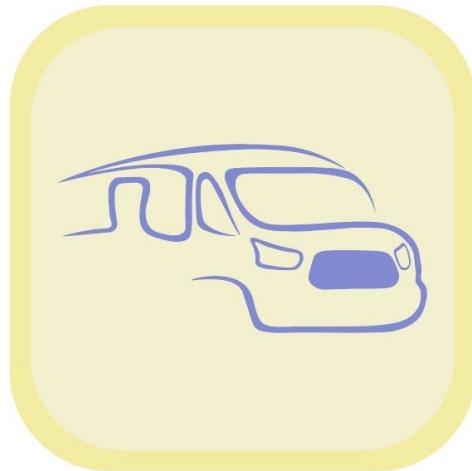


# Gebze Technical University

CSE 343 SOFTWARE ENGINEERING

V E G A



Prepared by

- Oğuz Mutlu
- Ahmet Yazıcı
- Yunus Emre Yumşak
  - Eren Çakar
  - Mehmet Acar

Github : <https://github.com/adenu/Project-Vega>

## Contents

1.User Story and Use Case Scenario.....	5
1.1 User Story.....	5
1.2 Use Case Scenarios.....	7
2.PROJECT BACKLOG DOCUMENTATION .....	9
3.UML DIAGRAMS.....	10
<b>3.1 Context Model of Vega Shuttle App.....</b>	<b>10</b>
<b>3.2 Process Model.....</b>	<b>11</b>
3.2.1 User Sign Up Process.....	11
3.2.2 Creation of Shuttle Process.....	11
3.2.3 Linking Child Profile and Add Shuttle.....	12
3.2.4 Display Shuttle and Travel Information.....	13
<b>3.3 Vega Shuttle App Use Case Diagram.....</b>	<b>14</b>
3.3.1 Detailed Information of Use-Case.....	15
3.4. Sequence Diagrams .....	25
<b>3.4.1 Child Route Subscription .....</b>	<b>25</b>
<b>3.4.2 Connection Request .....</b>	<b>26</b>
<b>3.4.3 Create Child Sequence.....</b>	<b>27</b>
<b>3.4.4 Create Shuttle Sequence.....</b>	<b>28</b>
<b>3.4.5 Shuttle Employee Request .....</b>	<b>29</b>
<b>3.4.6 Subscribe Route Sequence.....</b>	<b>30</b>
<b>3.4.7 Show Shuttle Information .....</b>	<b>30</b>
<b>3.4.8 Employee Leave Sequence .....</b>	<b>31</b>
<b>3.4.9 Update Passenger Status .....</b>	<b>31</b>
3.5 Class Diagrams.....	32
3.5.1 Classes and Associations .....	32
3.5.2 Generalization Hierarchy .....	33
3.5.3 Class Diagram.....	34
3.6 State Diagrams .....	35
3.6.1 State Diagram .....	35
3.6.2 Profile Page State Diagram.....	36
3.6.3 My Shuttle Page State Diagram.....	36

3.6.4 Linked Profiles Page State Diagram.....	37
3.6.5 Settings Page State Diagram .....	37
4. Simple System Architecture Document .....	38
4.1. Introduction.....	38
4.1.1 Purpose.....	38
4.2 Architectural Representation.....	38
4.2.1. Use Case view .....	39
4.2.2 Logical View .....	40
4.2.3 Deployment View .....	41
Software Development Life Cycle.....	42
5.Working Graphical Interface.....	45
6.Test Cases.....	55
6.1. Test Case ( <i>Signing Up Successfully</i> ).....	55
6.2. Test Case ( <i>Invalid E-Mail Registration</i> ) .....	56
6.3. Test Case ( <i>Already in Use E-Mail Registration</i> ).....	57
6.4. Test Case ( <i>Empty E-Mail Registration</i> ) .....	58
6.5. Test Case ( <i>Weak Password Registration</i> ).....	59
6.6. Test Case ( <i>Null/Empty Password Registration</i> ).....	60
6.7. Test Case ( <i>Doesn't Match Password Registration (NULL)</i> ).....	61
6.8. Test Case ( <i>Doesn't Match Password Registration</i> ).....	62
6.9. Test Case ( <i>Visibility On/Off icon Test </i> ) .....	63
6.10. Test Case ( <i>Null/Empty Name Entrance Registration</i> ).....	64
6.11. Test Case ( <i>Null/Empty Surname Entrance Registration</i> ) .....	65
6.12. Test Case ( <i>Empty/Null Mail-Password Entrance Login</i> ).....	66
6.13. Test Case ( <i>Empty/Null Email Entrance Login</i> ).....	67
6.14. Test Case ( <i>Empty/Null Password Entrance Login</i> ) .....	68
6.15. Test Case ( <i>Invalid Password Login</i> ).....	69
6.16. Test Case ( <i>Invalid Email Login </i> ) .....	70
6.17. Test Case ( <i>Reset Password </i> ).....	71
6.18. Test Case ( <i>Clicking Same Password Reset Link </i> ) .....	72
6.19. Test Case ( <i>Validate Account </i> ) .....	73
6.20. Test Case ( <i>Shuttle Live Location Tracking </i> ) .....	74

<b>6.21. Test Case (Set Usage Status) .....</b>	75
<b>6.22. Test Case (Already been Used Verification Link ).....</b>	76
<b>6.23. Test Case (Changing Theme and Notifications ) .....</b>	77
<b>6.24. Test Case (Changing Username).....</b>	78
<b>6.25. Test Case (Create Child Profile and Subscribe Route).....</b>	79
<b>6.26. Test Case (Create Child Profile and Subscribe Route Wrong Route Code).....</b>	80
<b>6.27. Test Case (Create Child Profile and Subscribe Route more than one child).....</b>	81
<b>6.28. Test Case (Create Shuttle Enter Wrong Plate).....</b>	82
<b>6.29. Test Case (Create Shuttle Enter Wrong Seat) .....</b>	83
<b>6.30. Test Case (Create Shuttle Enter Wrong Plate and Seat) .....</b>	84
<b>6.31. Test Case (Create Shuttle Succesfully) .....</b>	85
<b>6.32. Test Case (Enter One Route).....</b>	86
<b>6.33. Test Case (Enter more than one Route) .....</b>	87
<b>6.34. Test Case (Shuttle's Live Location).....</b>	88

# 1. User Story and Use Case Scenario

Vega is the app that provides safe shuttle bus system for families and all kind of users. You can keep track of live location, speed etc. of your shuttle. You can also get informed about activities of your child such as getting off and getting into the shuttle.

## 1.1 User Story

Sinem launches the application and signs up for the first time by entering required information such as name, email etc. Then creates two child profiles for both her 5 years old daughter Elif and 14 years old son Hayri by using create child profile button in the connected profiles screen. Hayri has a phone and wants to use the app too. Hayri also downloads the app and registers with his own mail. Sinem adds his profile as a child. Then she wants to learn the codes for both of her kids in order to register them to their shuttles. She contacts the shuttle employee and asks him the code for her daughter's shuttle. He messages the code and Sinem copies it and re-enters the app. Then she goes to child profiles and clicks on her daughter's profile. There she clicks the add shuffle button and enters the code. After that she contacts the other shuttle employee for Hayri. He tells her that he did not register to the app yet and says "I will immediately register and send you the code". Then he launches the app and does the same as Sinem, enters his name and e-mail. When he comes to the main menu, he goes to My Profile page->Settings-> Upgrade my profile as Shuttle Employee. After that he goes to My Shuttles screen and creates a shuttle by entering the required information such as plate of the shuttle, contracted institution etc. After that he got the code for the shuttle and sends it to Sinem. Sinem does the same as before and adds the shuffle to her son's profile.

Sometimes later Hayri and Elif prepares for school and waits for their shuttles. Sinem gets a notification that says Elif's shuttle is 5 minutes away. She takes Elif to outside and helps her to get on the shuttle. Shuttle employee marks Elif as 'inside' from his phone. Sinem now can check it from her phone. Sinem gets back home and Hayri says that he wants to go to the cinema with his friends after school and doesn't want to use the shuttle back to home. Sinem permits that and informs him that he needs to declare that from the app, so the shuttle does not wait for him. His shuttle also comes, and he gets on and goes to school. Sinem than checks for both of her children's live location from the app, for Hayri the app says the estimated time of arrival is 35 minutes, for Elif it says 20 minutes. When they reach to their schools, both Shuttle employees mark them as "got off". Sinem gets notifications for both of her kids when they got off the shuttle.

## 1.2 Use Case Scenarios

### 1-Sign logs into the application

1. If the user is not registered with the application, he/she registers with his/her name, e-mail etc.
2. If the user is registered and logged out before, he/she logs into the application with his/her e-mail address and password.

2- Parent can create child profiles by using create child profile button in the connected profiles screen.

3- After shuttle employee registers to application, he/she can upgrade his/her profile with My Profile page->Settings-> Upgrade my profile as Shuttle Employee steps.

4- Shuttle employee can create a shuttle by entering the required information such as plate of the shuttle, contracted institution etc. with going to My Shuttles screen. After creating shuttle, shuttle employee gets the code for the shuttle.

5-Parent can register his/her kids to shuttles.

1. If parent hasn't got code of shuttle, he/she should contact with the service employee.
2. If parent has got code of the shuttle, he/she goes to child profiles and clicks on his/her child profile. Then, he/she clicks add shuffle button and enters the code.

6- Shuttle employee can mark child as 'inside' when child gets on the shuttle.

7- Shuttle employee can mark child as 'offside' when child gets off the shuttle.

**8-** Parent gets notification when his/her child gets on the shuttle and gets off the shuttle.

**9-** Parent can check his/her child's live location.

**10-** Parent can check his/her child's estimated time of arrival.

**11-** When child does not use shuttle when going to school or back to home, if he/she has got the phone and registered to the application, he/she can declare that from the app. If child has not got the phone, his/her parent can do same thing from the app.

## 2.PROJECT BACKLOG DOCUMENTATION

High Priority

As a user, I can see brief information about connected profiles and, my and their shuttles in main page so I'm informed instantly upon launching the app

As an authorized user, I can set passengers' profiles as on the shuttle or not on the shuttle so shuttle's capacity status and their status stay current

As a user, I can create profiles for my children who don't own mobile phones so that I can check on their situation and safety

As a user, I can connect to my children's profiles so that I can check on their situation and safety

As a user, I can connect myself and my connected profiles to shuttles so I can know about the connected shuttles

As a user, I can upgrade my account to authorized user so I can use app as authorized user

As an authorized user, I can create a new shuttle whenever a new shuttle is in service

As an authorized user, I can change the route of my shuttles as I see fit

As an authorized user, I can switch to user so I can use the app as user

As a user, I can change my user password if it's forgotten

As a user, I can see my profile so I can check my information

As a user, I can update my user information so it stays current

As a user, I can see my subscribed shuttle's location on the map in real time

As a user, I can see my connected profiles so I can check their information

As a user, I can update my connected and created profiles' information so they stay current

As an authorized user, I can update my shuttles' information so they stay current

As a user, I can navigate to anywhere directly from main page for convenience

As a user, I can navigate to any page from any page as every page has a page redirection menu that is opened as swiping from the side of the screen

As a user, I can see routes of my shuttles on map so I know where I get on and off

As a user, I can navigate to any of my shuttles' page so I can get detailed information about it

As a user, I can turn on notifications for status change of my connected profiles so I can be sure they arrived safely

As a user, I can turn on notifications for whenever any of my shuttles arrives so I don't miss any shuttle

As a user, I can choose to remain logged in for convenience

As a user, I can change the theme of the app as I please so it is easier on my eyes

Low Priority

## 3.UML DIAGRAMS

### 3.1 Context Model of Vega Shuttle App

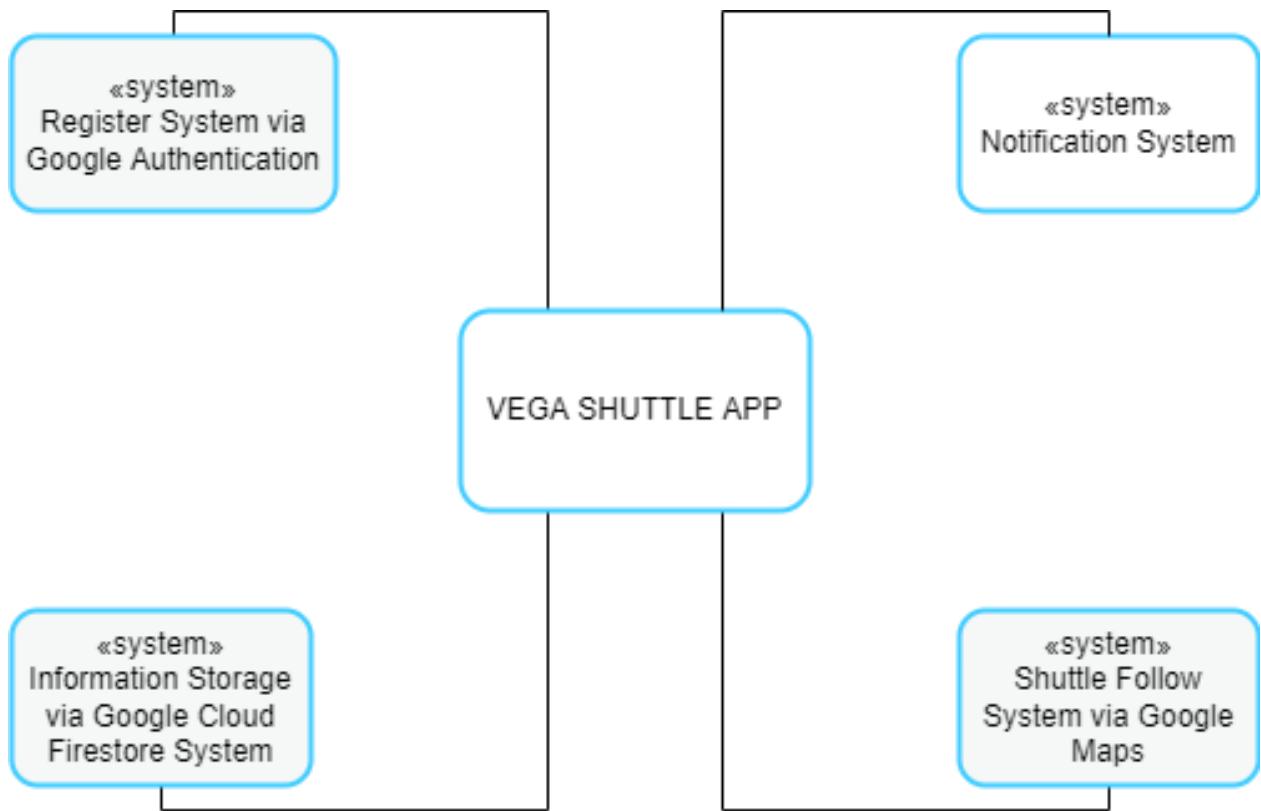
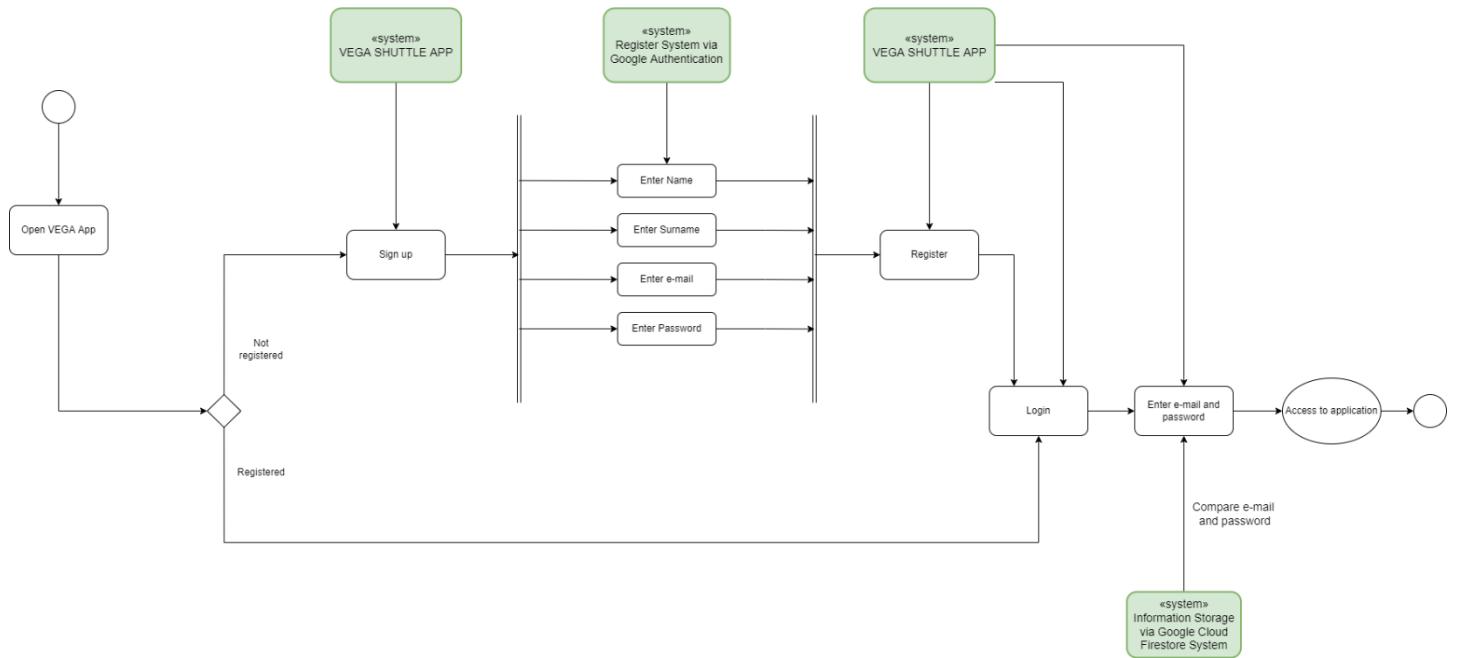


Figure 1 Context Model of Vega App

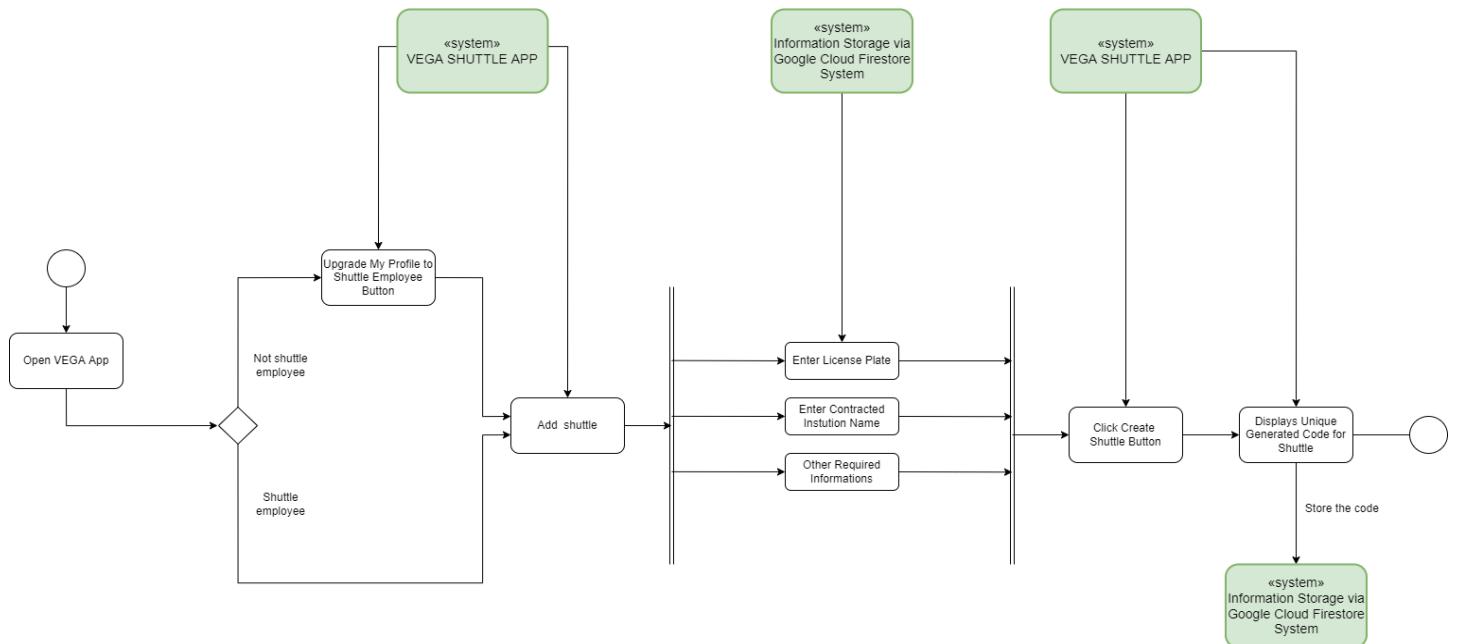
## 3.2 Process Model

### 3.2.1 User Sign Up Process



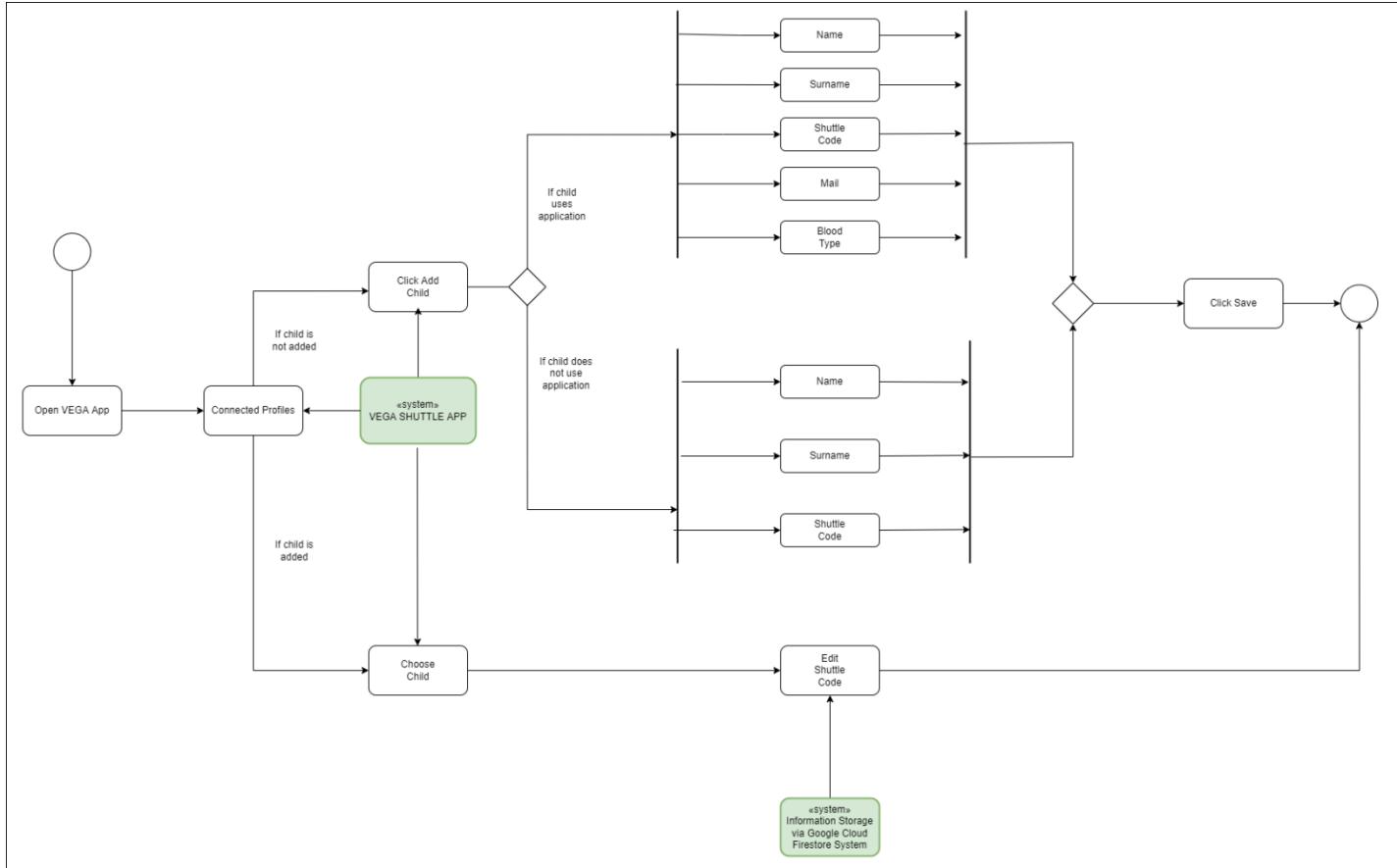
**Figure 2** Sign Up – Login Process Diagram

### 3.2.2 Creation of Shuttle Process



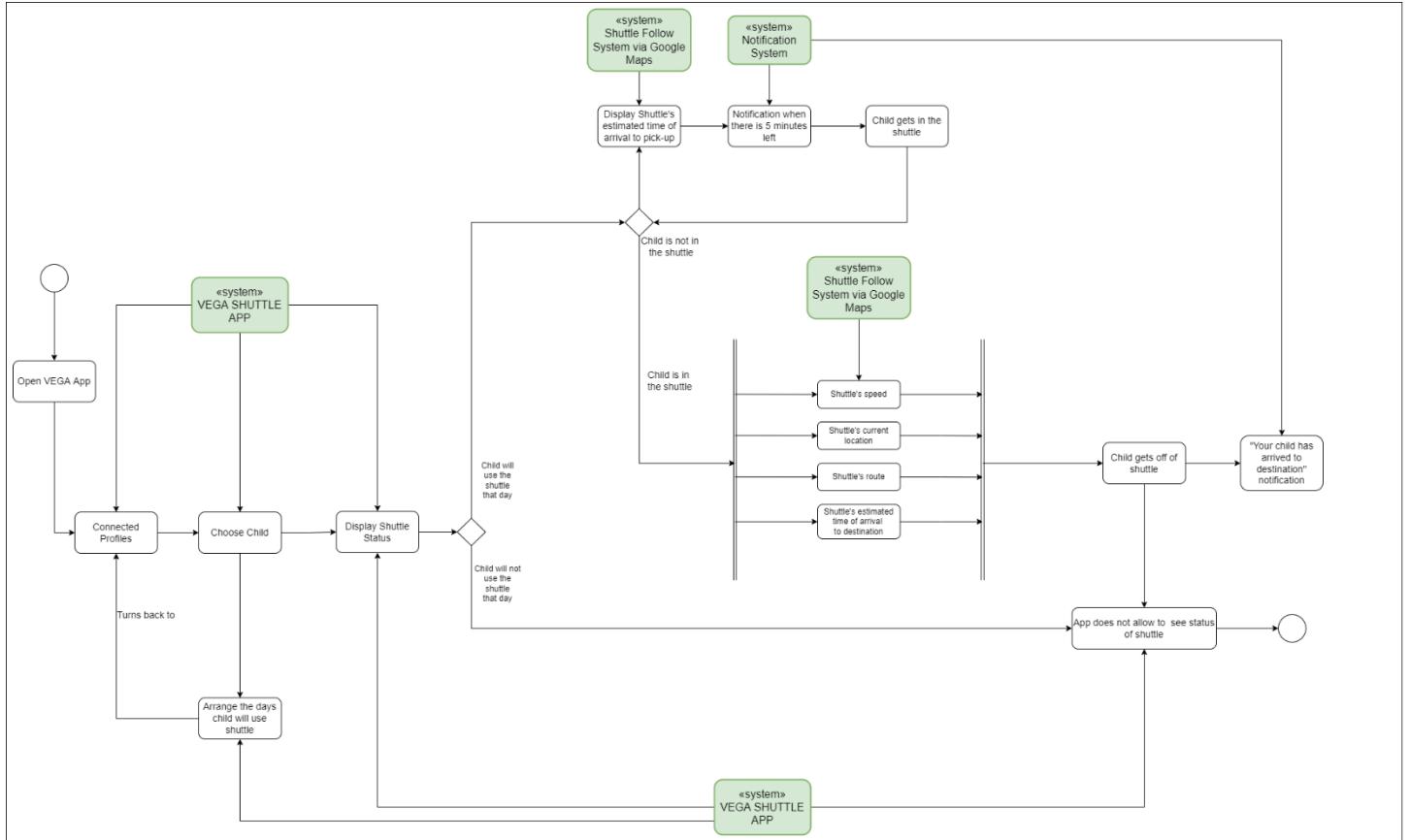
**Figure 3** Creation of Shuttle Diagram

