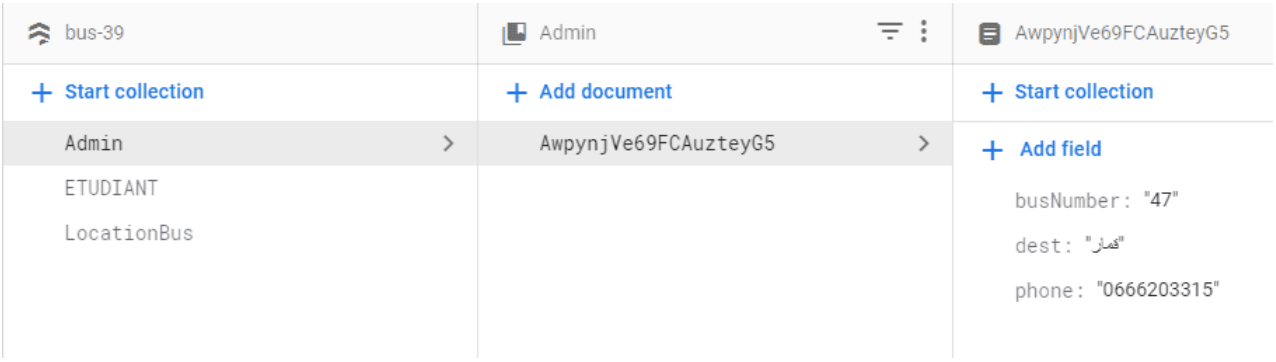

Figure 3.23: Cloud Firestore Admin add a new bus

The system can use the coordinates of the bus by using this code , when it is executed the system will have the current coordinates and the correct location of the bus in the park .

```
return StreamBuilder<QuerySnapshot>(
    stream:
        FirebaseFirestore.instance.collection("LocationBus").snapshots(),
    builder: (BuildContext context, AsyncSnapshot<QuerySnapshot> snapshot) {
      if (!snapshot.hasData) {
        return Text("No Value");
      }
      if (snapshot.connectionState == ConnectionState.waiting) {
        return Column(
          mainAxisAlignment: MainAxisAlignment.center,
          crossAxisAlignment: CrossAxisAlignment.center,
          children: [const CircularProgressIndicator()],
        ); // Column
      }
      snapshot.data?.docs.forEach((element) {
        latitude = element["latitude"];
        longitude = element["longitude"];
      });
```

Figure 3.24: Source Code sending coordinates to the system

3.4 Conclusion

This phase is the most important stage in the life cycle of the application, in this chapter we briefly describe the process of creating our application by identifying the environment , tools and development languages associated with our system .

General Conclusion

Faced with the importance of helping students to take the bus, we studied , designed and realized through this work a mobile application that allows passengers to know the bus location at the park .

We first presented the problem and the proposed solution . Then we presented the concepts phase in which the functionalities of our application were described .In the end we implemetate this work in order to implement the proposed solution.

Although our application is not yet finished , we intend to complete and improve it in terms of design (ergonomics), and increase functionality and add other languages to the app to attract a large number of users and deploy it to the Play store

Bibliography

[1] https://en.wikipedia.org/wiki/Dart$_{(programming_language)}$. *Dart. Accessed on June* 01, 2022.

[2] https://en.wikipedia.org/wiki/Flutter$_{(software)}$. *Flutter. Accessed on June* 01, 2022.

[3] https://en.wikipedia.org/wiki/Google$_{Maps}$. *Google Maps. Accessed on June* 01, 2022.

[4] https://firebase.google.com/docs/firestore. *Cloud Firestore*. Accessed on June 01 , 2022.

[5] https://www.geeksforgeeks.org/unified-modeling-language-uml introduction/. *Unified Modeling Language (UML) — An Introduction*, 2019. Accessed on April 25 , 2022.

[6] https://www.ldplayer.net/blog/what-is-ldplayer-android emulator.html. *What is LDPlayer Android Emulator?* Accessed on June 01 , 2022.

[7] https://www.techopedia.com/definition/3243/unified-modeling-language uml. *Unified Modeling Language (UML)*, 2012. Accessed on June 01 , 2022.

[8] https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is uml/. *What is Unified Modeling Language (UML)?* Accessed on April 25 , 2022.

[9] Anusheh Zohair Mustafeez. *What is Visual Studio Code?* https://www.educative.io/edpresso/what-is-visual-studio-code, note = Accessed on June 01 , 2022 .

[10] Edpresso Team. *What is Firebase?* https://www.educative.io/edpresso/what-is-firebase.