### 3.2.3 Linking Child Profile and Add Shuttle



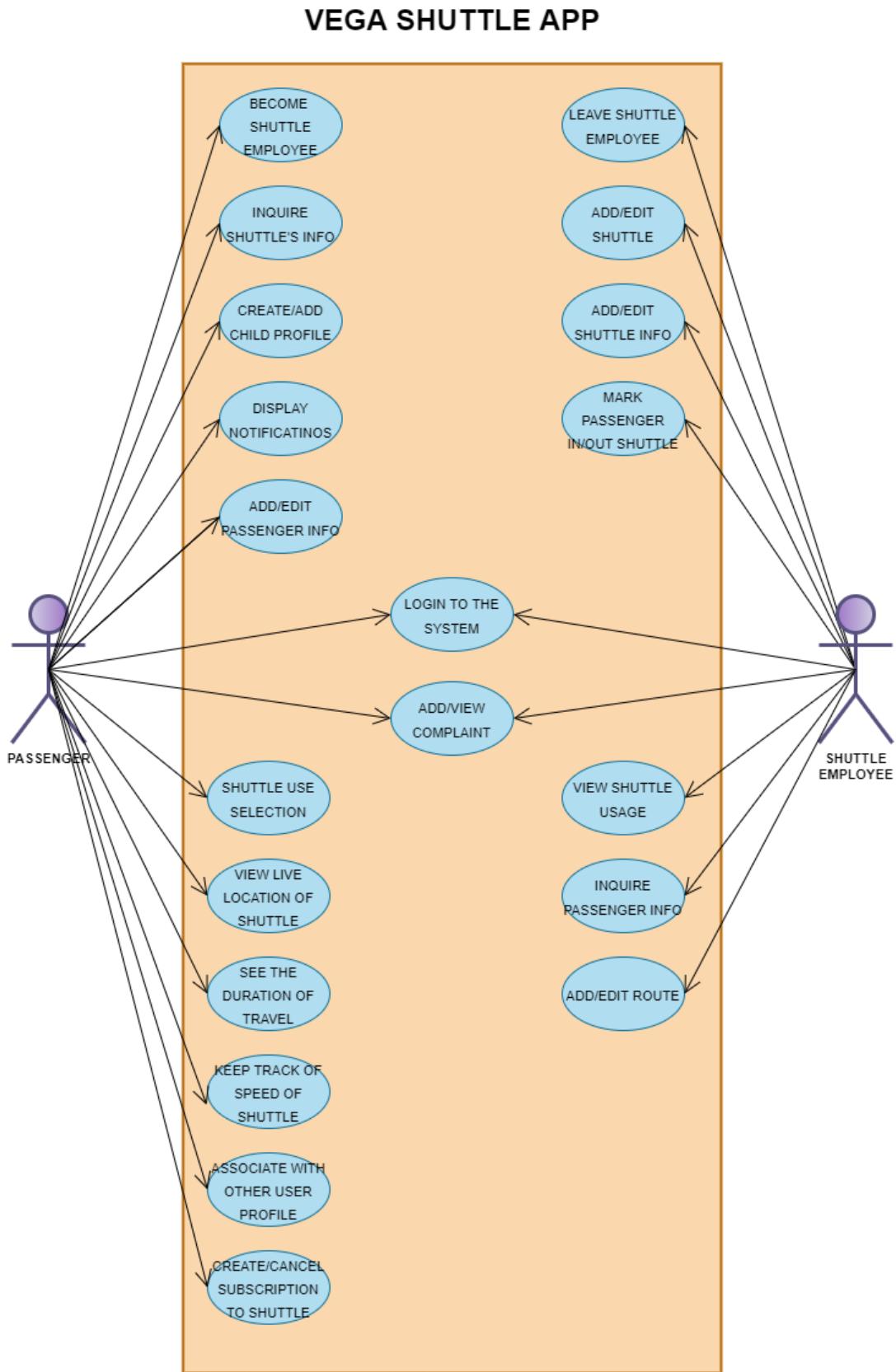
**Figure 4** Linking Child Profile and Add Shuttle Diagram

### 3.2.4 Display Shuttle and Travel Information



**Figure 5** Display Shuttle Information Process Diagram

### 3.3 Vega Shuttle App Use Case Diagram



**Figure 6** Use Case Diagram of Vega App

### 3.3.1 Detailed Information of Use-Case

#### 3.2.1 Become Shuttle Employee

<b>System</b>	Vega Shuttle App
<b>Use Case</b>	Become Shuttle Employee
<b>Actors</b>	Passenger
<b>Data</b>	Passenger information and shuttle code.
<b>Stimulus</b>	Passenger sends request to become a Shuttle Employee by entering the Shuttle code.
<b>Response</b>	Passenger becomes a Shuttle Employee.
<b>Comments</b>	The request is accepted or rejected by other employees.

#### 3.2.2 Inquire Shuttle's Information

<b>System</b>	Vega Shuttle App
<b>Use Case</b>	Inquire Shuttle's Information
<b>Actors</b>	Passenger
<b>Data</b>	The information of Shuttle used by the passenger.
<b>Stimulus</b>	The passenger requests information of shuttle.
<b>Response</b>	The information of shuttle is provided to the passenger through the database.
<b>Comments</b>	Passenger can inquire and see the information of shuttle which he/she uses.

### 3.2.3 Create/Add Child Profile

<b>System</b>	Vega Shuttle App
<b>Use Case</b>	Create/Add Child Profile
<b>Actors</b>	Passenger
<b>Data</b>	The information of child profile.
<b>Stimulus</b>	Passenger sends requests to create new child profile or sends requests to add new child profile.
<b>Response</b>	The created/added profile becomes the child profile of the passenger.
<b>Comments</b>	Child profile is created for users who do not have a phone, child profile is added for users who has phone.

### 3.2.4 Display Notifications

<b>System</b>	Vega Shuttle App
<b>Use Case</b>	Display Notifications
<b>Actors</b>	Passenger
<b>Data</b>	The notification of shuttle is 5 min away from passenger location.
<b>Stimulus</b>	When there is a 5 min distance of shuttle and passenger.
<b>Response</b>	Application sends a notification to the user.
<b>Comments</b>	Application warn the user with notification, when the shuttle comes close to passenger location.

### 3.2.5 Add/Edit Passenger Information

<b>System</b>	Vega Shuttle App
<b>Use Case</b>	Add/Edit Passenger Information
<b>Actors</b>	Passenger
<b>Data</b>	The new information of passenger or the information of passenger which will change.
<b>Stimulus</b>	Passengers want to add/or edit her/his information.
<b>Response</b>	The information of passenger updated with new information.
<b>Comments</b>	Passenger can change the information of themselves, when there is a situation which requires them to update their information.

### 3.2.6 Shuttle Use Selection

<b>System</b>	Vega Shuttle App
<b>Use Case</b>	Shuttle Use Selection
<b>Actors</b>	Passenger
<b>Data</b>	Passenger will use shuttle/ will be late to shuttle/ won't use the shuttle.
<b>Stimulus</b>	Passenger selects one of 3 use cases.
<b>Response</b>	Database receives the usage information.
<b>Comments</b>	Passengers can select one of the case every time when they use shuttle. "Will use shuttle" case is a default case. When passengers do not select any case default case become active.

### 3.2.7 View Live Location of Shuttle

<b>System</b>	Vega Shuttle App
<b>Use Case</b>	View Live Location Of Shuttle
<b>Actors</b>	Passenger, Shuttle Employee
<b>Data</b>	The location of the Shuttle Employee( which is the location of the shuttle)
<b>Stimulus</b>	Passenger clicks the refresh button of the map.
<b>Response</b>	The live location of the shuttle will be visible by passenger.
<b>Comments</b>	The live location information is refreshed when the passenger clicks the refresh button.

### 3.2.8 See the Duration of Travel

<b>System</b>	Vega Shuttle App
<b>Use Case</b>	See The Duration of Travel
<b>Actors</b>	Passenger, Shuttle Employee
<b>Data</b>	The time distance between the location of the Shuttle Employee(which is the location of the shuttle) and the destination.
<b>Stimulus</b>	Passenger wants the see the duration of travel.
<b>Response</b>	The duration of travel will be visible by passenger.
<b>Comments</b>	The duration of travel information is up-to-date.

### 3.2.9 Keep Track of Speed of Shuttle

<b>System</b>	Vega Shuttle App
<b>Use Case</b>	Keep Track Of Speed Of Shuttle
<b>Actors</b>	Passenger, Shuttle Employee
<b>Data</b>	The speed of the Shuttle(which is receive by Shuttle Employee)
<b>Stimulus</b>	Passenger wants the see the speed of shuttle.
<b>Response</b>	The speed of travel will be visible by passenger.
<b>Comments</b>	The speed of shuttle information is up-to-date. Beside the shuttle speed the speed limit of the road is also be shown.

### 3.2.10 Associate with Other User Profile

<b>System</b>	Vega Shuttle App
<b>Use Case</b>	Associate with Other User Profile
<b>Actors</b>	Passenger, Other Passenger Profile
<b>Data</b>	The information of both profiles.
<b>Stimulus</b>	Passenger wants to link herself/himself with other profile.
<b>Response</b>	The profiles is linked if both profiles have permission.
<b>Comments</b>	The unique codes for each profile is a key point for having a permission.

### 3.2.11 Leave Shuttle Employee

<b>System</b>	Vega Shuttle App
<b>Use Case</b>	Leave Shuttle Employee
<b>Actors</b>	Shuttle Employee
<b>Data</b>	Shuttle Employee information and selected Shuttle
<b>Stimulus</b>	Shuttle Employee selects the shuttle which she/he wants to leave.
<b>Response</b>	The user is no longer an employee of the selected shuttle.
<b>Comments</b>	User can be re-employee, if she/he wants.

### 3.2.12 Add/Edit Shuttle Information

<b>System</b>	Vega Shuttle App
<b>Use Case</b>	Add/Edit Shuttle Information
<b>Actors</b>	Shuttle Employee
<b>Data</b>	The new information of shuttle or the information of shuttle which will change.
<b>Stimulus</b>	Shuttle Employee wants to add/edit shuttle information which she/he responsible of.
<b>Response</b>	The information of shuttle updated with new information.
<b>Comments</b>	Shuttle Employee can change the information of shuttle ,when there is a situation which requires them to update shuttle information.

### 3.2.13 Mark Passenger In/Out Shuttle

<b>System</b>	Vega Shuttle App
<b>Use Case</b>	Mark Passenger In/Out Shuttle
<b>Actors</b>	Shuttle Employee,Passenger
<b>Data</b>	Live locations of shuttle employee and passenger.
<b>Stimulus</b>	When the user sees the shuttle or vice versa.
<b>Response</b>	Passenger will be marked as a inside/outside of shuttle.
<b>Comments</b>	As a result of this case, the parent of child user can see their child is in the shuttle with the peace of mind.

### 3.2.14 Add/Edit Shuttle

<b>System</b>	Vega Shuttle App
<b>Use Case</b>	Add/Edit Shuttle
<b>Actors</b>	Shuttle Employee
<b>Data</b>	The new shuttle or the shuttle which will change.
<b>Stimulus</b>	Shuttle Employee wants to add/edit shuttle which she/he responsible of.
<b>Response</b>	The new shuttle or the shuttle which is changed, is updated to database and shown in the application.
<b>Comments</b>	Shuttle Employee can add new shuttle to the system or edit the currently added shuttle.

### 3.2.15 View Shuttle Usage

<b>System</b>	Vega Shuttle App
<b>Use Case</b>	View Shuttle Usage
<b>Actors</b>	Shuttle Employee,Passenger
<b>Data</b>	The shuttle usage information of passenger.
<b>Stimulus</b>	Shuttle Employee wants to see the usage of her/his shuttle.
<b>Response</b>	The Shuttle usage information will be visible by Shuttle Employee.
<b>Comments</b>	User selected usage case will be visible by Shuttle Employee. If the user did not select any, the default selection will be visible.

### 3.2.16 Inquire Passenger Information

<b>System</b>	Vega Shuttle App
<b>Use Case</b>	Inquire Passenger Information
<b>Actors</b>	Shuttle Employee, Passenger
<b>Data</b>	The information of passenger.
<b>Stimulus</b>	Shuttle Employee wants to see the information of the passenger.
<b>Response</b>	The information of passenger will be visible.
<b>Comments</b>	Shuttle Employee can inquire passengers who uses her/his shuttle.

### 3.2.17 Add/Edit Route

<b>System</b>	Vega Shuttle App
<b>Use Case</b>	Add/Edit Route
<b>Actors</b>	Shuttle Employee,Passengers
<b>Data</b>	The station of passengers and the destination location.
<b>Stimulus</b>	Shuttle Employee wants to add/edit route.
<b>Response</b>	The created/edited route will be added to the database, and can be visible in the application.
<b>Comments</b>	The route will be visible by all the passengers who uses the shuttle. However they can only see their station not all of the passengers stations.

### 3.2.18 Login to the System

<b>System</b>	Vega Shuttle App
<b>Use Case</b>	Login To The System
<b>Actors</b>	Shuttle Employee, Passenger
<b>Data</b>	Email Adress and Password
<b>Stimulus</b>	User enter their valid email and password.
<b>Response</b>	User logs in to the system.
<b>Comments</b>	Users can log in to the system.

### 3.2.19 Create/Cancel Subscription to Shuttle

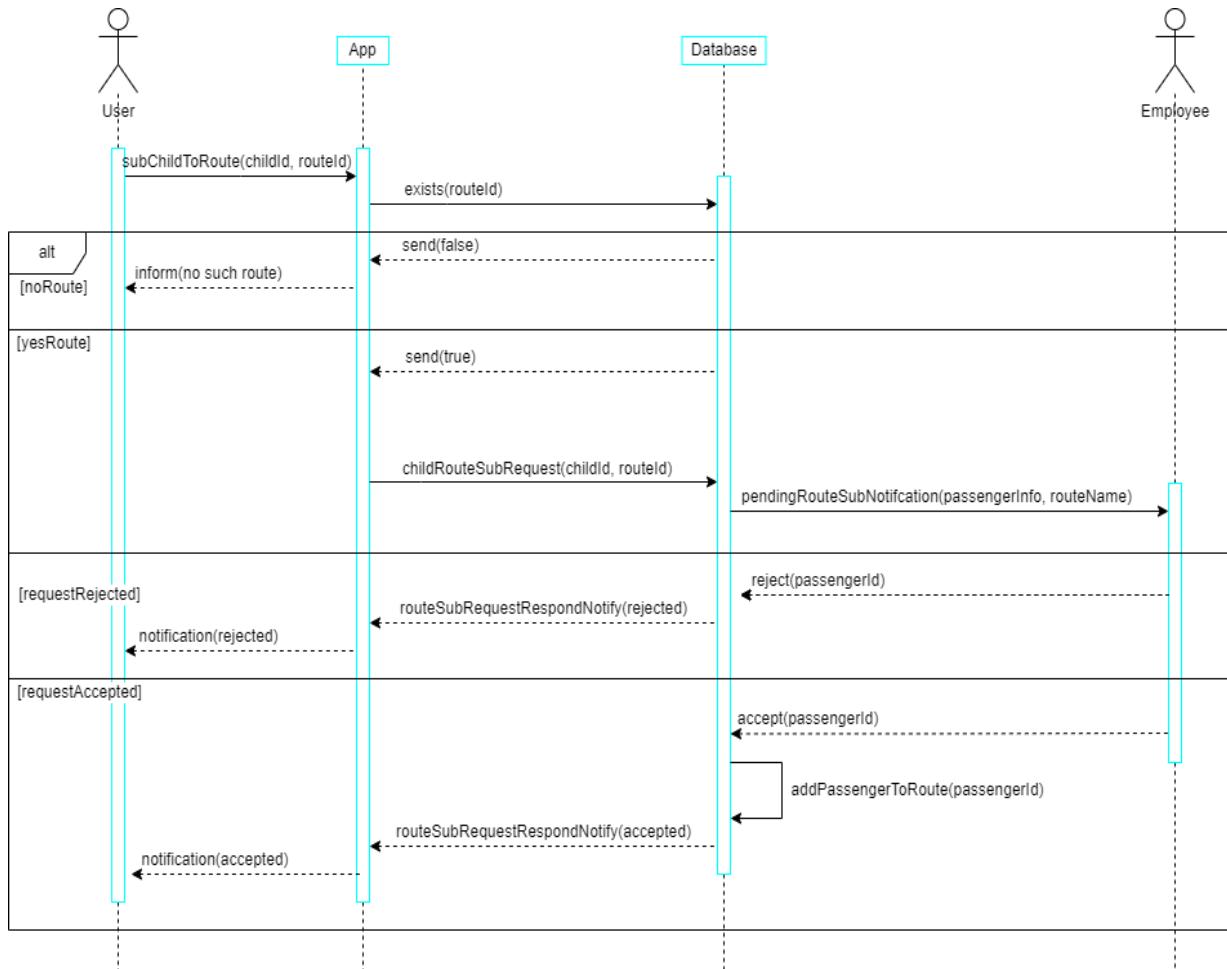
<b>System</b>	Vega Shuttle App
<b>Use Case</b>	Create/Cancel Subscription To Shuttle
<b>Actors</b>	Passenger
<b>Data</b>	Passenger information and shuttle code
<b>Stimulus</b>	Passenger enters the code of the shuttle she/he wants to subscribe and sends the request./Passenger selects the shuttle she/he wants to cancel the subscription.
<b>Response</b>	Passenger subscribes to the shuttle and becomes the passenger of that shuttle./ User no longer passenger of that(selected) shuttle.
<b>Comments</b>	The request is accepted or rejected by Shuttle Employee.

### 3.2.20 Add/View Complaint

<b>System</b>	Vega Shuttle App
<b>Use Case</b>	Add/View Complaint
<b>Actors</b>	Shuttle Employee, Passenger
<b>Data</b>	The complaint text of the users.
<b>Stimulus</b>	User wants to give the feedback about application.
<b>Response</b>	The feedback will be received from system and the thank you message will be sent.
<b>Comments</b>	Users can give complaint or feedback about their usage about application. And these feedbacks improves the application in a good way.

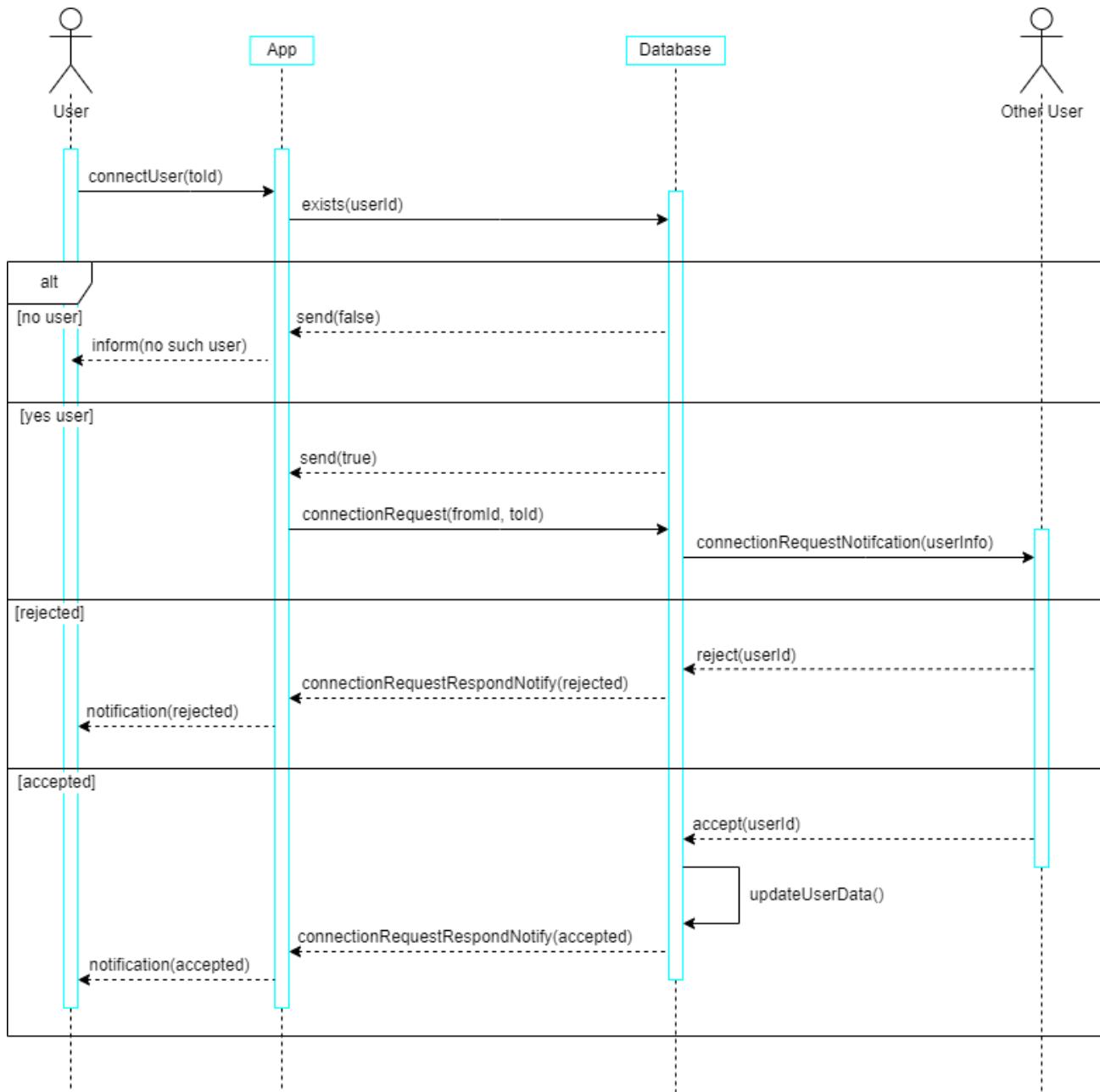
## 3.4. Sequence Diagrams

### 3.4.1 Child Route Subscription



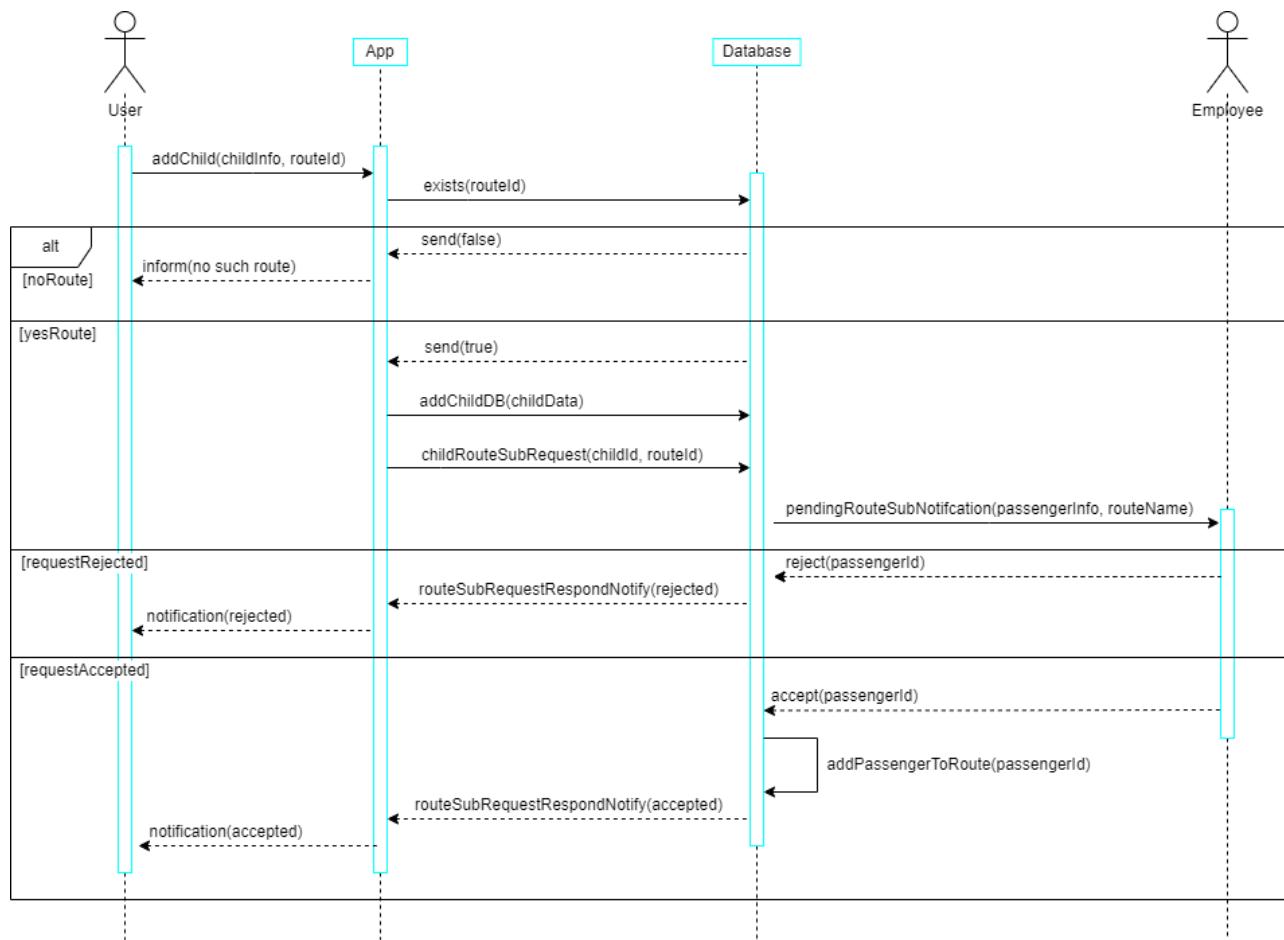
*Figure 7 Child Route Subscription*

### 3.4.2 Connection Request



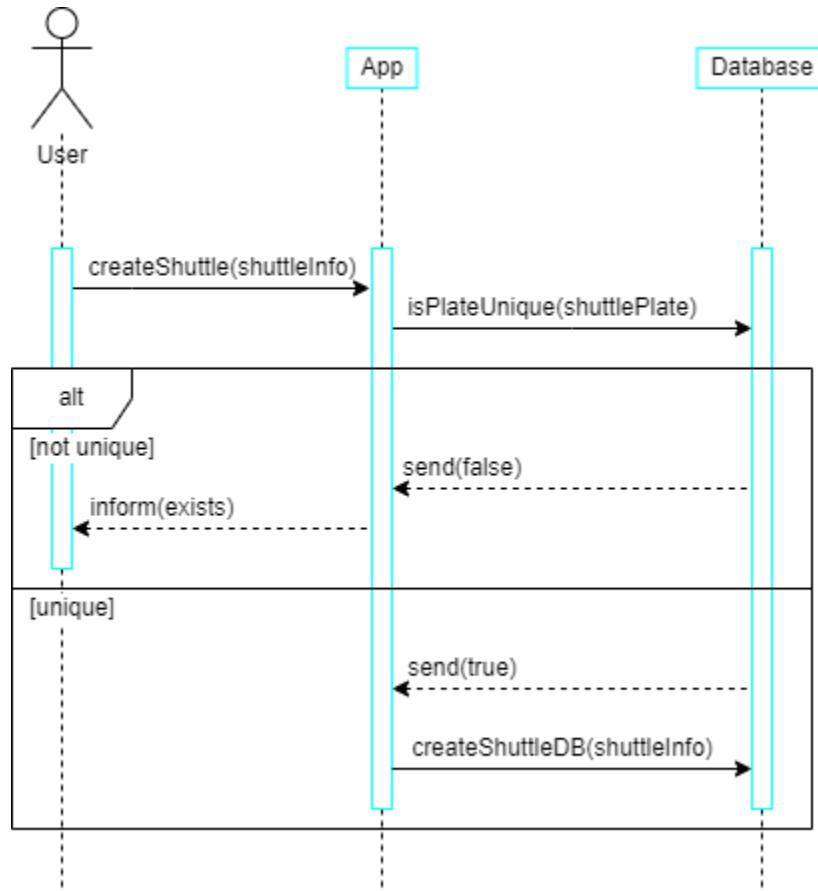
**Figure 8** Connection Request Sequence

### 3.4.3 Create Child Sequence



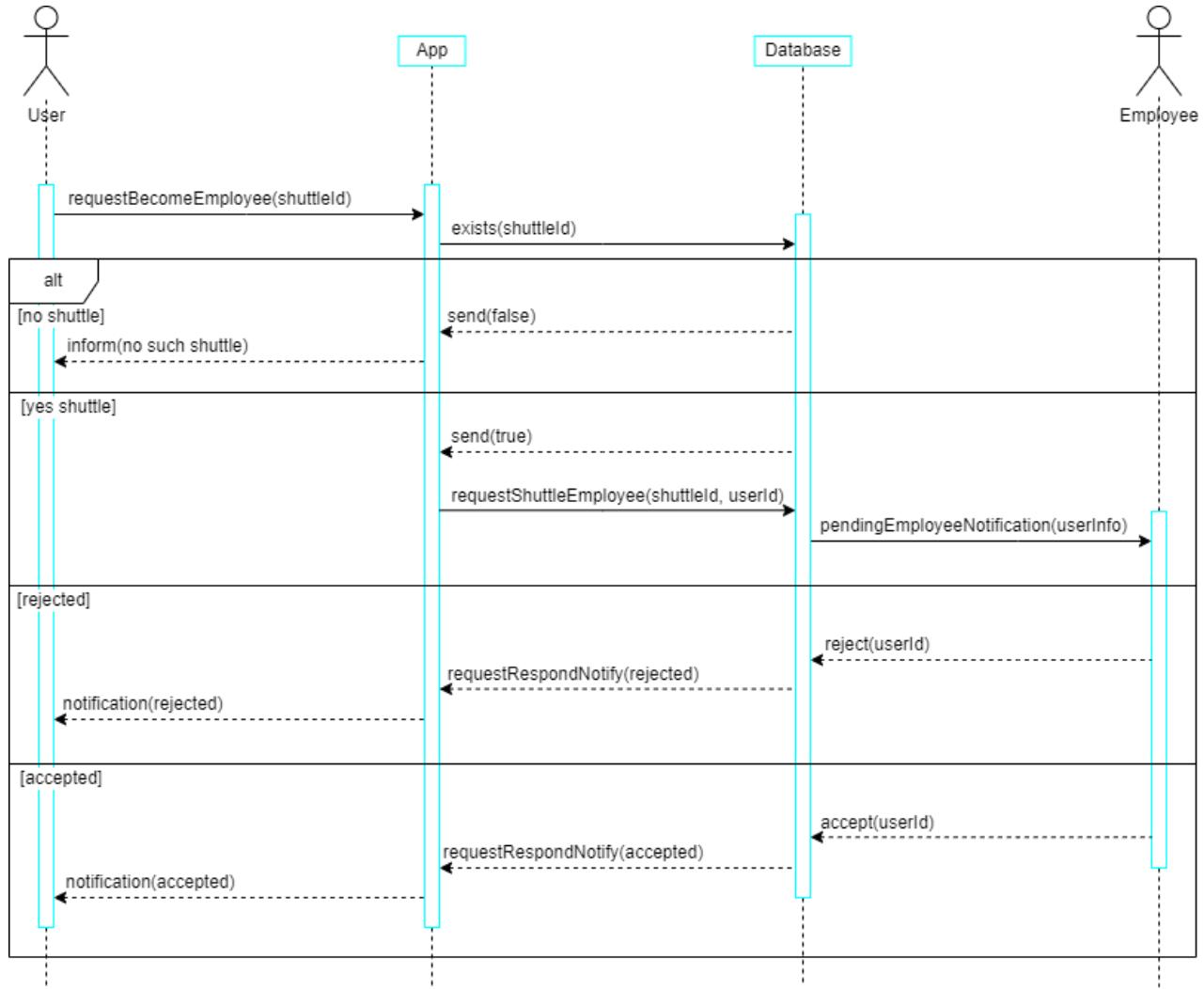
**Figure 10** Creating Child Sequence

### 3.4.4 Create Shuttle Sequence



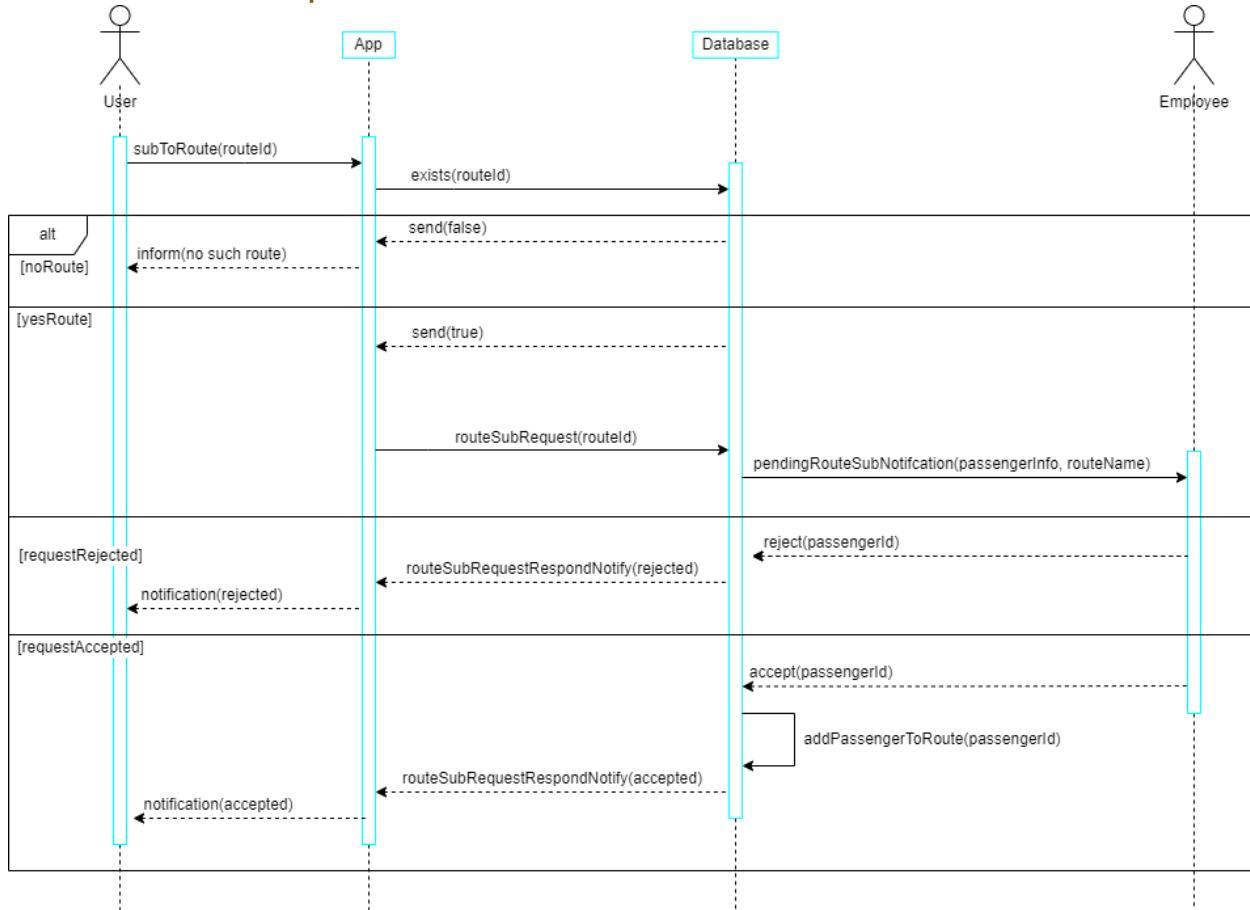
**Figure 11** Create Shuttle Sequence

### 3.4.5 Shuttle Employee Request



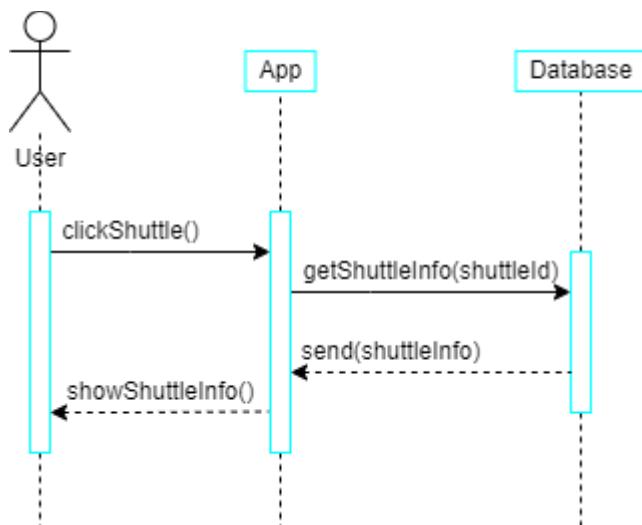
**Figure 12** Shuttle Employee Request

### 3.4.6 Subscribe Route Sequence



**Figure 13** Subscribe Route Sequence

### 3.4.7 Show Shuttle Information



**Figure 14** Show Shuttle Info

### 3.4.8 Employee Leave Sequence

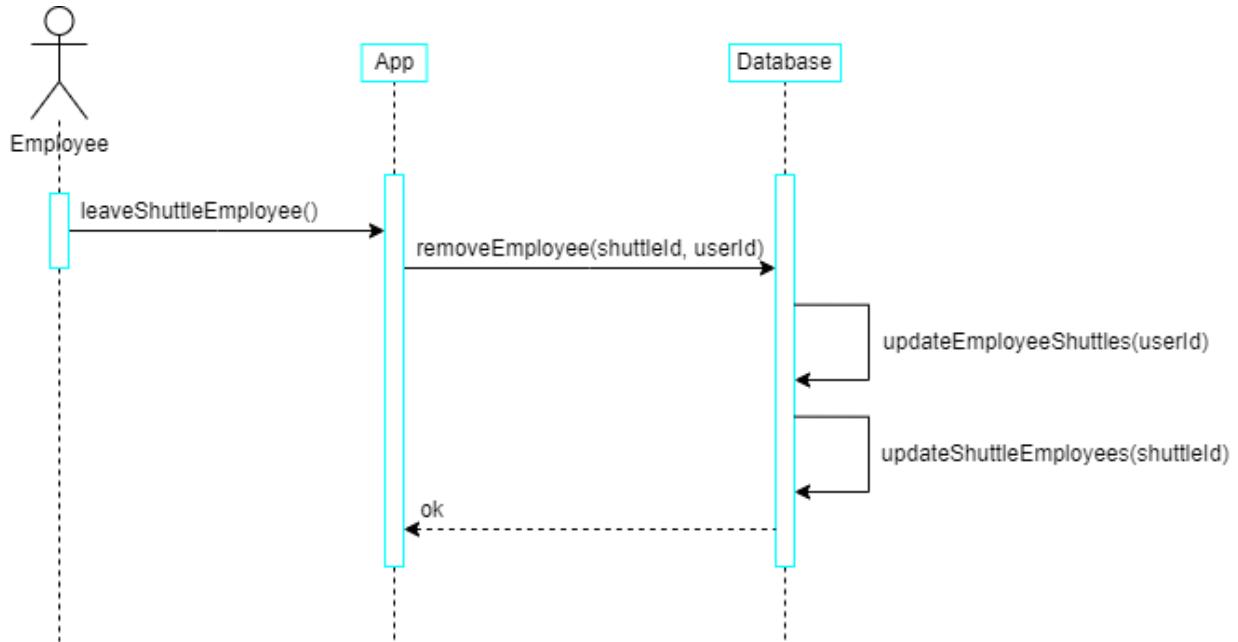


Figure 15 Leave Employee Seq.

### 3.4.9 Update Passenger Status

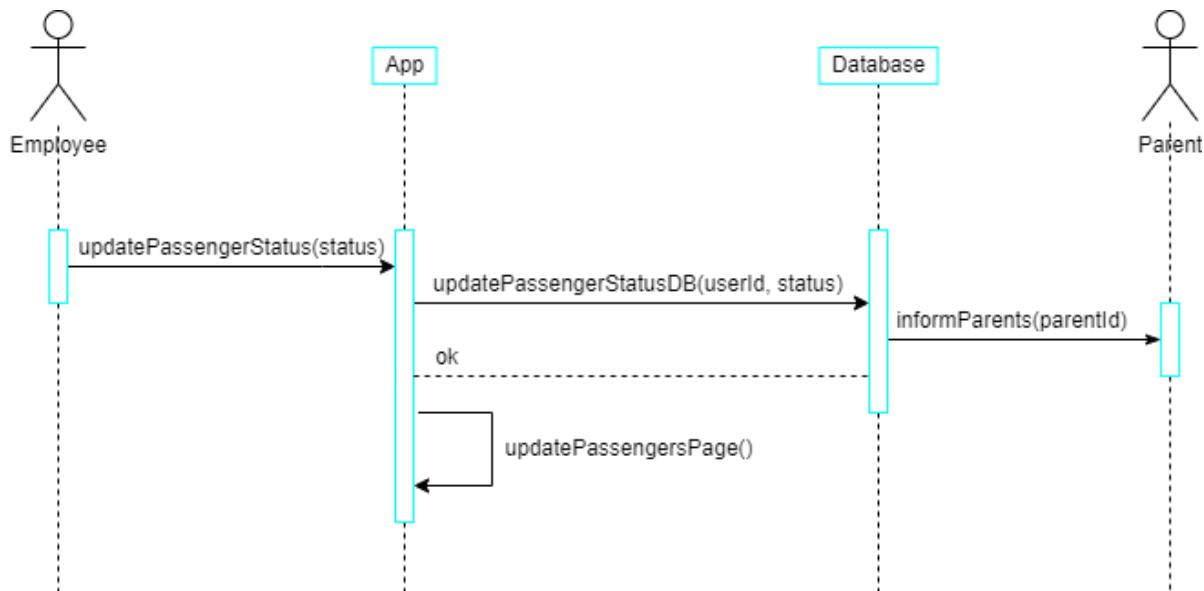
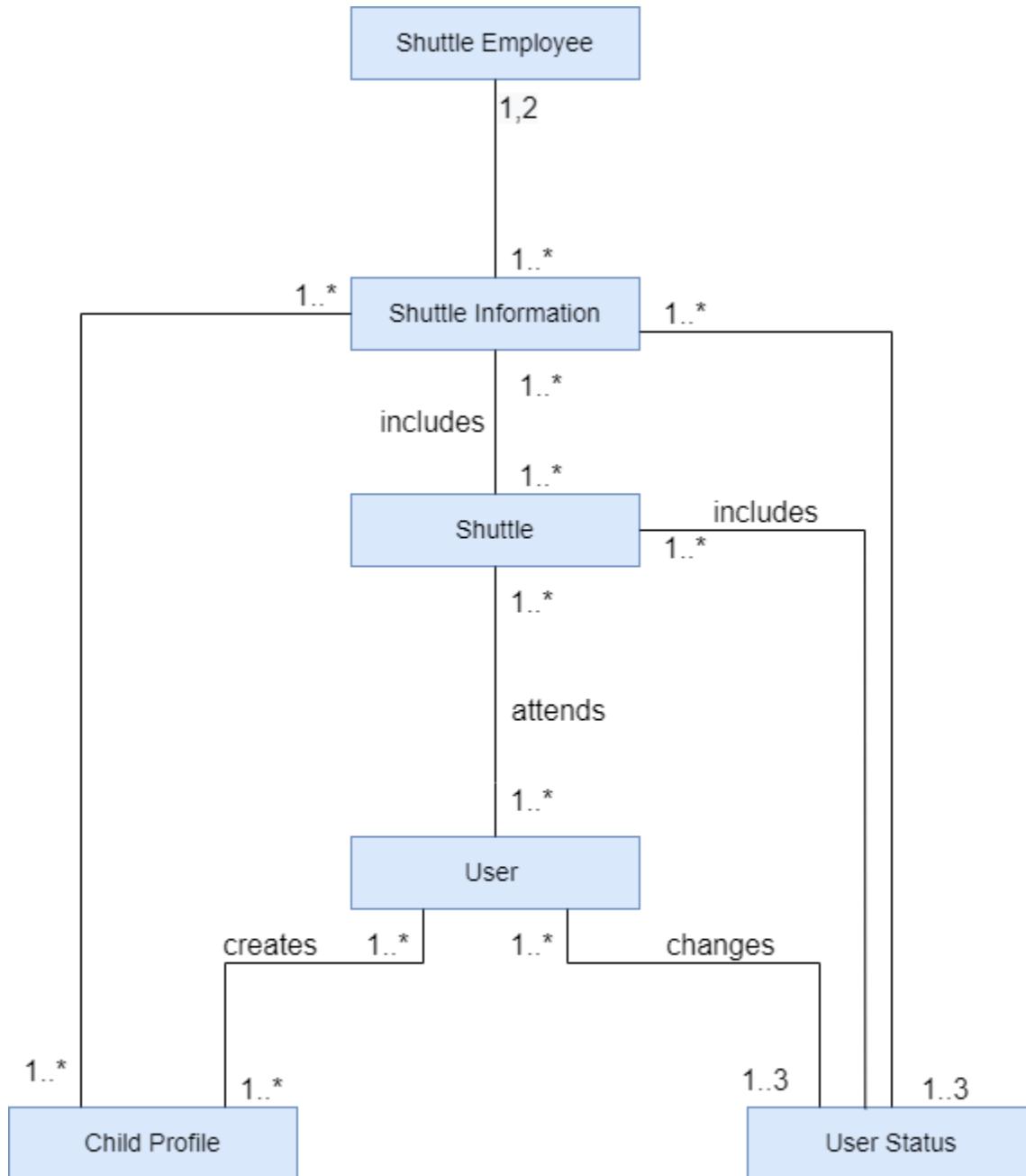


Figure 16 Employee Leave Sequence

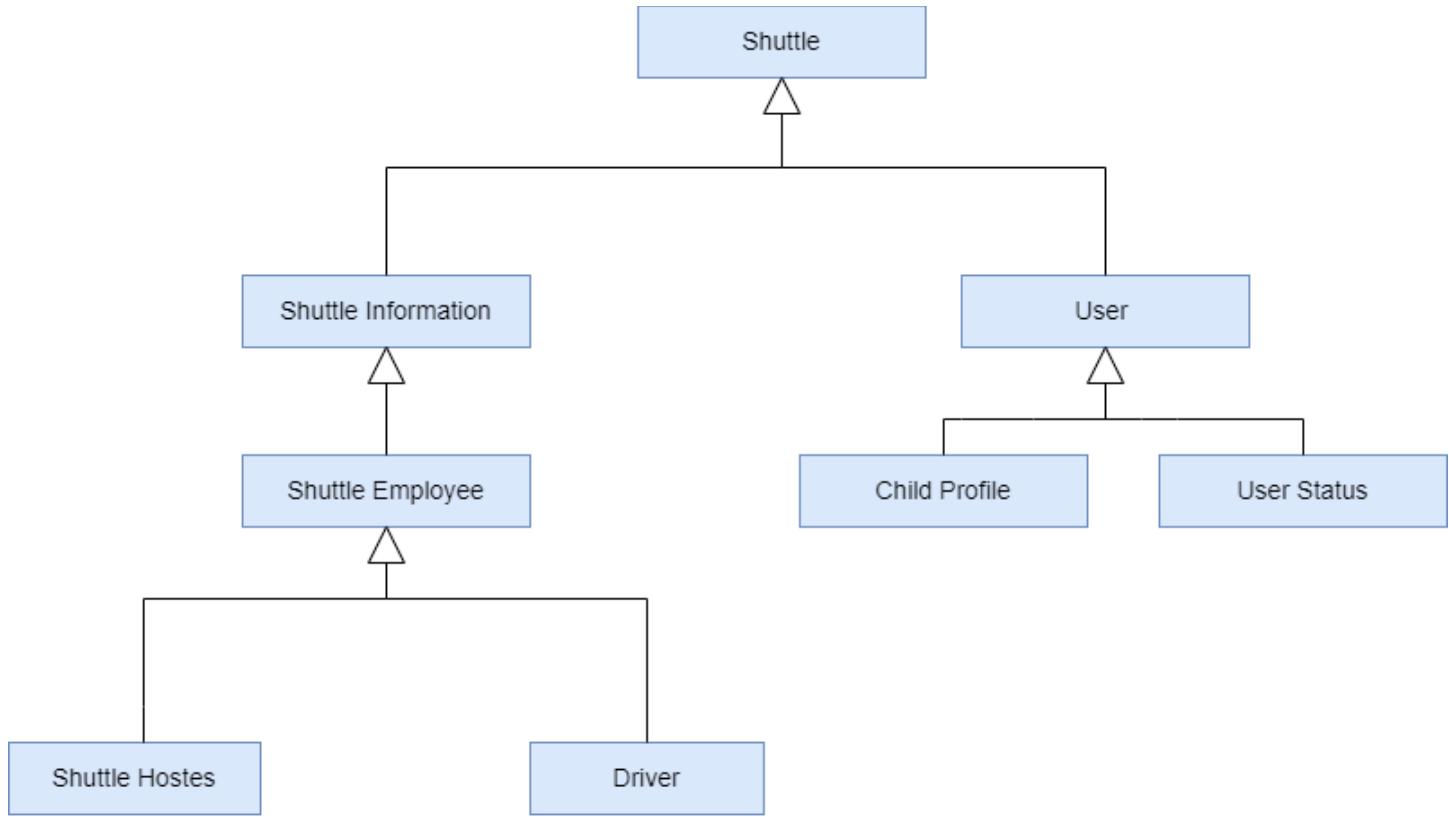
## 3.5 Class Diagrams

### 3.5.1 Classes and Associations



**Fig3.5.1** Classes and associations in the Vega Shuttle App

### 3.5.2 Generalization Hierarchy



**Fig 3.5.2** Generalization Hierarchy of VegaShuttle App

### 3.5.3 Class Diagram

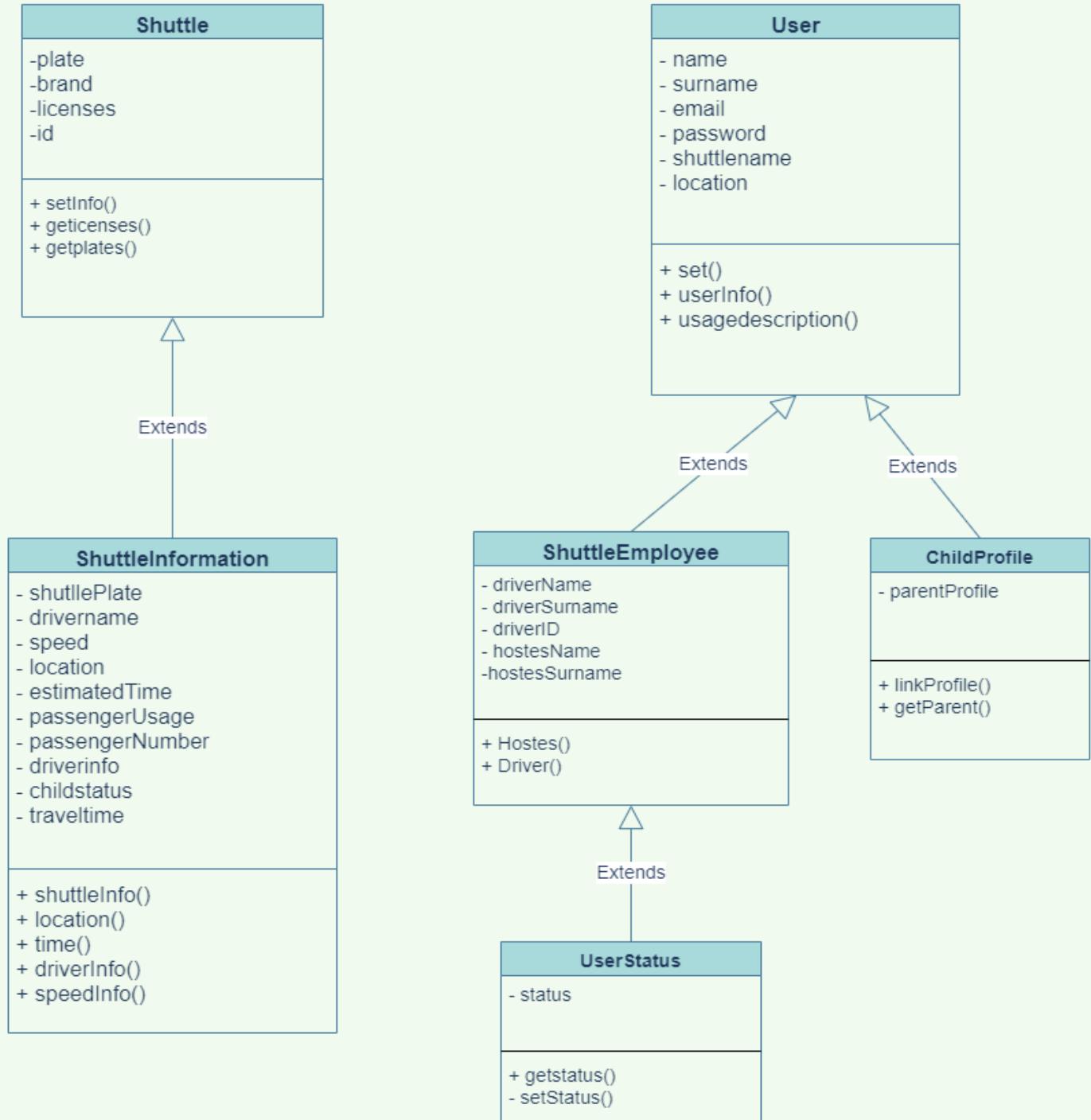
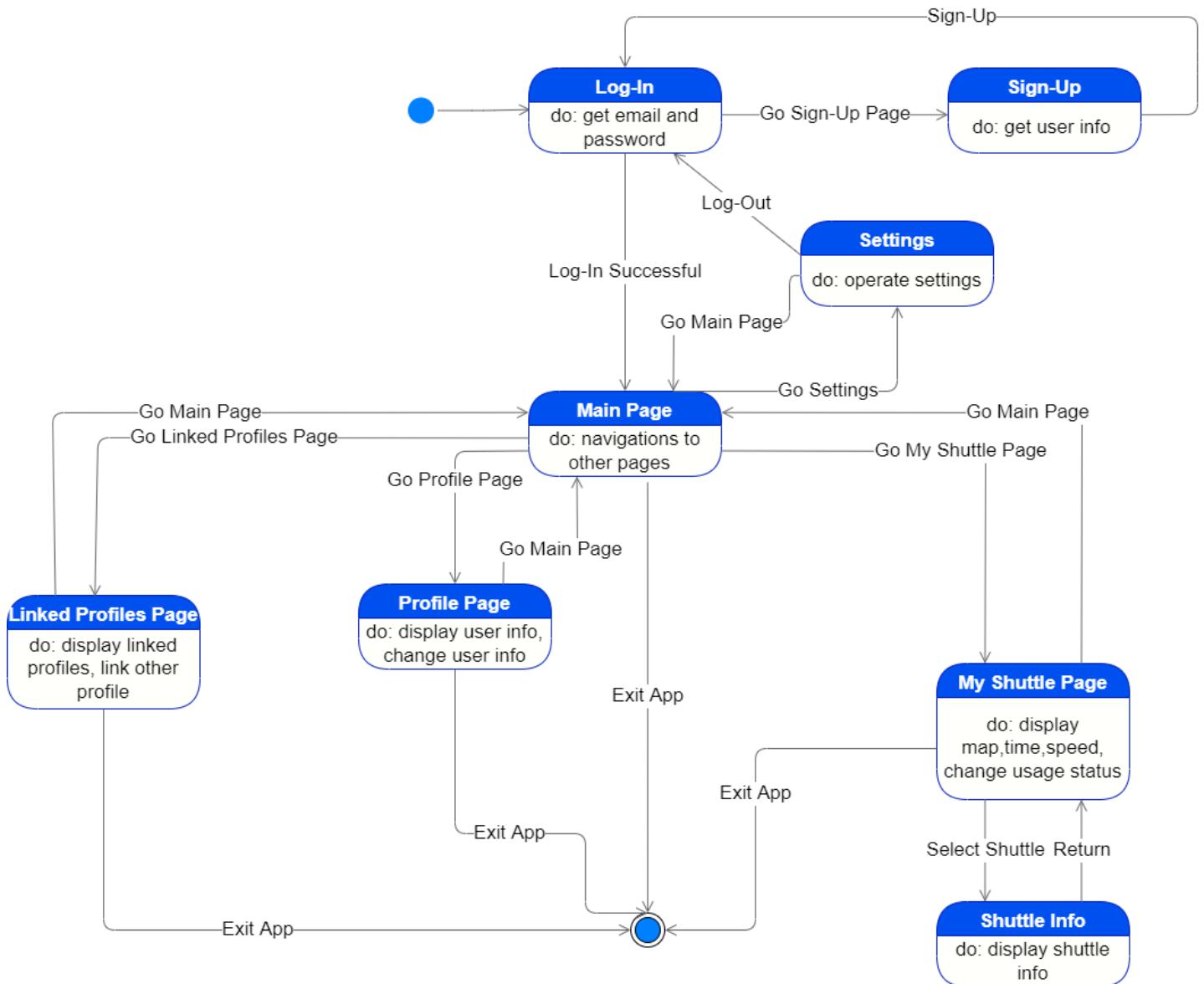


Figure 3.5.3 Class Diagram of Vega App

## 3.6 State Diagram

### 3.6.1 State Diagram



**Figure 13** State Diagram

### 3.6.2 Profile Page State Diagram

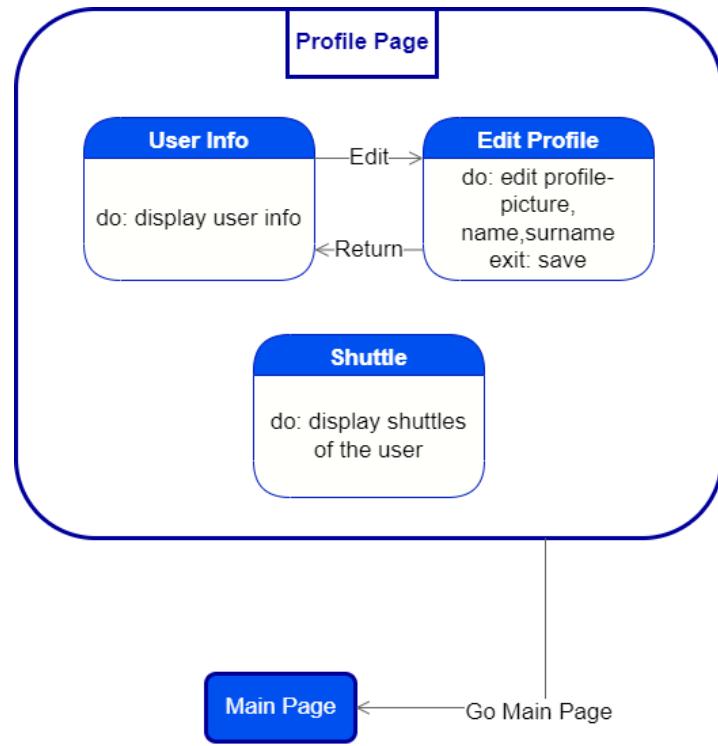


Figure 14 Profile Page State Diagram

### 3.6.3 My Shuttle Page State Diagram

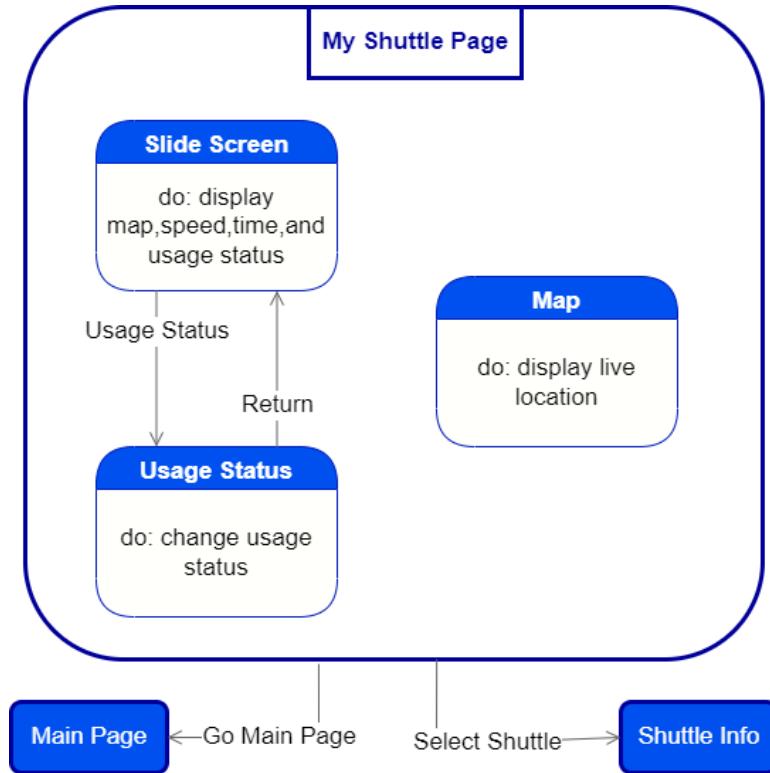


Figure 15 My Shuttle Page State Diagram

### 3.6.4 Linked Profiles Page State Diagram

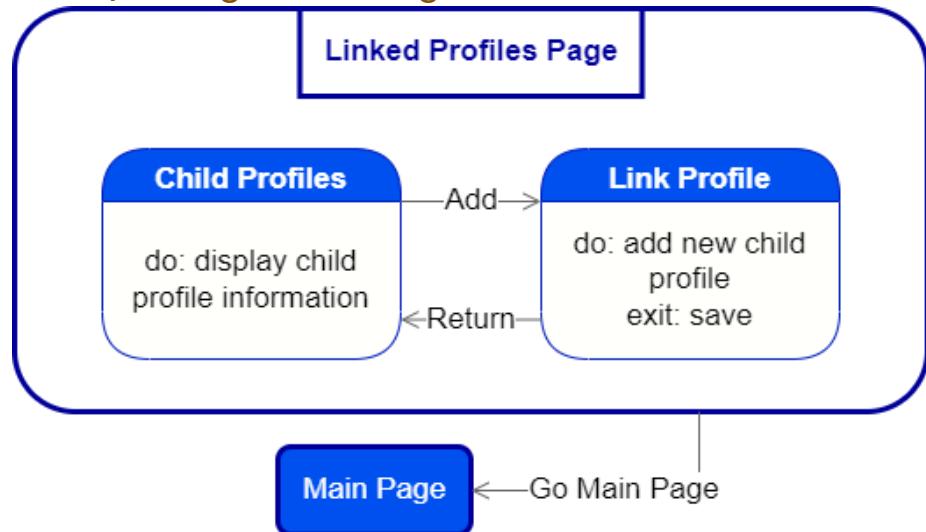


Figure 16 Linked Profiles Page State Diagram

### 3.6.5 Settings Page State Diagram

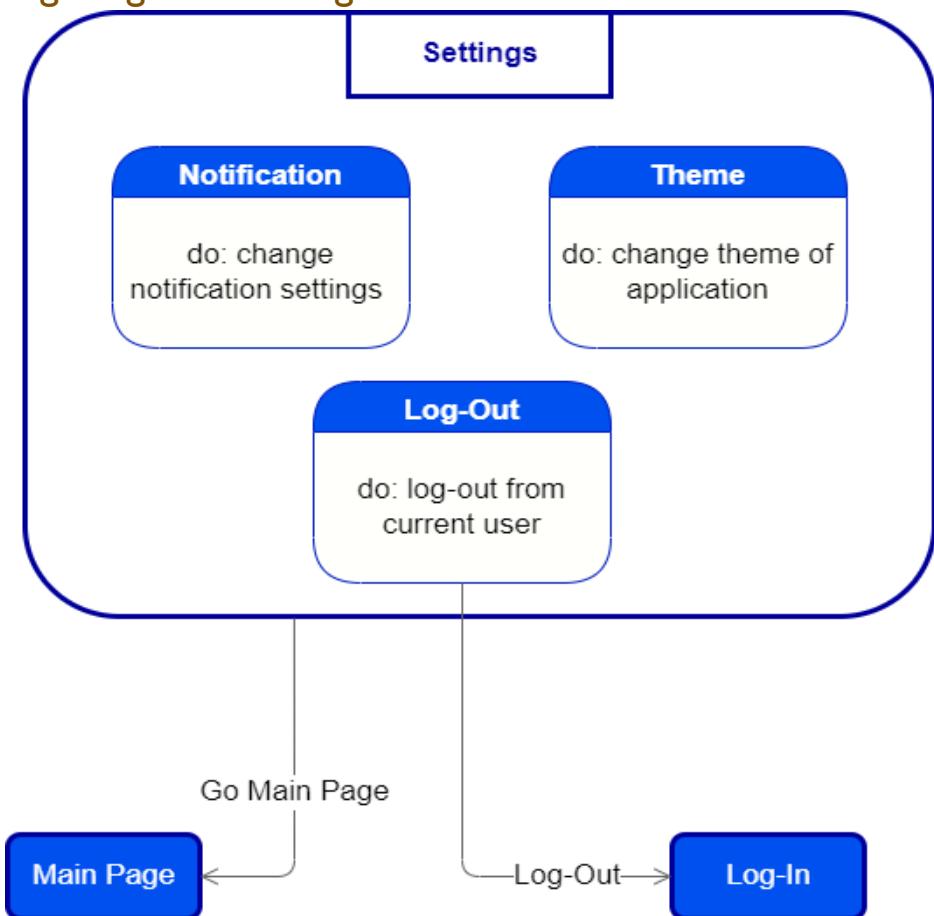


Figure 17 Settings Page State Diagram

# **4. Simple System Architecture Document**

## **4.1. Introduction**

This report will define the high level design and technology decisions of the VegApp.

This simple system architecture document defines and describes the use of each view, the architectural constraints of the system, the functional requirements with a significant impact on the architecture, use-case realization, concurrency aspects, the layers and subsystems of the application, performance issues and constraints.

### **4.1.1 Purpose**

The purpose of this VegApp is provide safe trip for children and other users by knowing and following the shuttle's information. Likewise, this software provides the driver, with the convenience of using passenger information and ensuring their safety in an emergency. With the help of VegApp, families whose children use shuttle can follow status of their children with peace of mind.

## **4.2 Architectural Representation**

The views used to document the VegaApp system are:

- 1) Use Case view
- 2) Logical View
- 3) Pyhsical View/Deployment View

## **4.2.1. Use Case view**

**Audience:** all the stakeholders of the system, including the end-users.

**Area:** describes the set of scenarios and/or use cases that represent some significant, central functionality of the system. Describes the actors and use cases for the system, this view presents the needs of the user and is elaborated further at the design level to describe discrete flows and constraints in more detail.

**Concerns:** Understandability and usability

**Related Artefacts :** Use-Case Model, Use-Case documents

### *4.2.1.1 Use Case Scenarios*

If you click the title, you can see use case scenarios.

### *4.2.1.2 Use-Case View*

If you click the title, you can see use case diagram

### *4.2.1.3 Detailed Use-Case View*

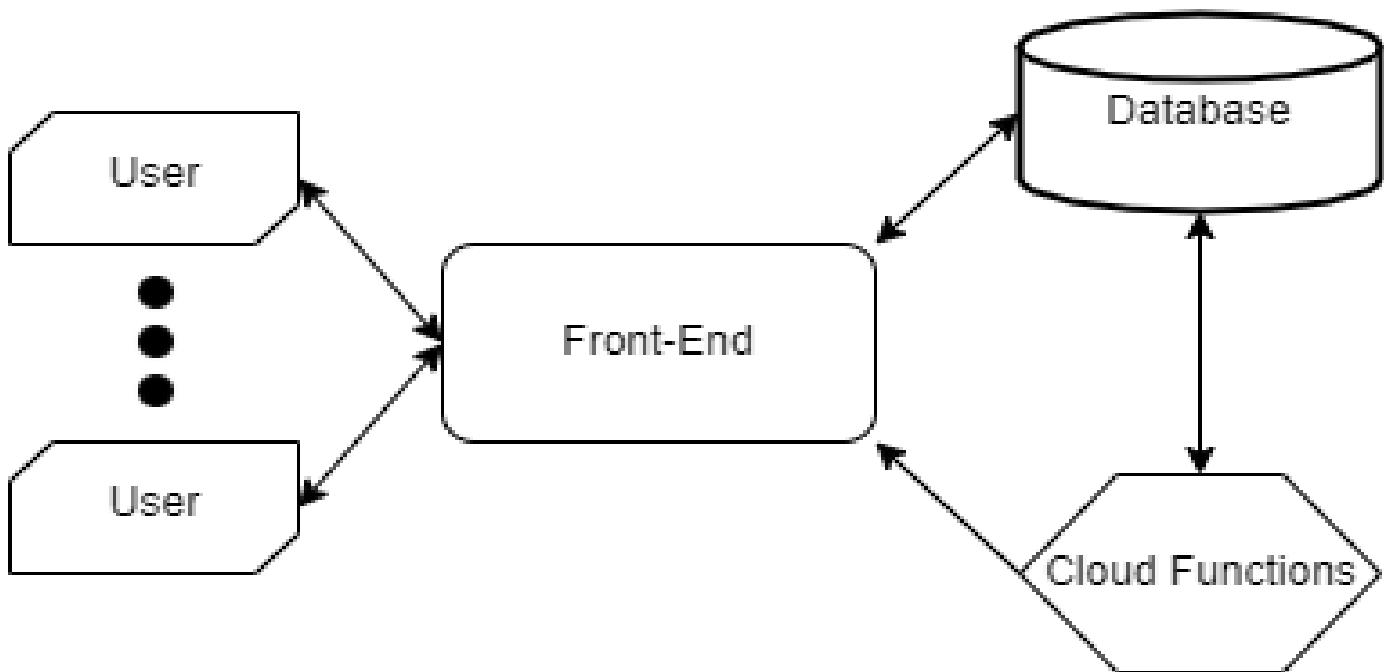
If you click the title, you can see detailed use case diagram

## 4.2.2 Logical View

### 4.2.2.1 Overview

The architecture of Vega consists of mobile app as front-end, real-time database, and cloud functions. Users make changes through interactions between front-end and database directly, database has rules to let whom to change what on it and checks every action itself. Cloud functions make further changes on database depending on the changes on it, where it is impractical or not possible to do through front-end. It also manages the notifications that are seen at front-end depending on the database but does not accept anything from front-end itself, therefore making it one way relation.

Architecture Diagram



### 4.2.2.2 Constraints and Decisions

Since what Vega offers for its users has real-time values in it, it can not be used without internet connection. All the data is stored in one database for easier and faster interactions with other components, but this comes with a disadvantage that it is either everyone can use it or no one. Cloud functions watch and handle the changes on database to keep it tidy and forward the interactions between data classes. It lightens the work on the clients' device for some actions, but again with the cost of not being able to do those actions if it was to go down.

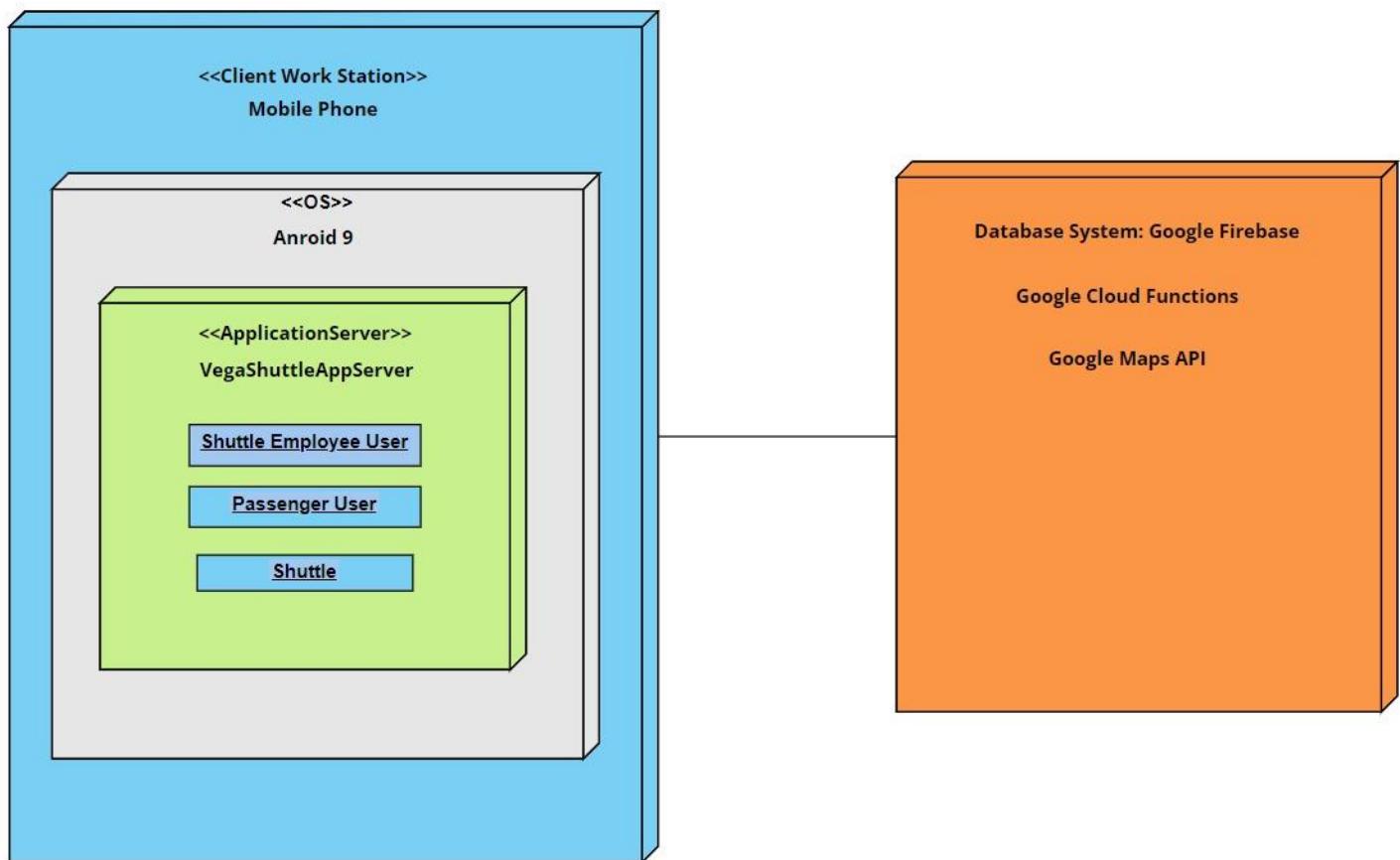
Front-end is written on Flutter for its easy connection to Firebase, and it being easy to program on framework.

## 4.2.3 Deployment View

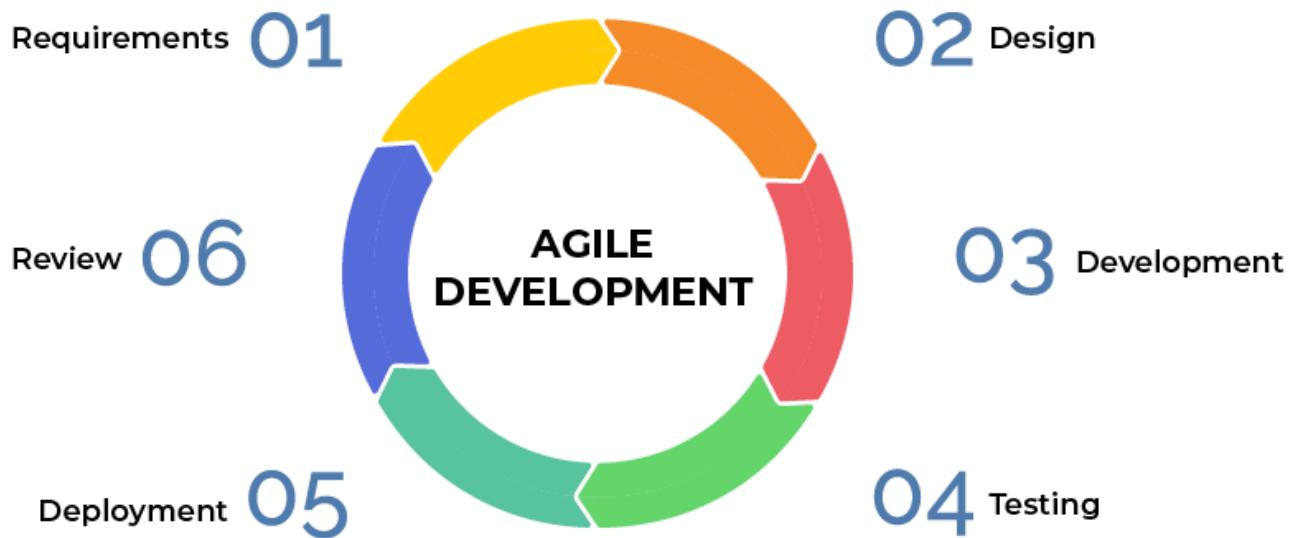
**Audience:** Deployment managers.

**Area:** describes the mapping of the software onto the hardware and shows the system's distributed aspects. Describes potential deployment structures, by including known and anticipated deployment scenarios in the architecture we allow the implementers to make certain assumptions on network performance, system interaction and so forth.

**Related Artifacts:** Deployment model.



# Software Development Life Cycle



## 1. Requirement analysis

At the first stage, all you need to do is about **deciding what you're going to develop**. Should you develop for clients, spend time getting to know their needs, goals, and expectations towards the final product. During the first phase of software development life cycle, outline the following:

- Answer questions such as: How will the software be used? Who is going to use the software?
- Next, analyze those requirements for feasibility and validity: cost, performance, functionality.
- Then, evaluate the scope of the project and identify available resources.
- Lastly, consider project opportunities and risks in each software development life cycle phase.

## 2. UI/UX Design

The second phase of the software development life cycle is determining which period the **entire architecture of the future project is created**. During the design phase, testers come up with a testing strategy suggesting what and how it needs to be tested.

- Every required component specifies systems architecture, configurations, data structure.
- Desired output including prototypes, pseudocode.
- Design details like the functional logic, interface details, dependency issues, and errors.

### **3. Software Development**

After all requirements analysis and design choices are already defined, the actual coding starts. In fact, this is **the longest phase** of the entire software development lifecycle.

### **4. Testing and Quality Assurance**

Once the code is fully created, testing is carried against the requirements. The testing phase of SDLC is about quality assurance. Testing also involves the **checking of any faulty parts of the code and their fixes**.

- Develop a test plan based on the predefined software requirements.
- Identify the resources available for testing such as instructions and assignments for testers, selects types of tests and reports to technical executives.
- Work collectively with development teams to improve test results.

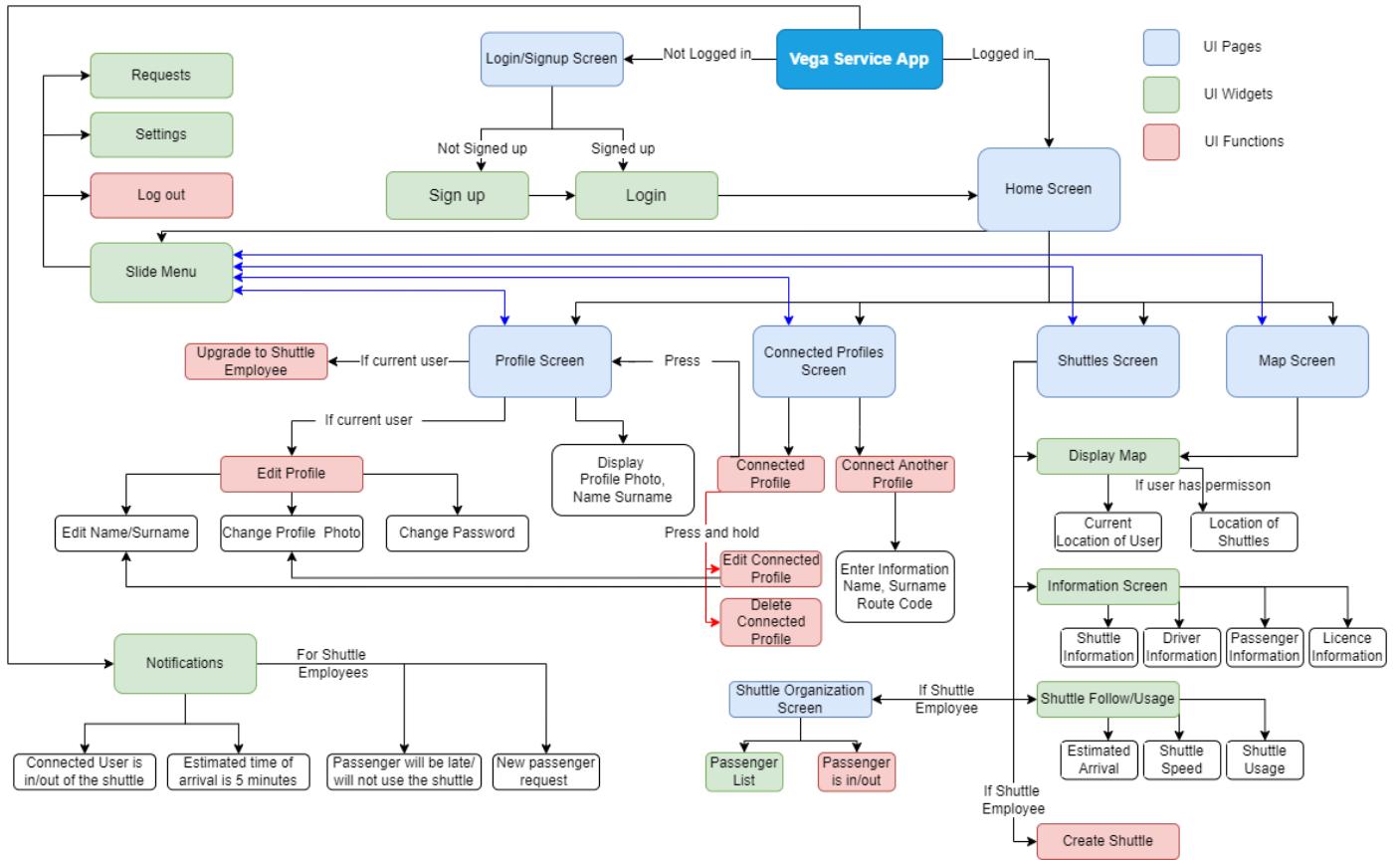
### **5. Deployment**

After the software has been tested, the final deployment takes place. But, this is **not as simple as the click of a button**. There are a lot of things to consider in this software development life cycle process, including:

- Who will give the final approval before launching?
- Do you launch it to everyone in production or selected users?
- If there are issues with the deployment, what will you do next?

## 6. Review

In the final phase of the SDLC, product is reviewed and decided whether or not any requirements are still present.

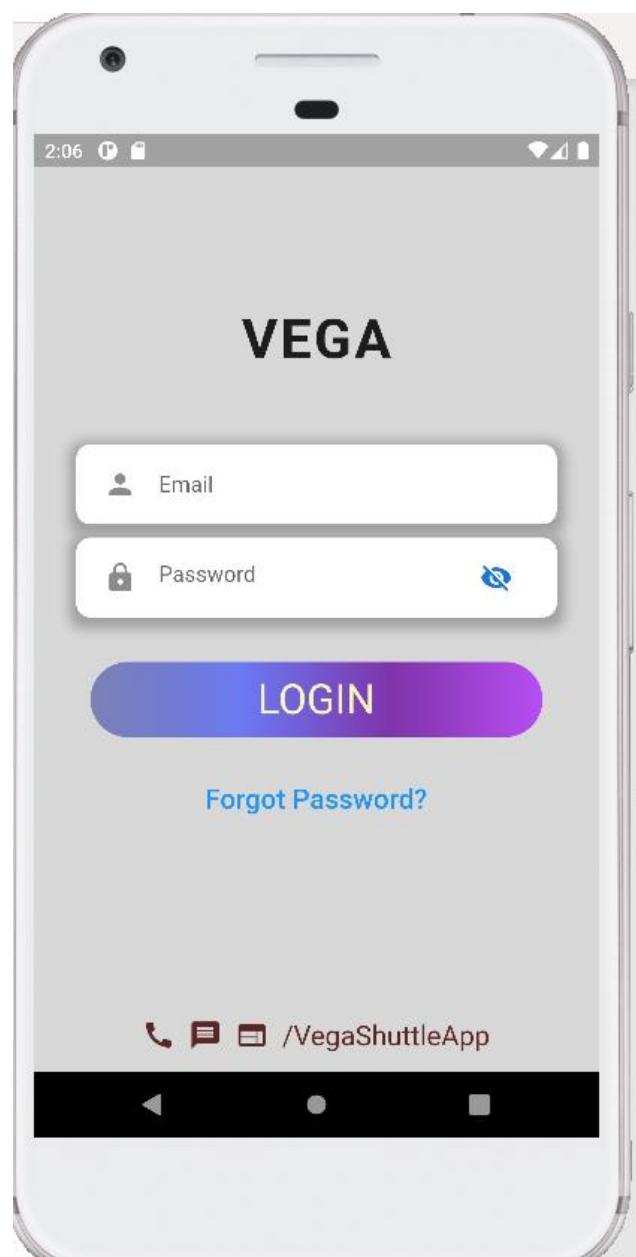


## **5. Working Graphical Interface**

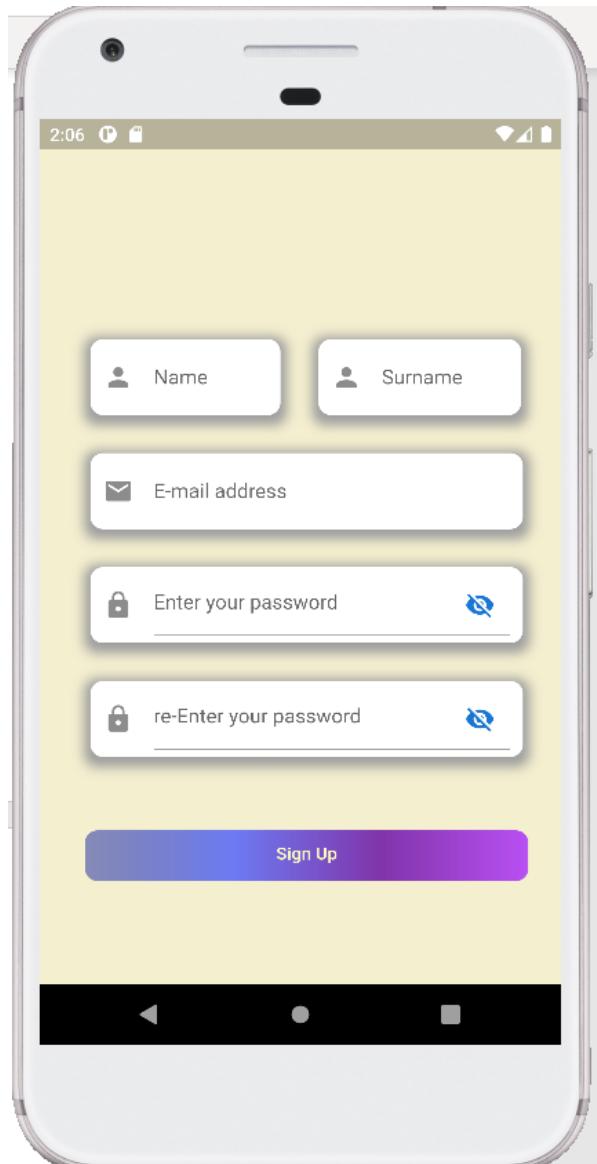
5.a.1) Welcome Page



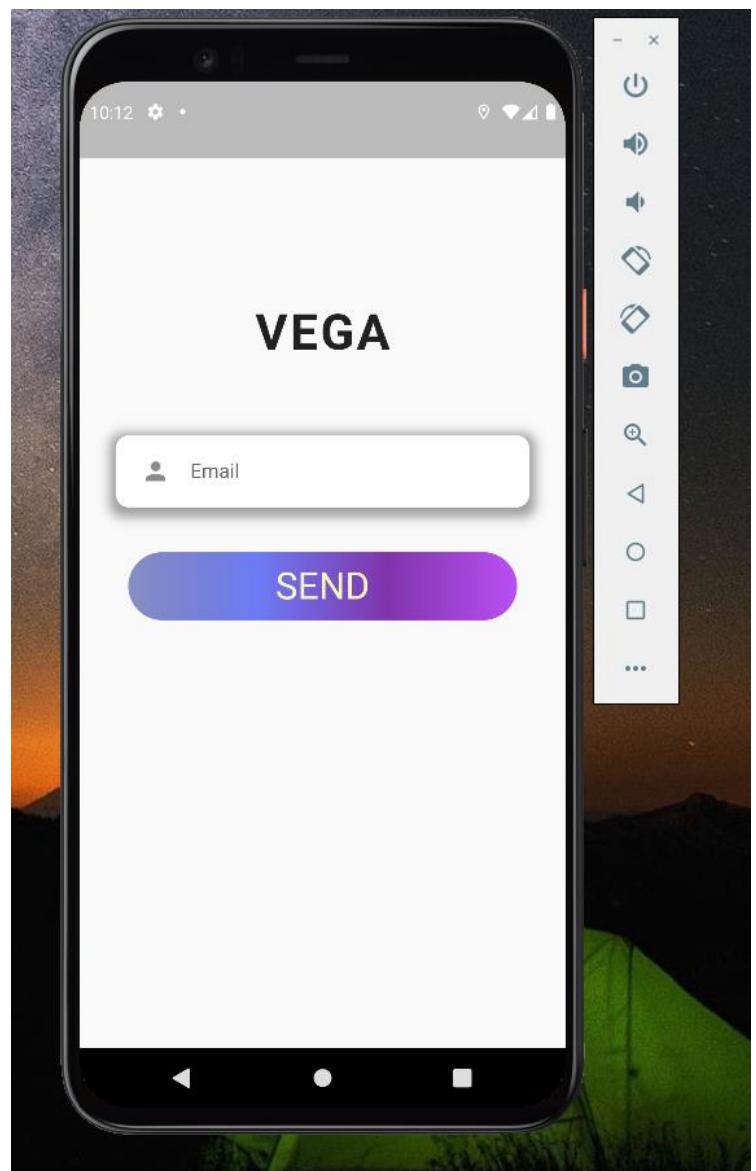
5.b.1) Login Page



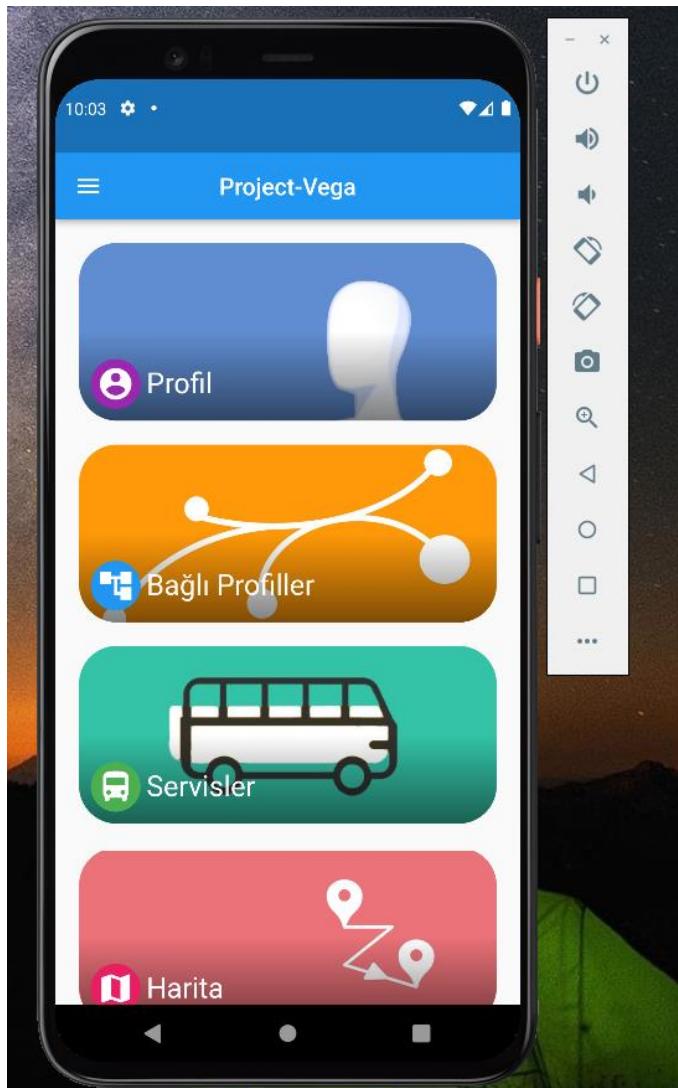
5.c.1) Sign Up Page



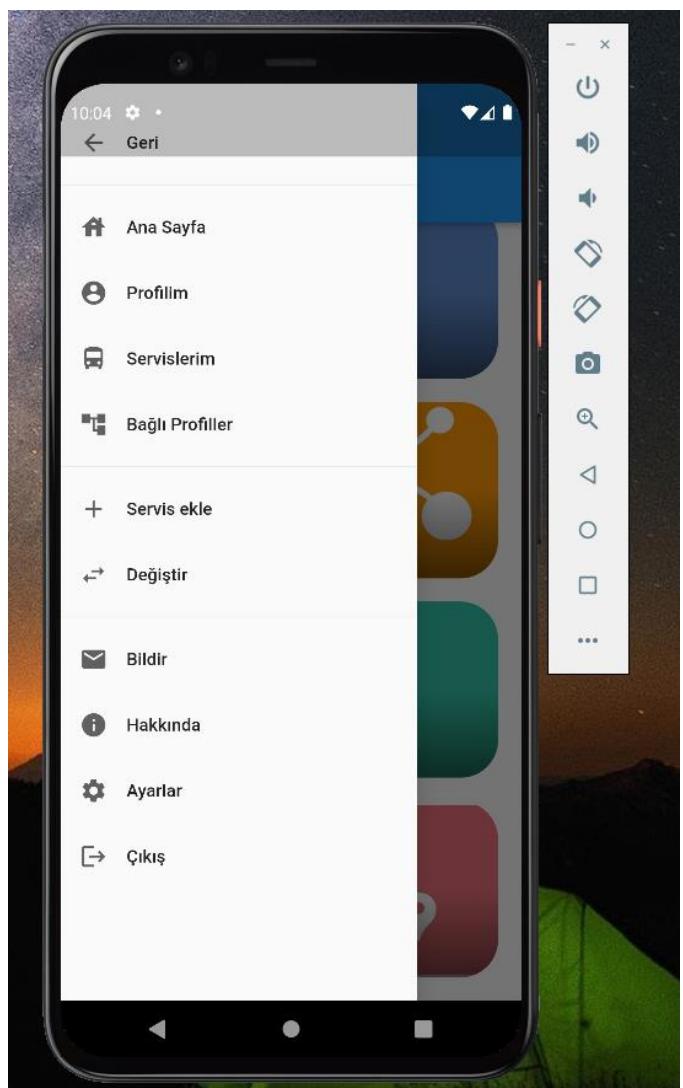
5.d.1) Forgot Password Page



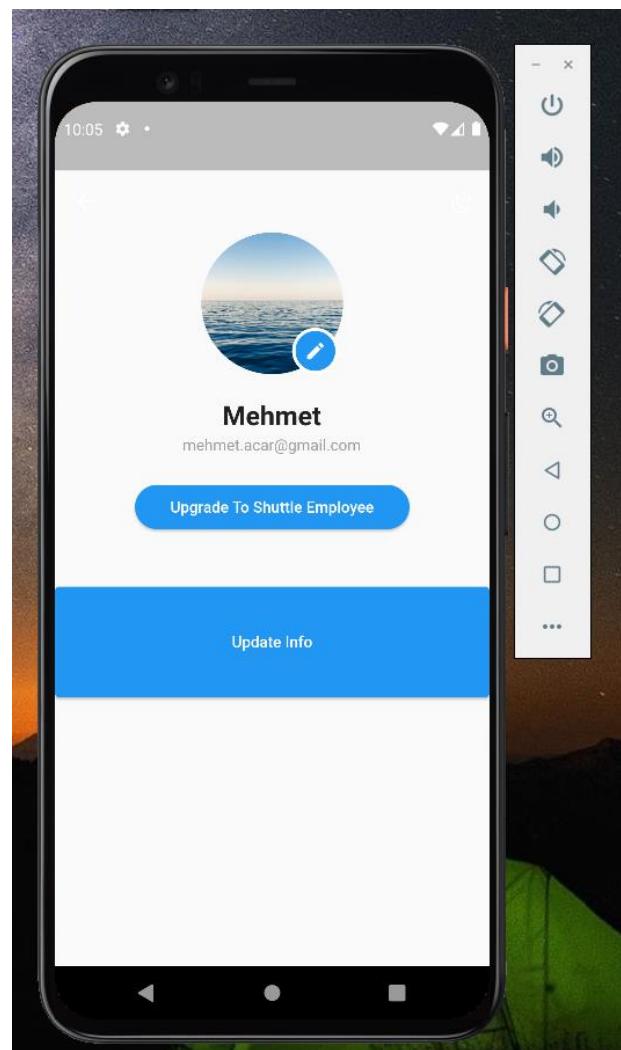
5.e.1 Main Page



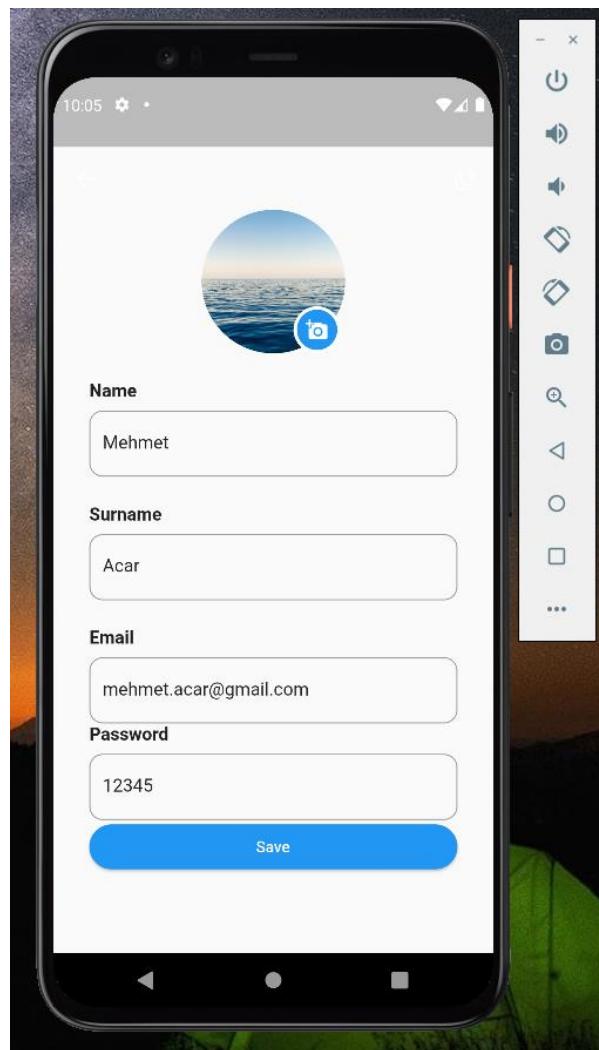
5.e.2 Main Page (Slide Screen )



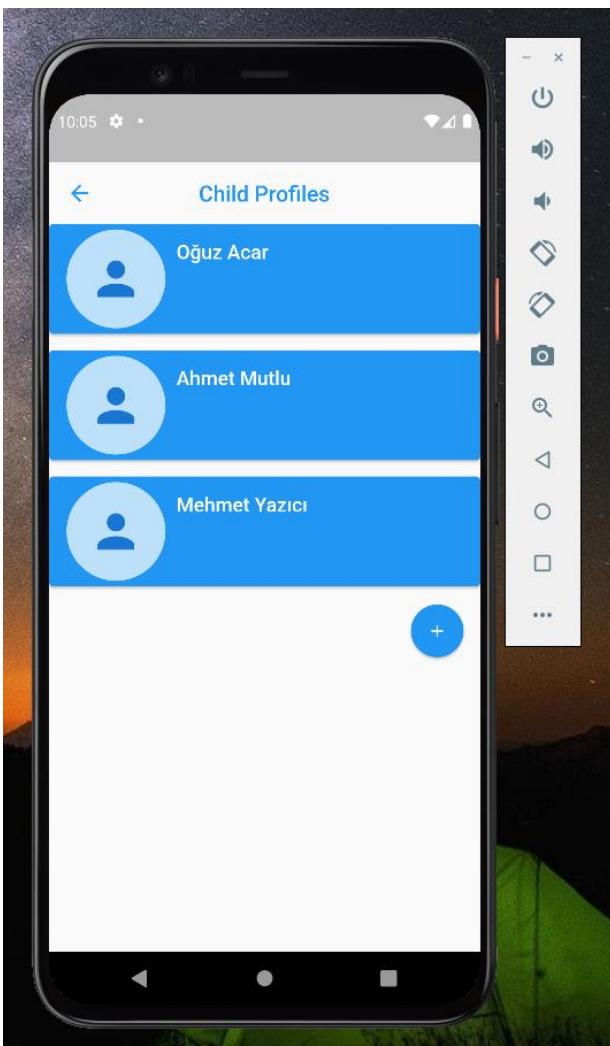
## 5.f Profile Page



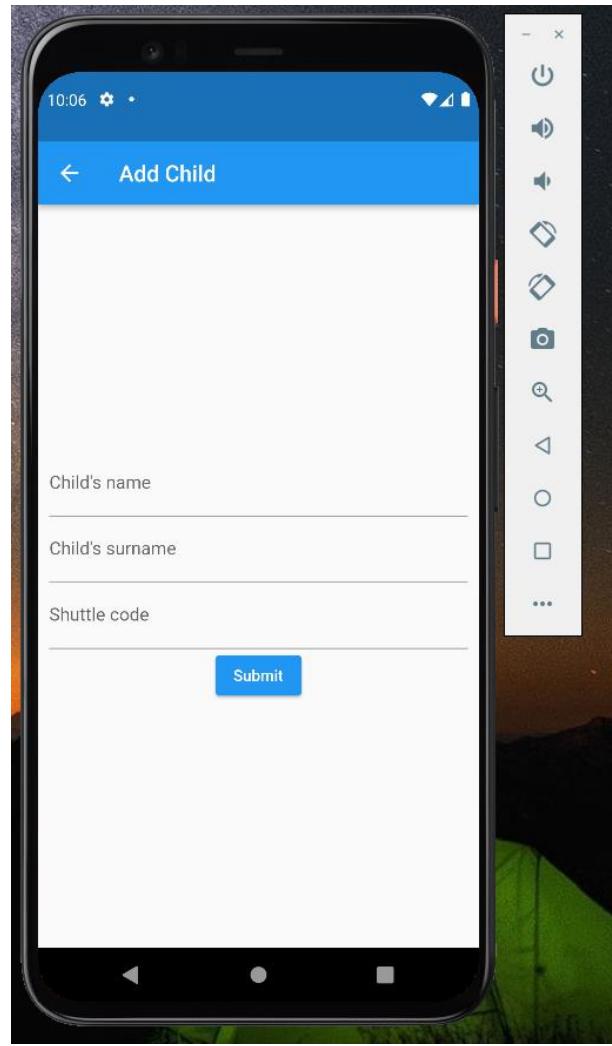
## 5.f.2 Profile Page (Update)



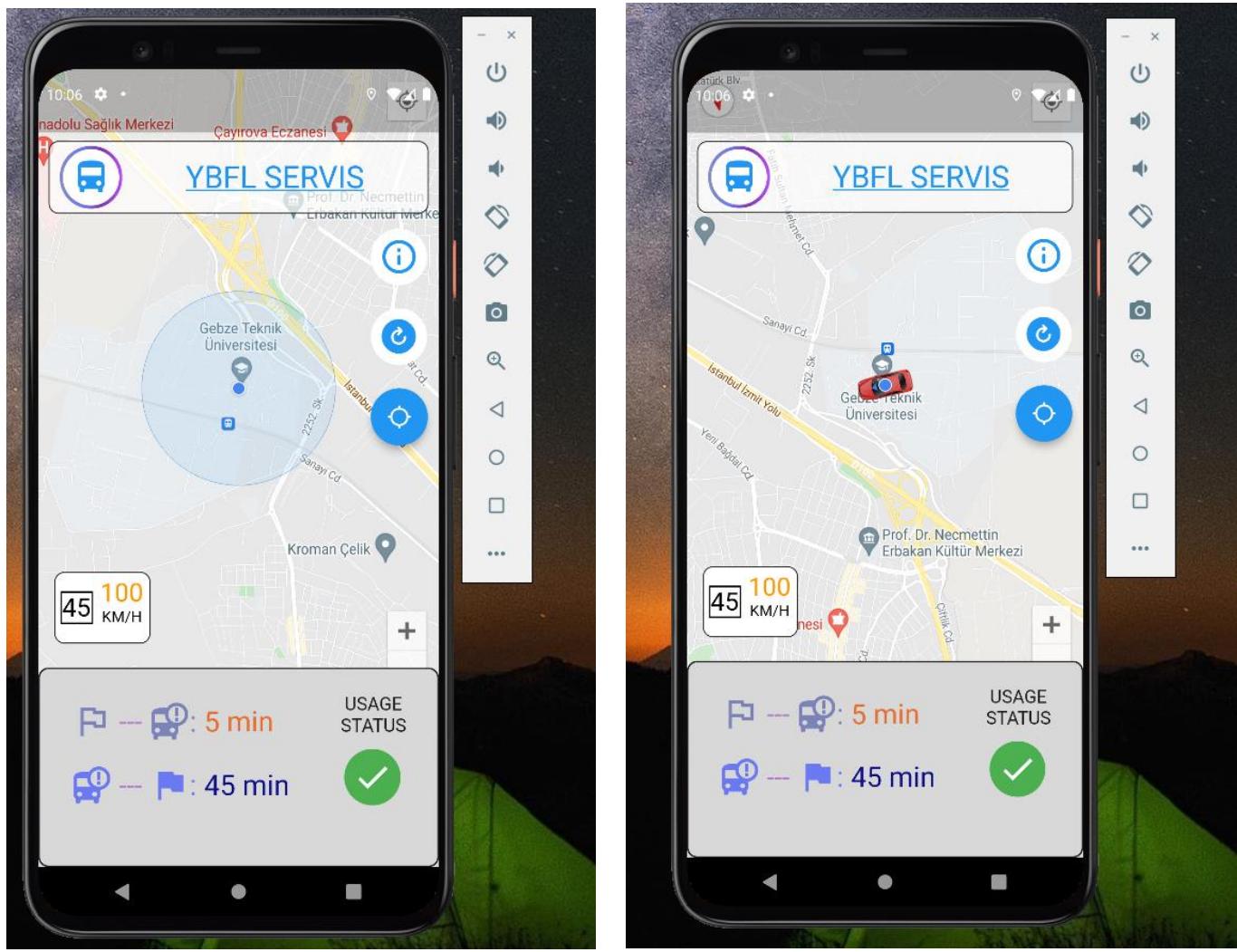
### 5.g.1 Child Profile



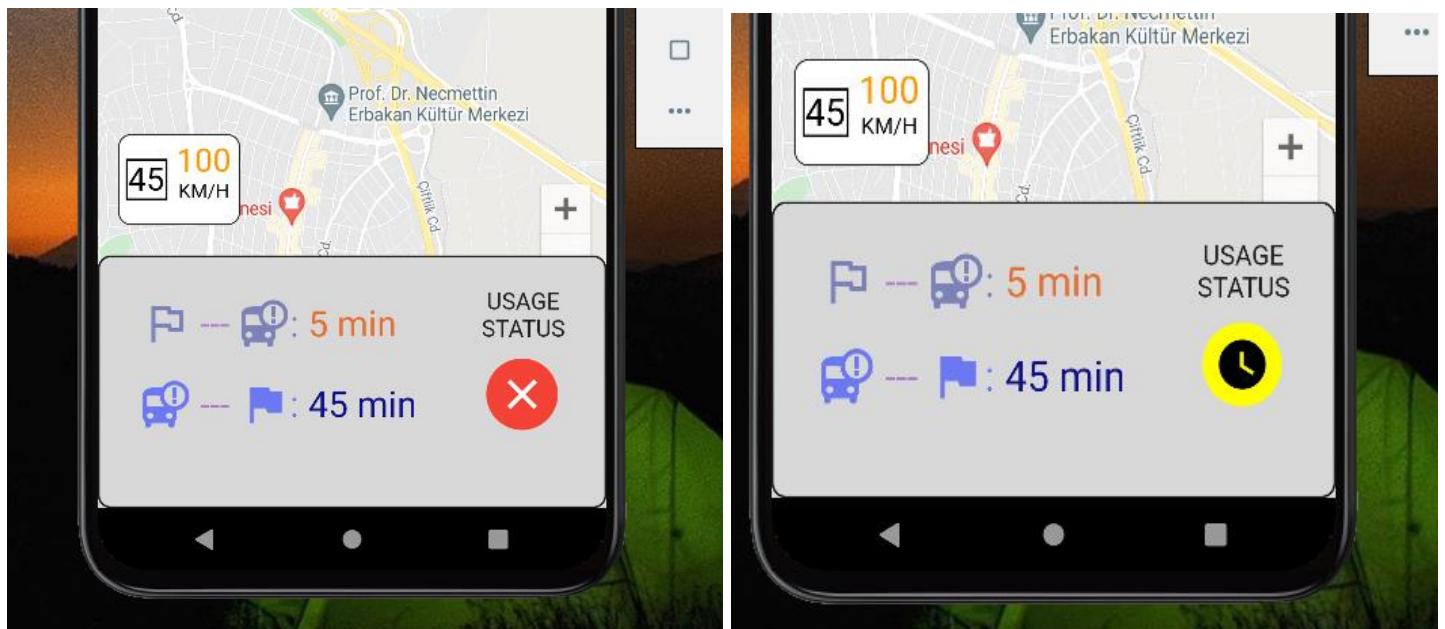
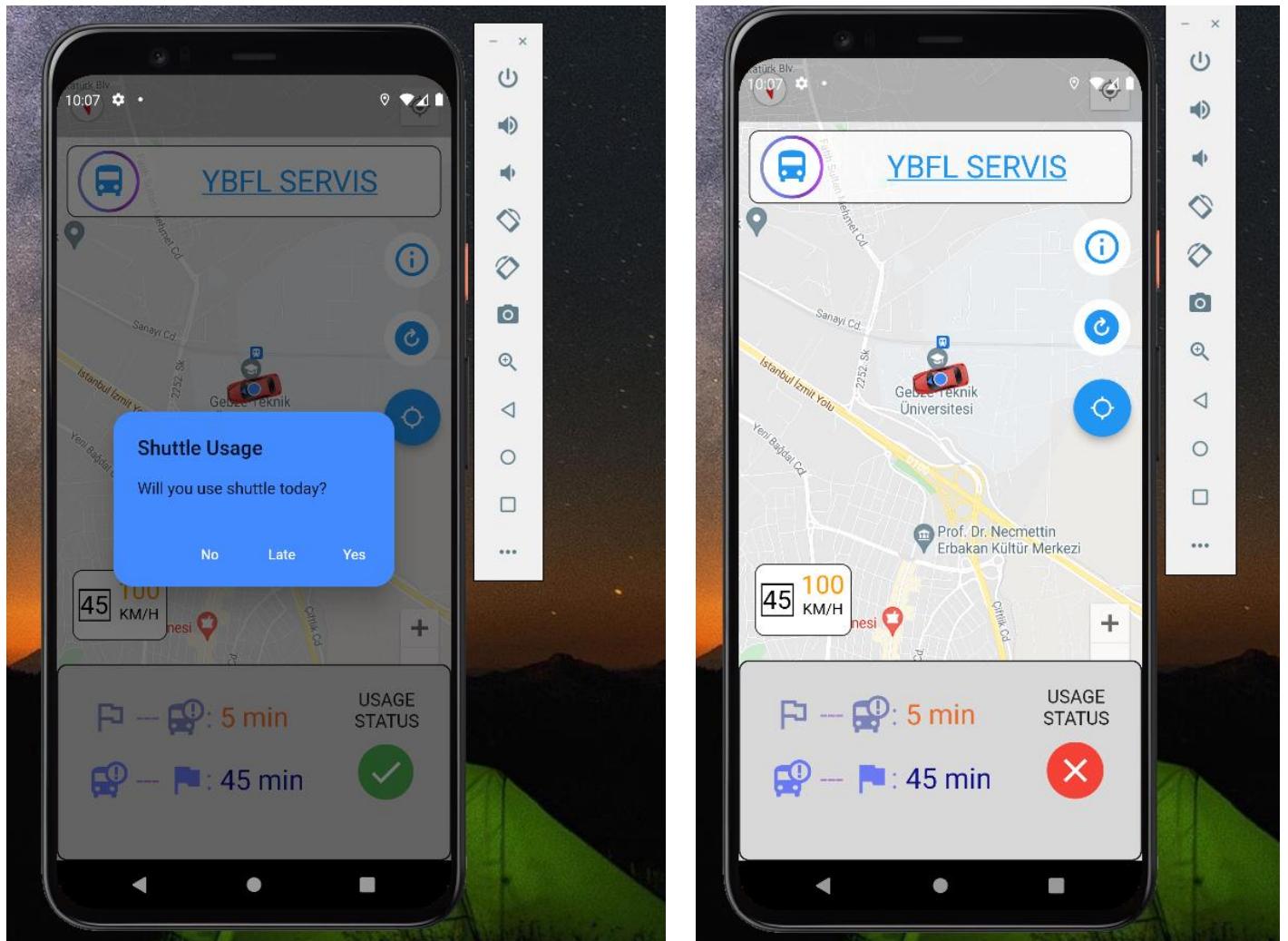
### 5.g.2 Child Profile (Add)



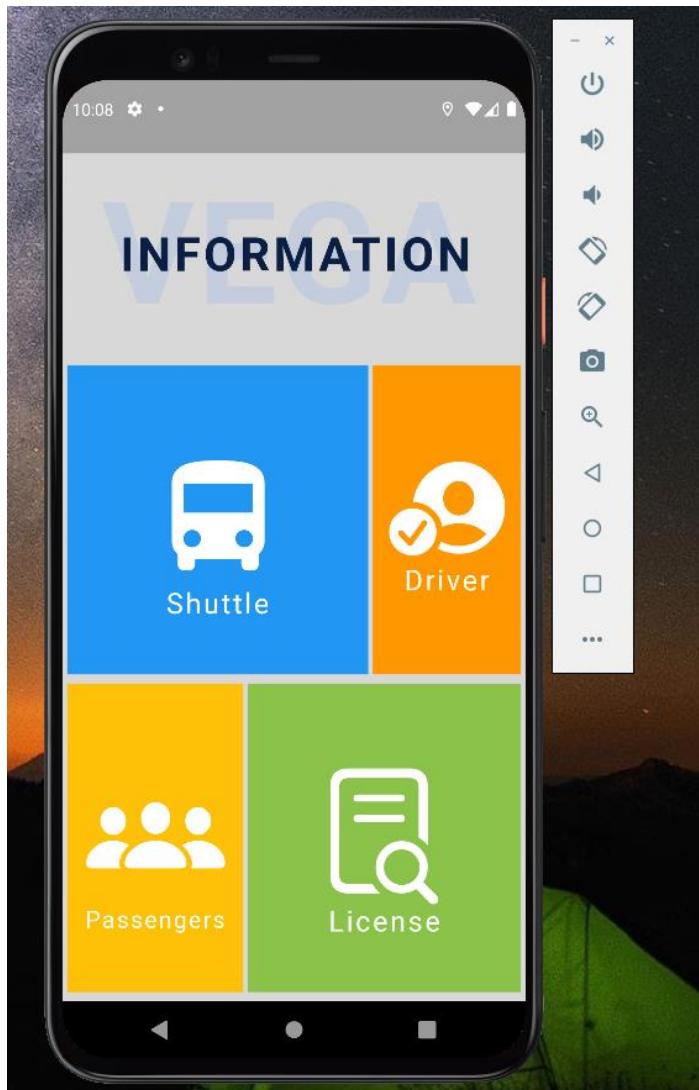
## 5.h.1 My Shuttles Page



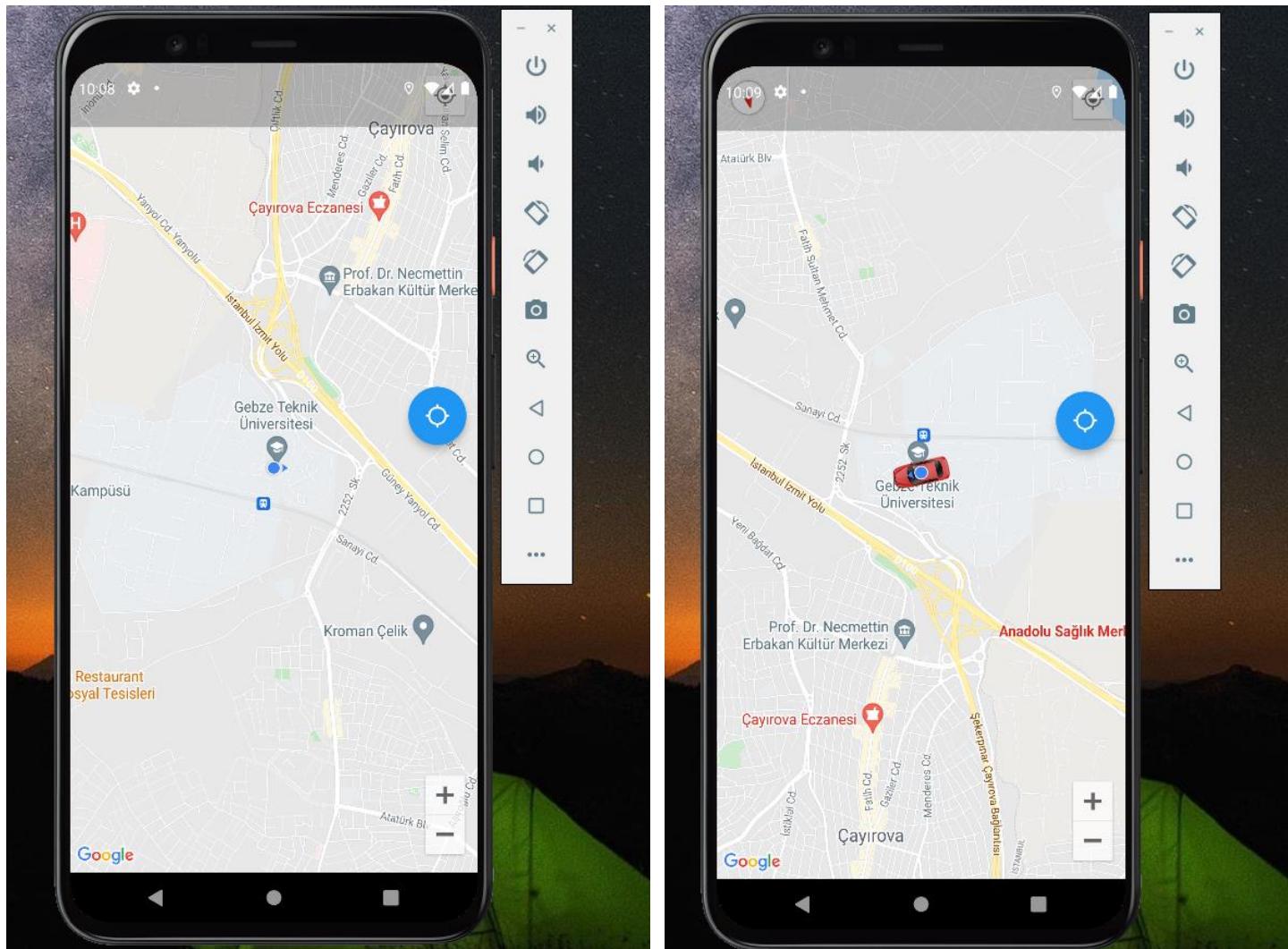
### 5.h.3 Usage Status of Shuttle



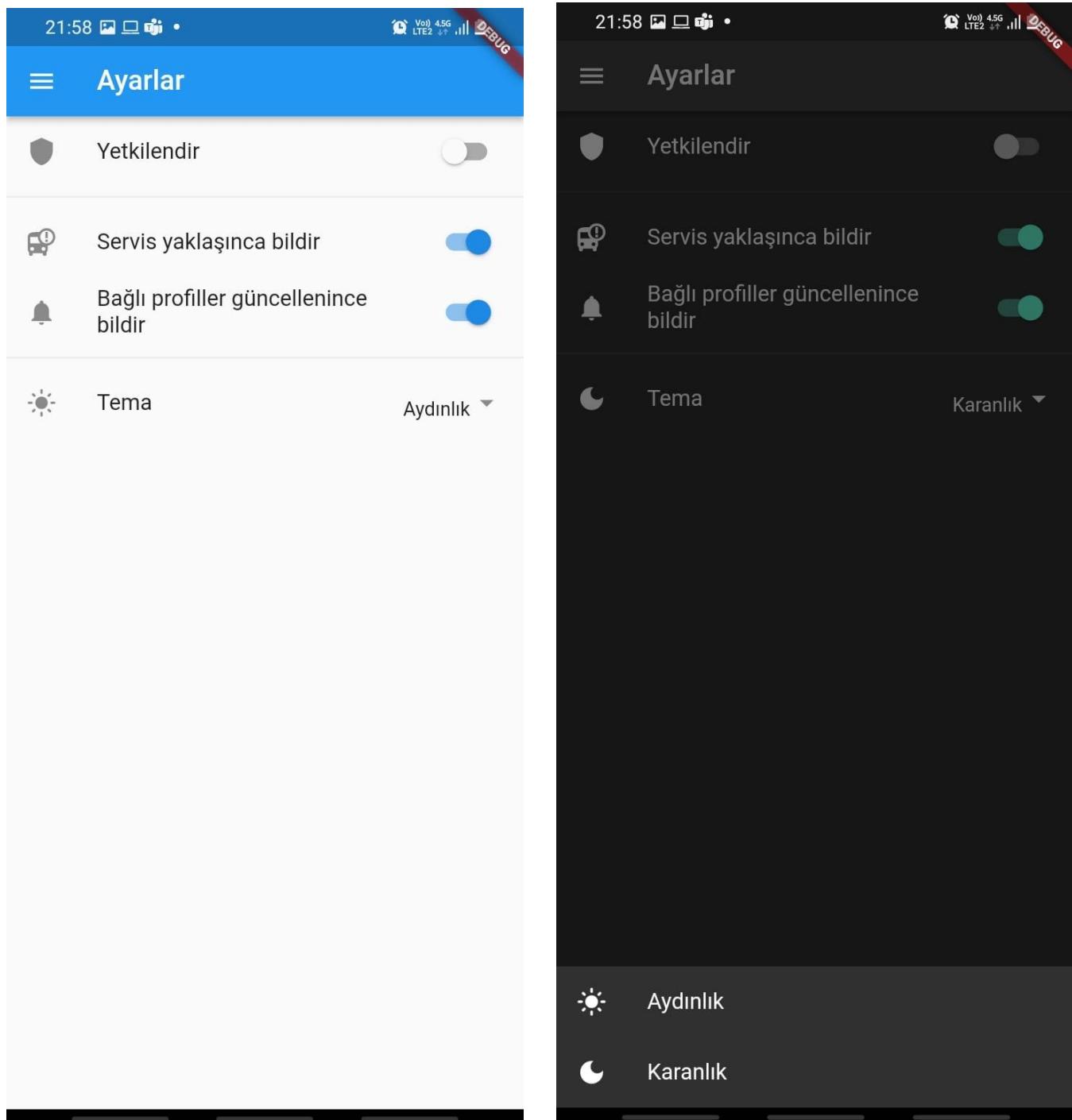
### 5.i.1 Shuttle Info Page



### 5.j.1 Map Screen



## 5.k.1 Settings Screen and Notifications



## 6. Test Cases

### 6.1. Test Case (Signing Up Successfully)

**Description:** An unregistered user should be able to successfully sign up.

**Test Steps:**

1. Enter valid email address
2. Enter at least 6 character password
3. Enter name
4. Enter surname
5. Click signup button

The screenshot shows a mobile application interface. At the top, there is a navigation bar with icons for power, volume, and camera. Below the navigation bar is a header with the text "Authentication". Underneath the header, there is a toolbar with tabs: "Users" (which is selected), "Sign-in method", "Templates", and "Usage". A note below the toolbar says "Prototype and test end-to-end with the Local Emulator Suite, now with Firebase Authentication".

The main area of the screen is a table titled "Users" showing a list of registered users. The columns are "Identifier", "Providers", "Created", "Signed In", and "User UID". The data in the table is as follows:

Identifier	Providers	Created	Signed In	User UID
helloworld@gmail.com	✉️	Nov 9, 2021	Nov 9, 2021	dqtH8...
oguz1@gmail.com	✉️	Nov 9, 2021	Nov 9, 2021	o4ZW...
ogz@gmail.com	✉️	Nov 4, 2021	Nov 4, 2021	7SAQ...
oguz@gmail.com	✉️	Nov 4, 2021	Nov 4, 2021	aIGv2...

At the bottom of the screen, there is a "Sign Up" button. To the right of the mobile screen, there is a vertical toolbar with various icons for navigation and settings.

**Result:** Signing up successful

## 6.2. Test Case (Invalid E-Mail Registration)

**Description:** Check response when valid email is entered and email badly formatted error

**Test Steps:**

1. Enter invalid email address
2. Enter at least 6 character password
3. Enter name
4. Enter surname
5. Click signup button

VEGA ▾

Go to docs



V

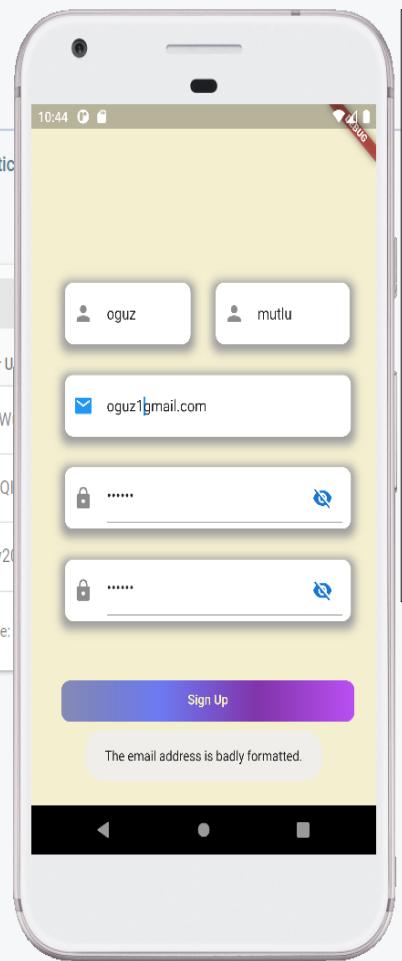
### Authentication

Users Sign-in method Templates Usage

❖ Prototype and test end-to-end with the Local Emulator Suite, now with Firebase Authentication

Search by email address, phone number, or user UID				
Identifier	Providers	Created	Signed In	User U
oguz1@gmail.com	✉	Nov 9, 2021	Nov 9, 2021	o4ZW
ogz@gmail.com	✉	Nov 4, 2021	Nov 4, 2021	7SAQ
oguz@gmail.com	✉	Nov 4, 2021	Nov 4, 2021	aiGv2

Rows per page:



**Result:** “The email address is badly formatted” error occurs in the bottom of the page and signing up is not successful.

## 6.3. Test Case (Already in Use E-Mail Registration)

**Description:** Check response when email is used by another user (unique email)

**Test Steps:**

- 1.Enter used email address
- 2.Enter at least 6 character password
- 3.Enter name
- 4.Enter surname
- 5.Click signup button

VEGA ▾

Go to docs



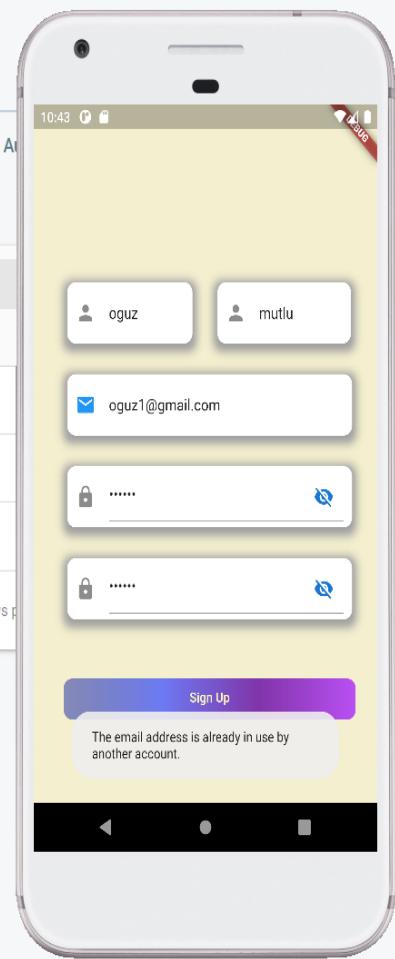
V

### Authentication

Users Sign-in method Templates Usage

❖ Prototype and test end-to-end with the Local Emulator Suite, now with Firebase A

Search by email address, phone number, or user UID			
Identifier	Providers	Created	Signed In
oguz1@gmail.com	✉	Nov 9, 2021	Nov 9, 2021
ogz@gmail.com	✉	Nov 4, 2021	Nov 4, 2021
oguz@gmail.com	✉	Nov 4, 2021	Nov 4, 2021



X

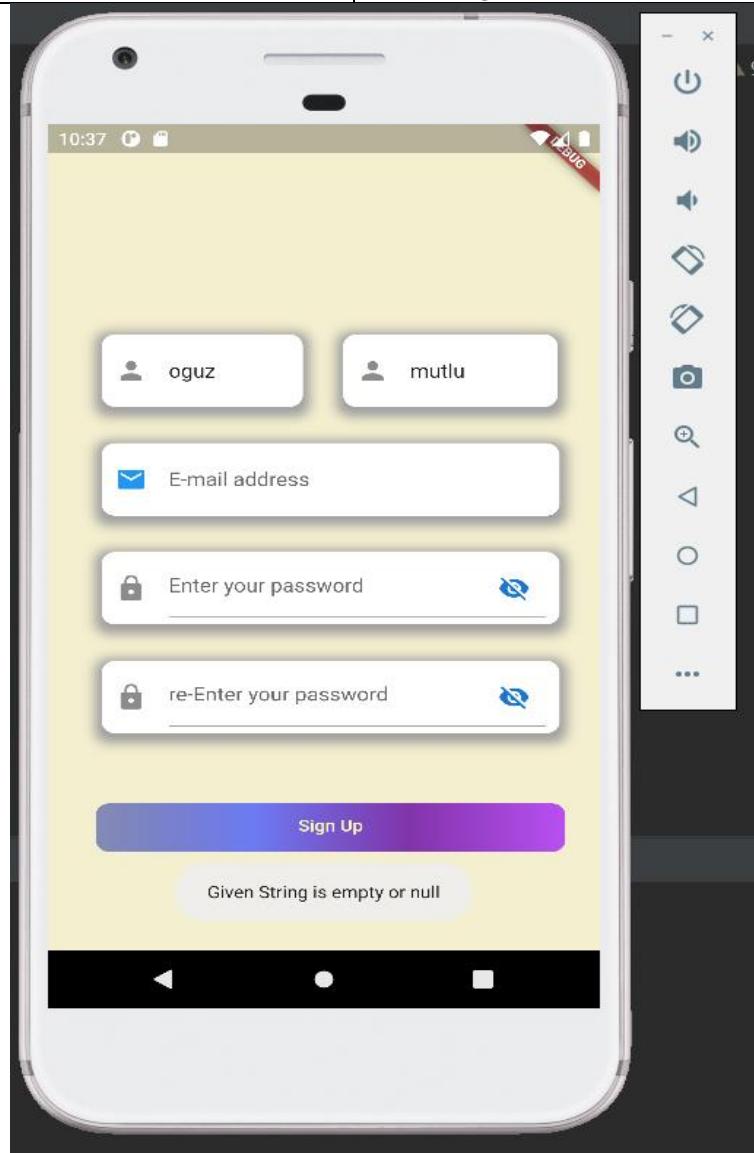
**Result:** “The email address is already in use by another account “ error occurs in the bottom of the page and signing up is not successful.

## 6.4. Test Case (Empty E-Mail Registration)

**Description:** An empty email text box will not be successfully registered

**Test Steps:**

1. Leave empty email text box
2. Enter at least 6 character password
3. Enter name
4. Enter surname
5. Click signup button



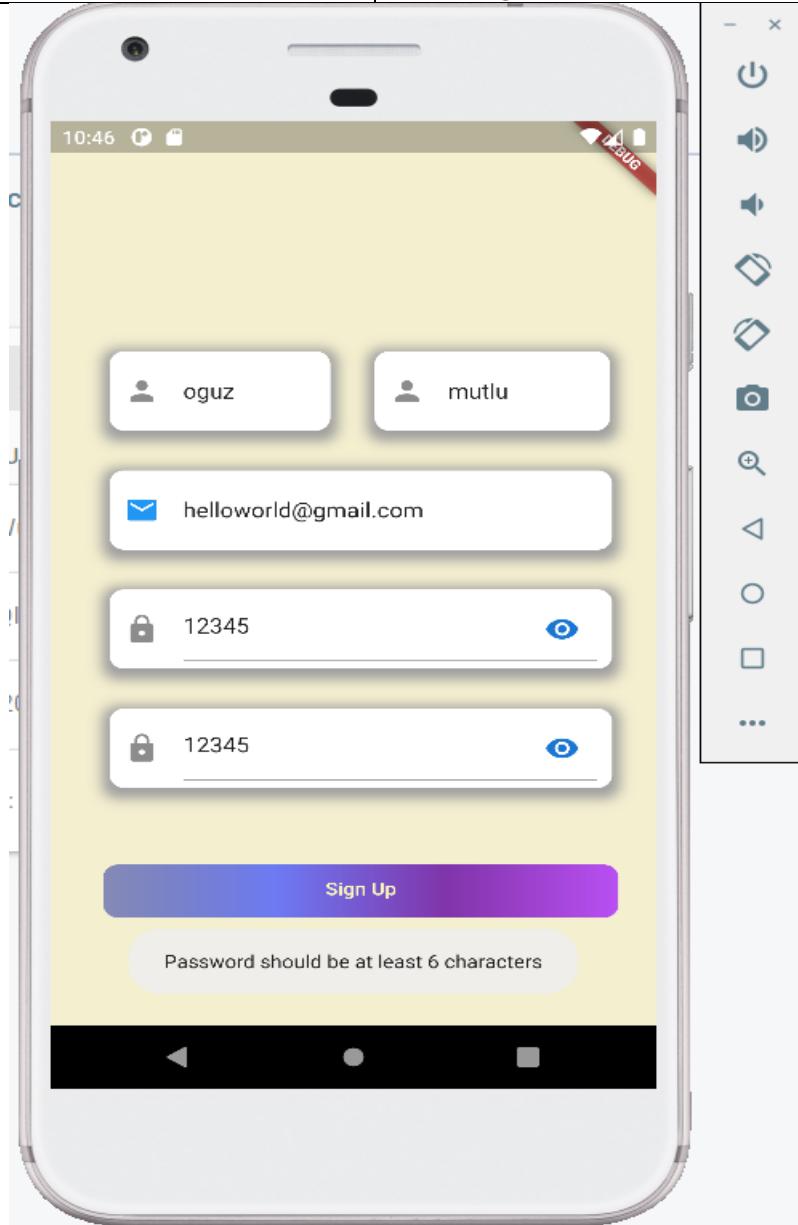
**Result:** "Given String is empty or null" error occurs in the bottom of the page and signing up is not successful.

## 6.5. Test Case (Weak Password Registration)

**Description:** Users cannot register without entering at least 6 character password

**Test Steps:**

1. Enter valid unused email
2. Enter at most 5 character password
3. Enter name
4. Enter surname
5. Click signup button



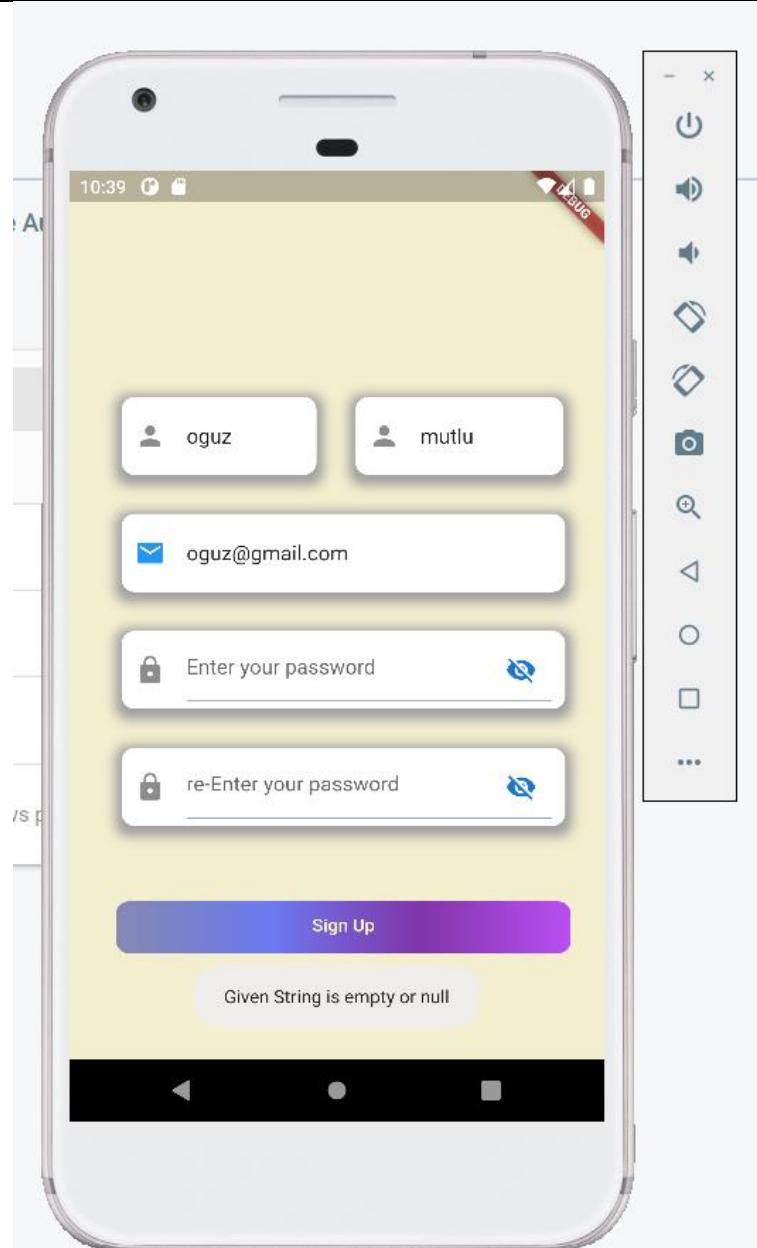
**Result:** “Password should be at least 6 characters” error occurs in the bottom of the page and signing up is not successful.

## 6.6. Test Case (Null/Empty Password Registration)

**Description:** Check if the entered password is blank or not

**Test Steps:**

1. Enter valid unused email
2. Leave password text box empty
3. Enter name
4. Enter surname
5. Click signup button



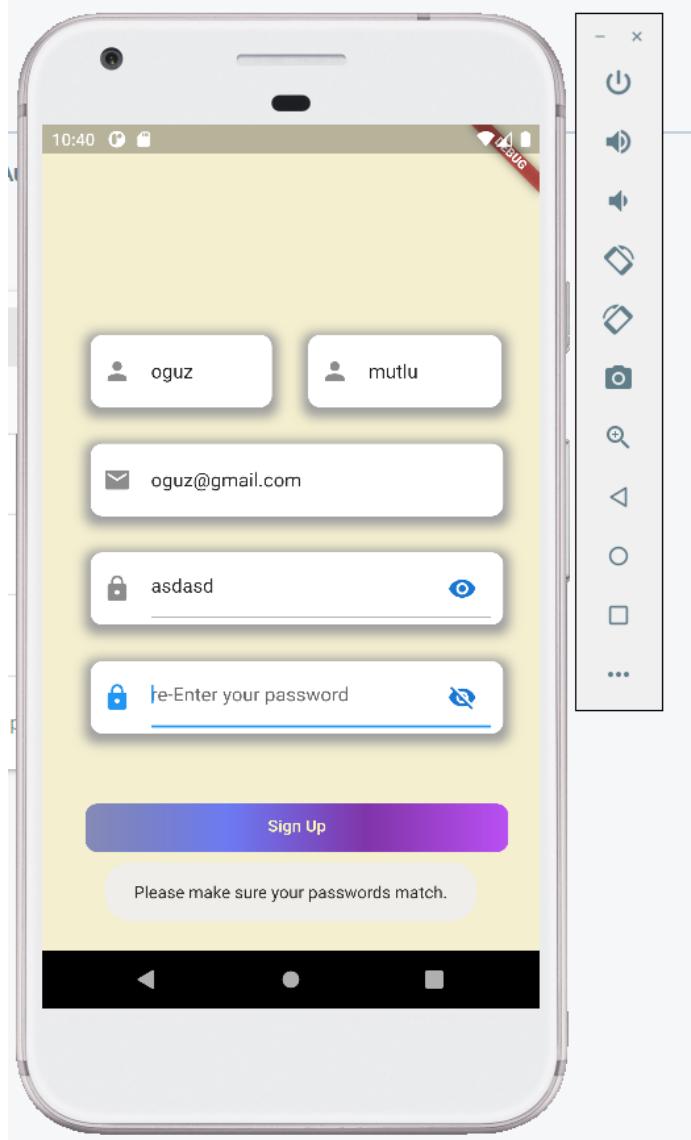
**Result:** "Given String is empty or null" error occurs in the bottom of the page and signing up is not successful.

## 6.7. Test Case (Doesn't Match Password Registration (NULL))

**Description:** Check if the entered password is blank or not

**Test Steps:**

1. Enter valid unused email
2. Enter valid password
3. Leave re-enter password text box empty
4. Enter name
5. Enter Surname
6. Click Signup button



**Result:** “Please make sure your passwords match“ error occurs in the bottom of the page and signing up is not successful.

## 6.8. Test Case (Doesn't Match Password Registration)

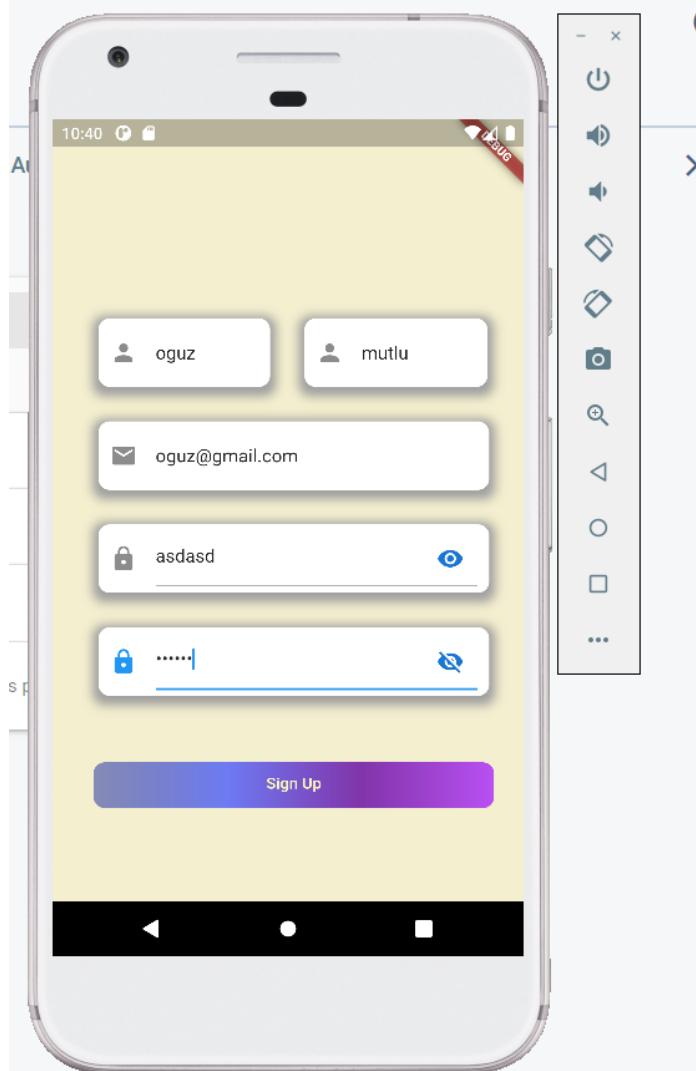
Description:	Test Steps:
Check if the password and re-entered passwords are same or not	<ol style="list-style-type: none"><li>1. Enter valid unused email</li><li>2. Enter valid password</li><li>3. Re-enter password not same as previous</li><li>4. Enter name</li><li>5. Enter Surname</li><li>6. Click Signup button</li></ol>
<p><b>Result:</b> "Please make sure your passwords match" error occurs in the bottom of the page and signing up is not successful.</p>	

## 6.9. Test Case (Visibility On/Off icon Test )

**Description:** Check if visibility on/off icon works successfully

**Test Steps:**

1. Enter valid unused email
2. Enter valid password
3. Re-enter password not same as previous
4. Enter name
5. Enter Surname
6. Click visibility on button
7. Click visibility off button
8. Click Signup button



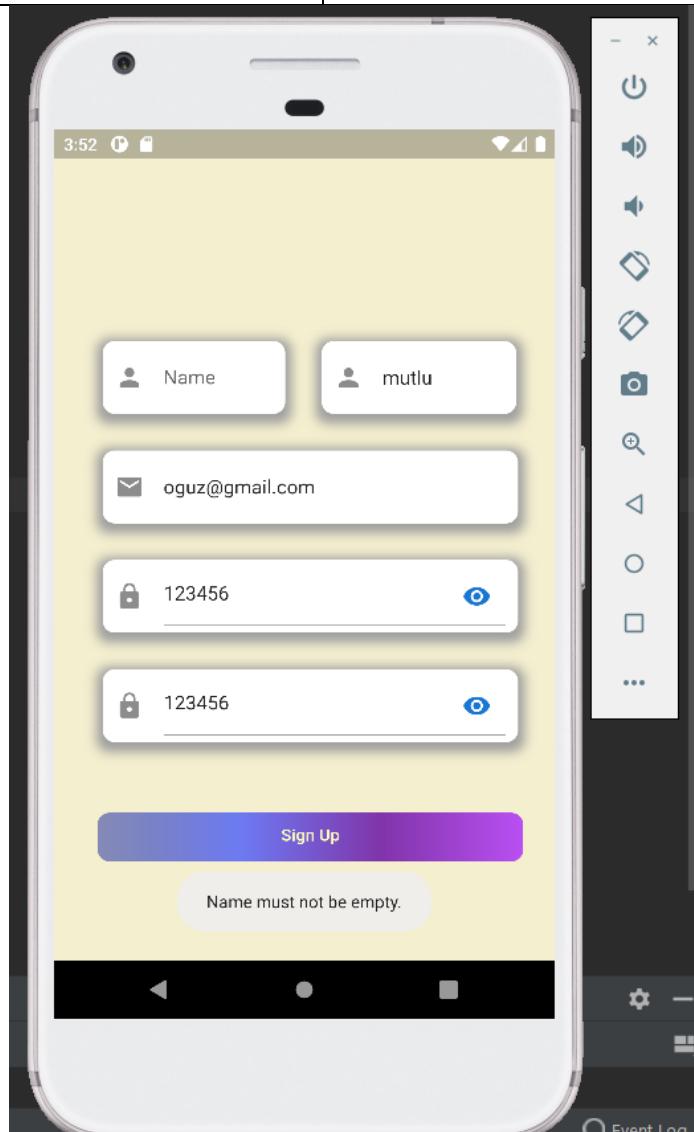
**Result:** When you click the icon, the password will appear, when you click it again, the password will be hidden.

## 6.10. Test Case (Null/Empty Name Entrance Registration)

**Description:** Check if the entered name is blank or not

**Test Steps:**

1. Enter valid unused email
2. Enter valid password
3. Re-enter password not same as previous
4. Leave name text box empty
5. Enter Surname
6. Click Signup button



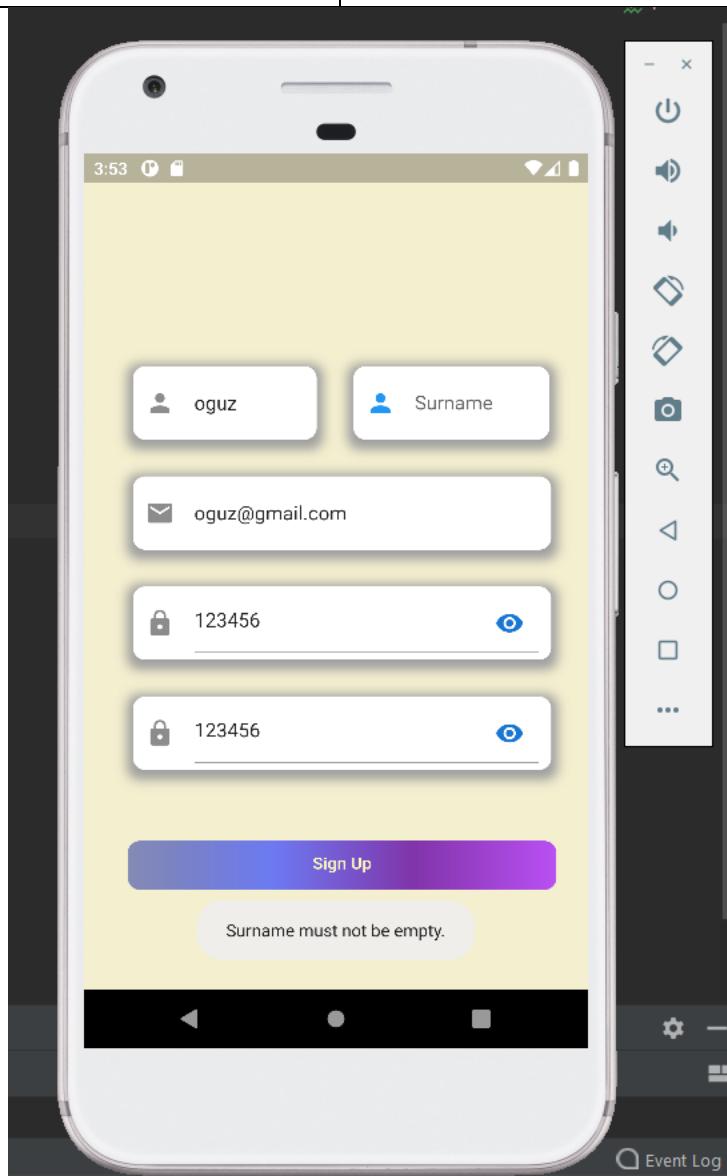
**Result:** “Name must not be empty“ error occurs in the bottom of the page and signing up is not successful.

## 6.11. Test Case (Null/Empty Surname Entrance Registration)

**Description:** Check if the entered surname is blank or not

**Test Steps:**

1. Enter valid unused email
2. Enter valid password
3. Re-enter password not same as previous
4. Enter name
5. Leave surname text box empty
6. Click Signup button



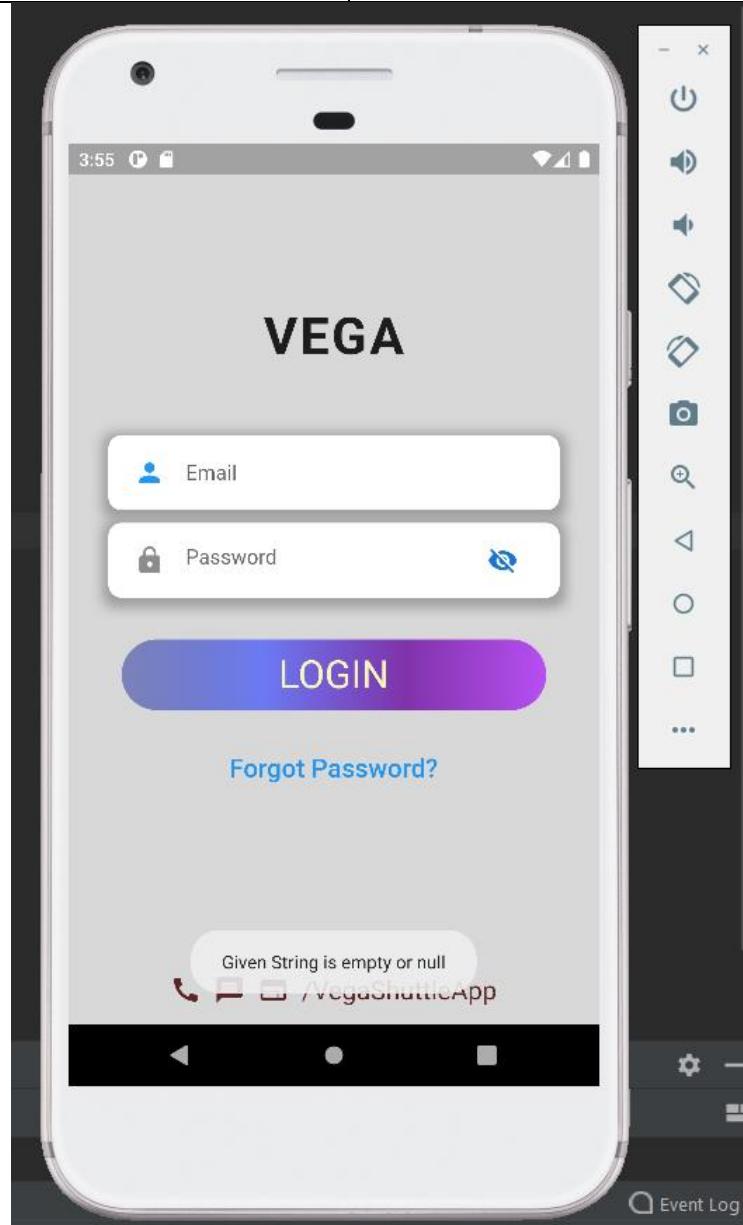
**Result:** "Surname must not be empty" error occurs in the bottom of the page and signing up is not successful.

## 6.12. Test Case (Empty/Null Mail-Password Entrance Login)

**Description:** Check if the both email and password are blank or not

**Test Steps:**

1. Enter login page
2. Leave email text box empty
3. Leave password text box empty
4. Click Login button



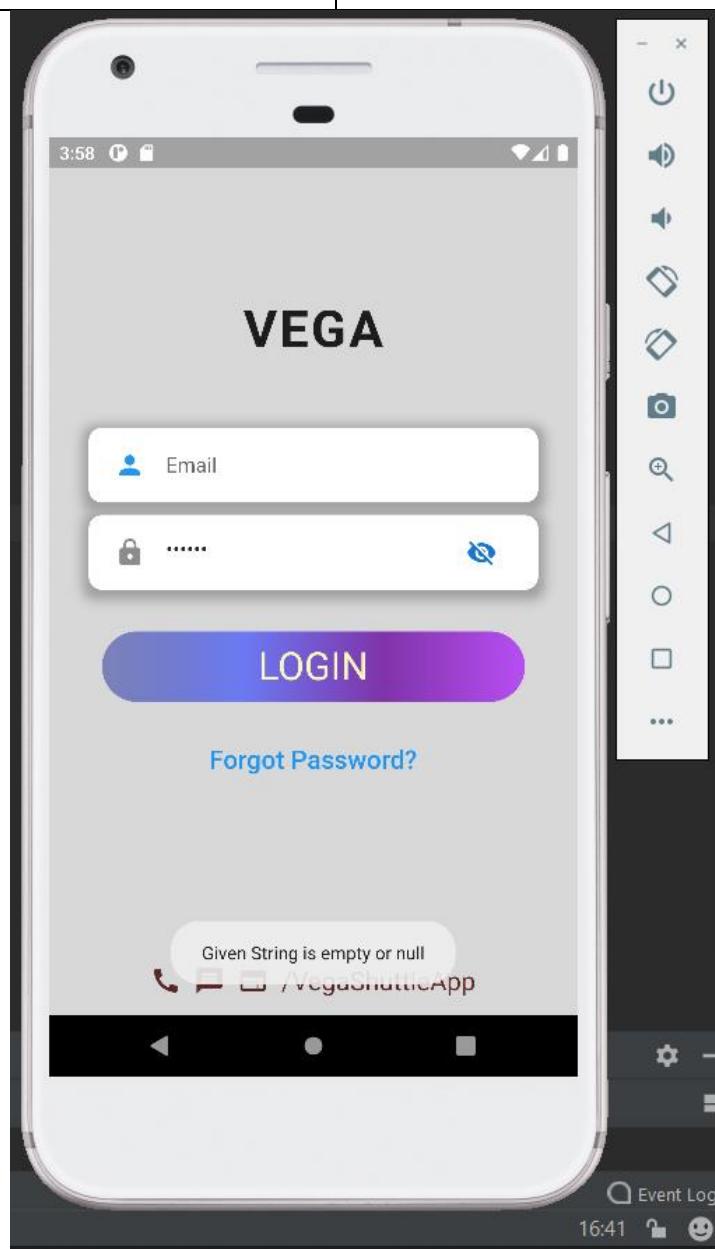
**Result:** "Given string is empty or null" error occurs in the bottom of the page and signing up is not successful.

## 6.13. Test Case (Empty/Null Email Entrance Login)

**Description:** Check if the entered email is blank or not

**Test Steps:**

1. Enter login page
2. Leave email text box empty
3. Enter valid password
4. Click Login button



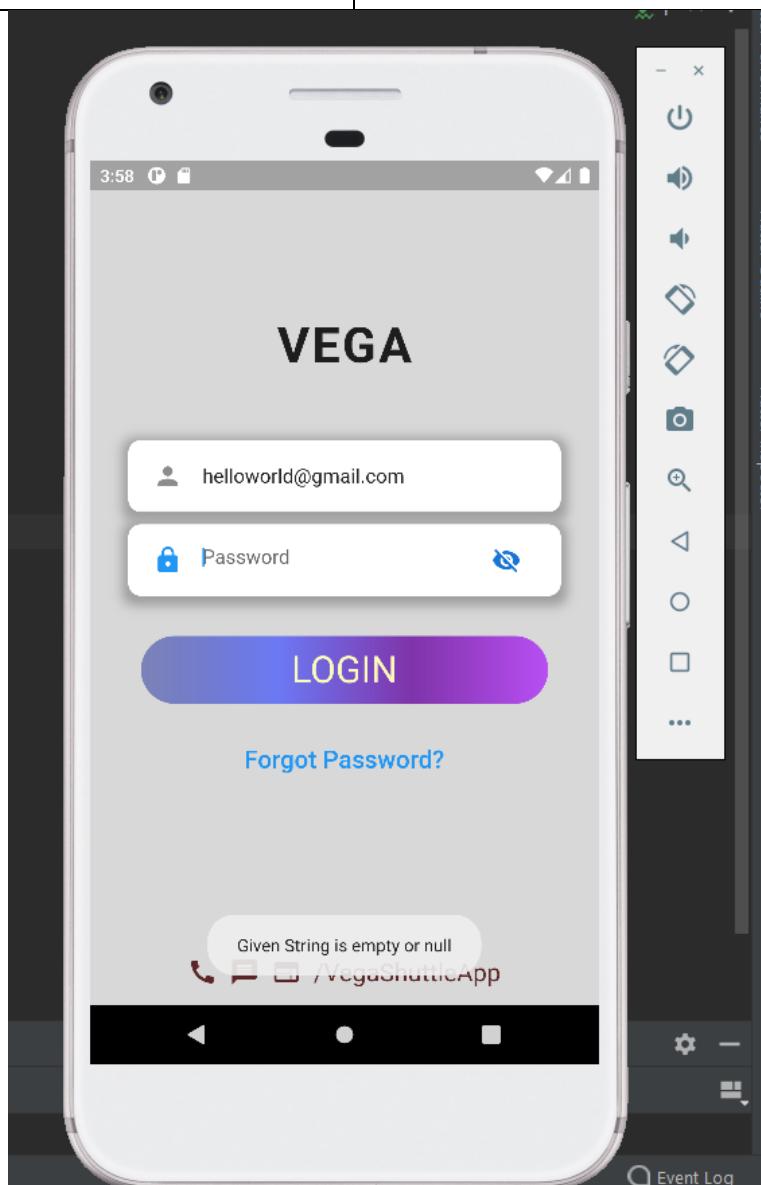
**Result:** “Given string is empty or null” error occurs in the bottom of the page and signing up is not successful.

## 6.14. Test Case (Empty/Null Password Entrance Login)

**Description:** Check if the entered password is blank or not

**Test Steps:**

1. Enter login page
2. Enter valid email
3. Leave password text box empty
4. Click Login button



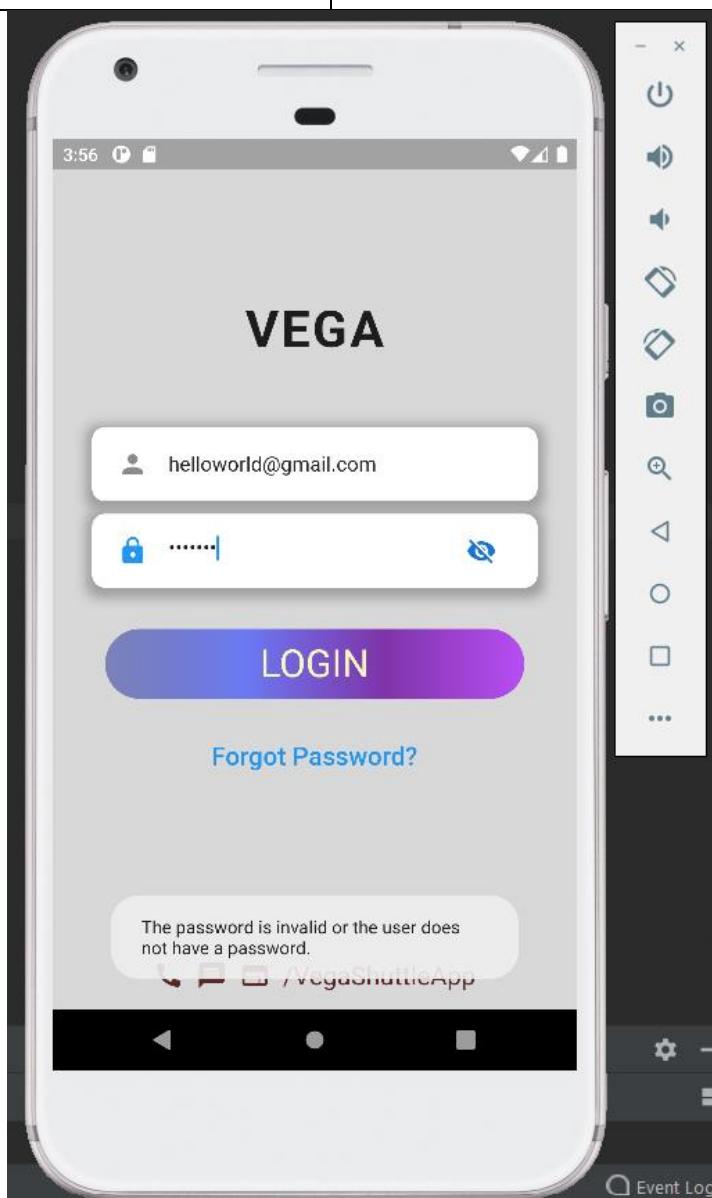
**Result:** "Given string is empty or null" error occurs in the bottom of the page and signing up is not successful.

## 6.15. Test Case (Invalid Password Login)

**Description:** Check if the entered password is true or not

**Test Steps:**

1. Enter login page
2. Enter valid email
3. Enter invalid password
4. Click Login button



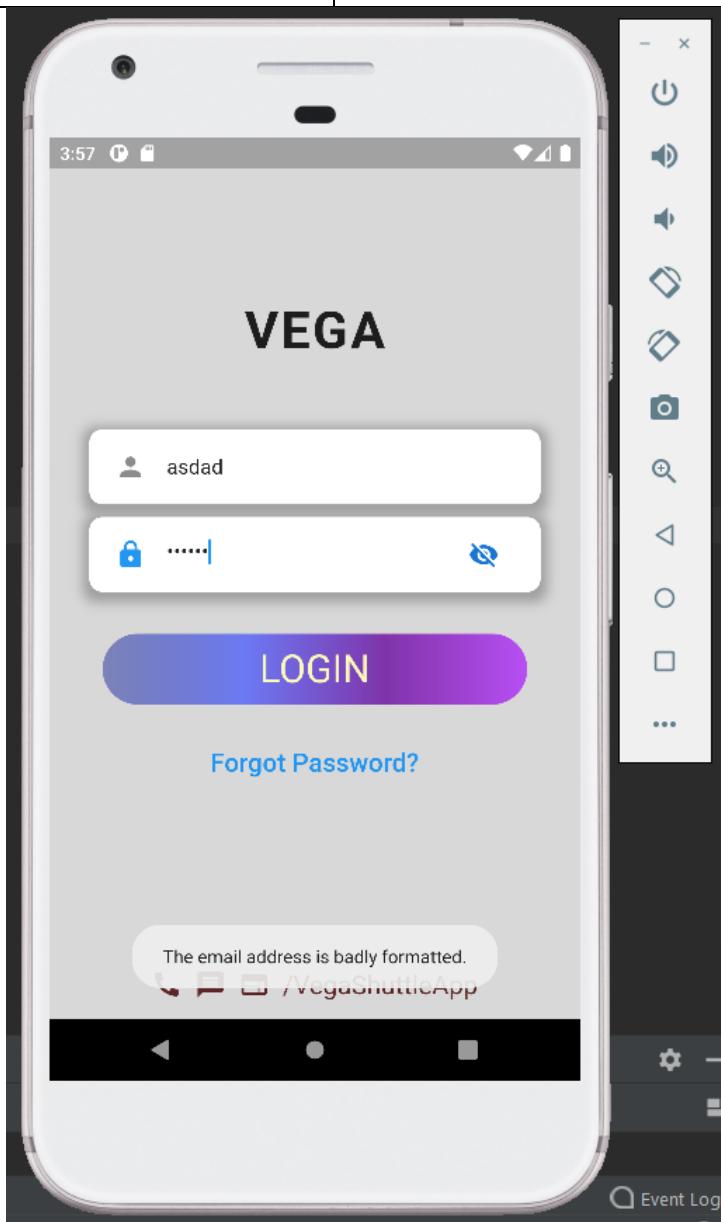
**Result:** “The password is invalid or the user does not have a password“ error occurs in the bottom of the page and signing up is not successful.

## 6.16. Test Case (Invalid Email Login )

**Description:** Check if the entered Email format is true or not

**Test Steps:**

1. Enter login page
2. Enter invalid email
3. Enter valid password
4. Click Login button



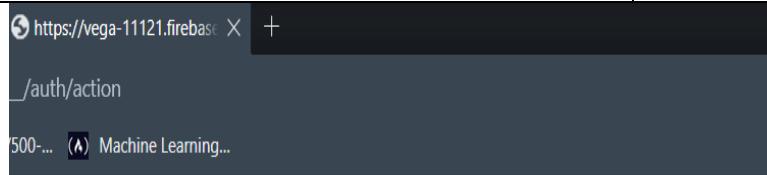
**Result:** "The email address is badly formatted" error occurs in the bottom of the page and signing up is not successful.

## 6.17. Test Case (Reset Password )

**Description:** Checking Forgot Password? sequence and reset password

**Test Steps:**

1. Enter login page
2. Enter invalid password
3. Click Login Button
4. Write your email and click send button
5. Enter your mail account
6. Enter New Password



The image contains two side-by-side screenshots. The left screenshot shows a "Reset your password" form. It has a text input field with the placeholder "New password" and a value "oguzmutlu81@gmail.com" followed by a clear icon (an "X" inside a circle). Below the input field is a blue "SAVE" button. The right screenshot shows a success message box with the title "Password changed" and the text "You can now sign in with your new password".

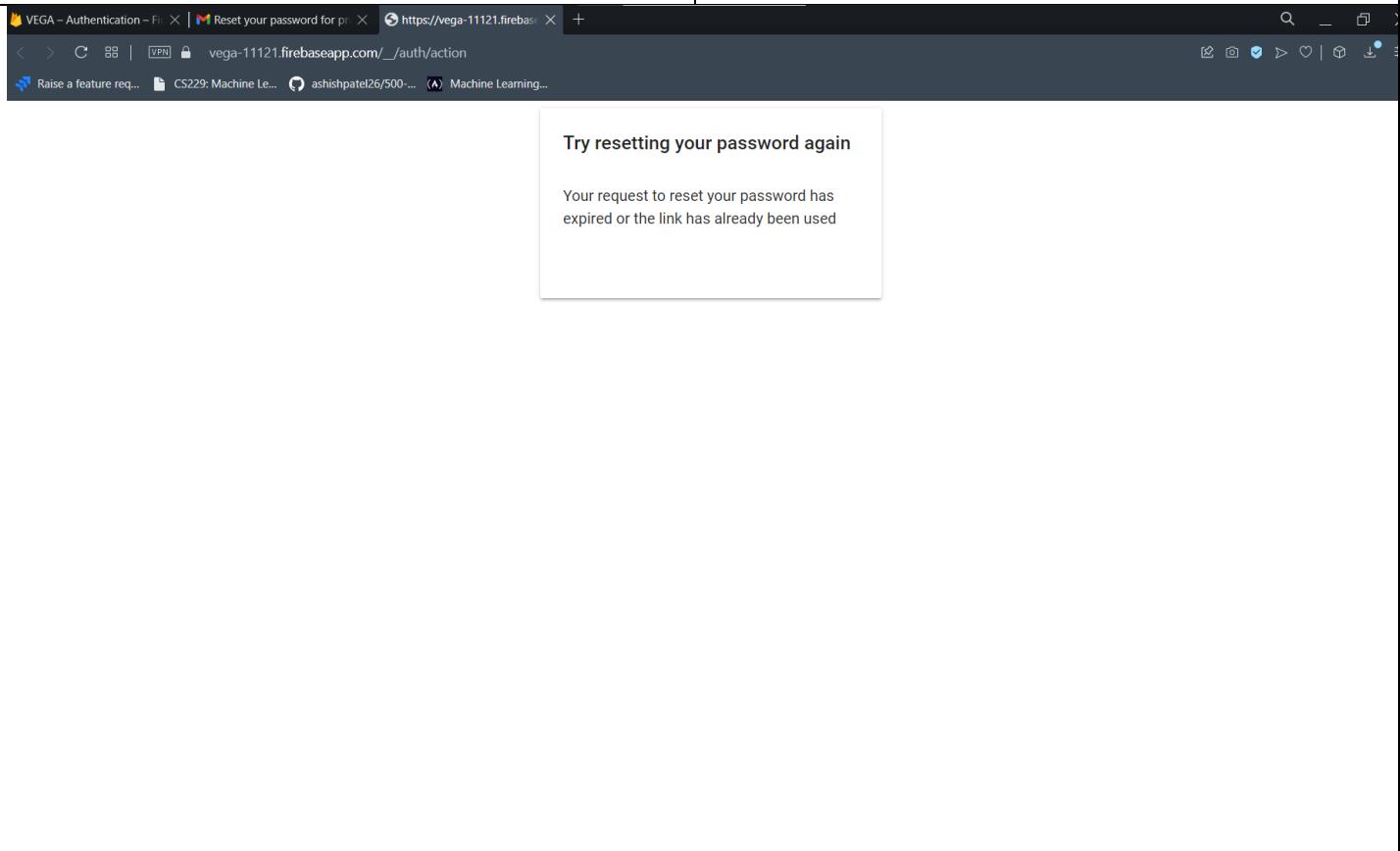
**Result:** Password Successfully changed and sign up successful with new password.

## 6.18. Test Case (Clicking Same Password Reset Link )

**Description:** Checks if reset password link usage is more than one =

**Test Steps:**

1. Enter your mail account
2. Enter again reset password link



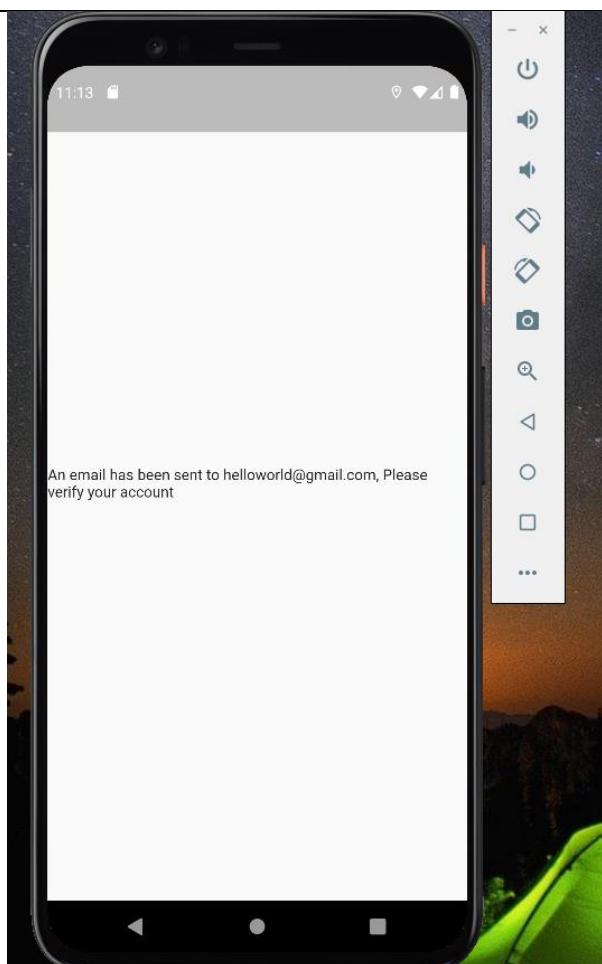
**Result:** “The link has already been used“ error occurs in the top of the page and user must be click again forgot password again.

## 6.19. Test Case (Validate Account )

**Description:** Check if the account is valid or not

**Test Steps:**

1. Enter your mail account
2. Click the link which sent from service
3. Validate your account



Your email has been verified

You can now sign in with your new account

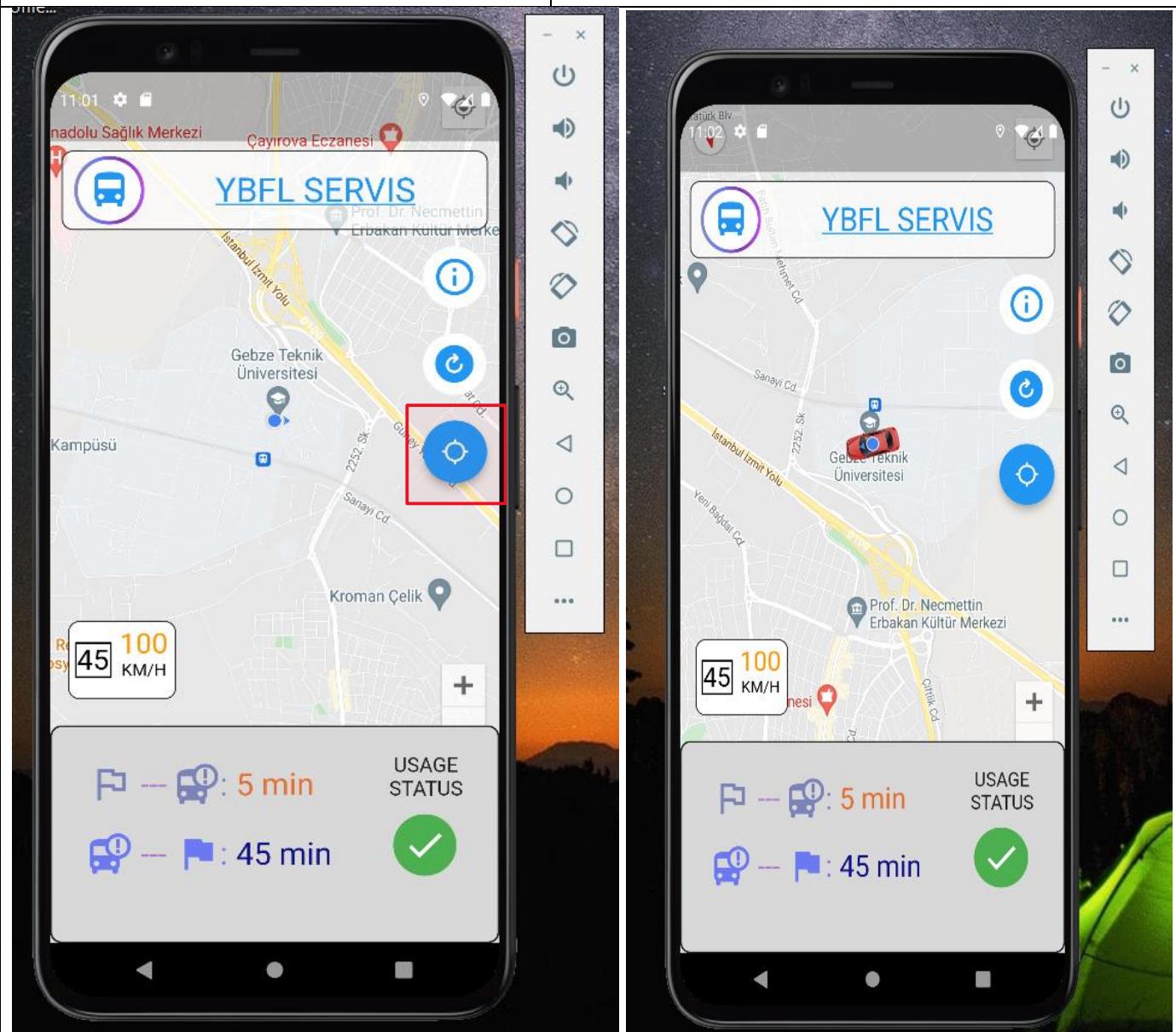
**Result:** Email has been verified now user can sign in with her/his new account

## 6.20. Test Case (Shuttle Live Location Tracking )

**Description:** Check shuttle location by refreshing map

**Test Steps:**

1. Enter application
2. Enter Shuttles Page
3. Click Search Button for display shuttle



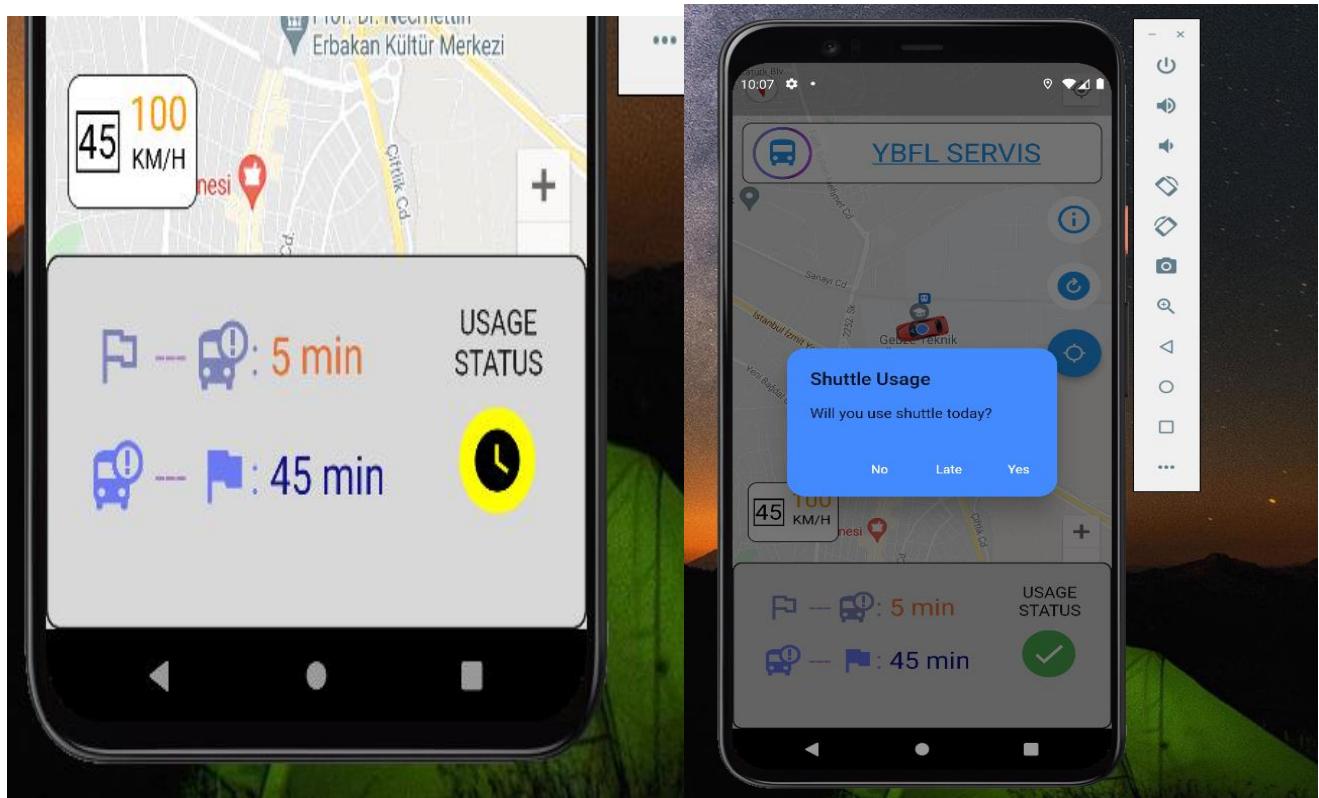
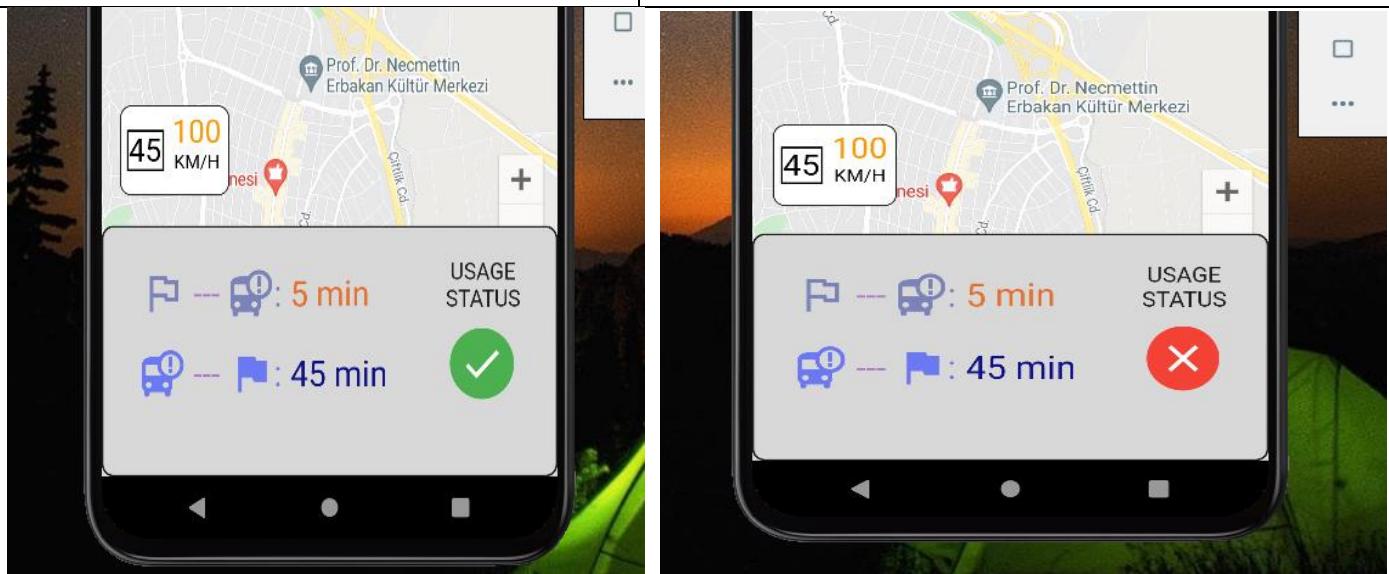
**Result:** "The Shuttle's live location displays in the map page

## 6.21. Test Case (Set Usage Status)

**Description:** Checks users service usage status (decline, late, allow)

**Test Steps:**

1. Enter main page of application
2. Enter My Shuttles Page
3. Set your usage value as decline, delay and accept for today usage



**Result:** Shuttle Employee can see the status with the help of database and plans her/his trips according to the users usage status

## 6.22. Test Case (Already been Used Verification Link )

**Description:** Check if the account is valid or not

**Test Steps:**

1. Enter your mail account
2. Click the link which is old or already used

Try verifying your email again

Your request to verify your email has expired or the link has already been used

 noreply@vega-11121.firebaseio.com 18:30 (4 saat önce)     
Alice: ben ▾

Hello,

Follow this link to verify your email address.

[https://vega-11121.firebaseio.com/\\_auth/action?mode=verifyEmail&oobCode=Paox2w0CrEqzo1Q4hB-Y82f4Alq1Lxrj8p7EqVfxGN0AAAFA9tGqT2w&apiKey=AIzaSyCUuvmA6-u7rlJ9Ubsns6Kx58Two79uogg&lang=en](https://vega-11121.firebaseio.com/_auth/action?mode=verifyEmail&oobCode=Paox2w0CrEqzo1Q4hB-Y82f4Alq1Lxrj8p7EqVfxGN0AAAFA9tGqT2w&apiKey=AIzaSyCUuvmA6-u7rlJ9Ubsns6Kx58Two79uogg&lang=en)

If you didn't ask to verify this address, you can ignore this email.

Thanks,

Your project-999497992779 team

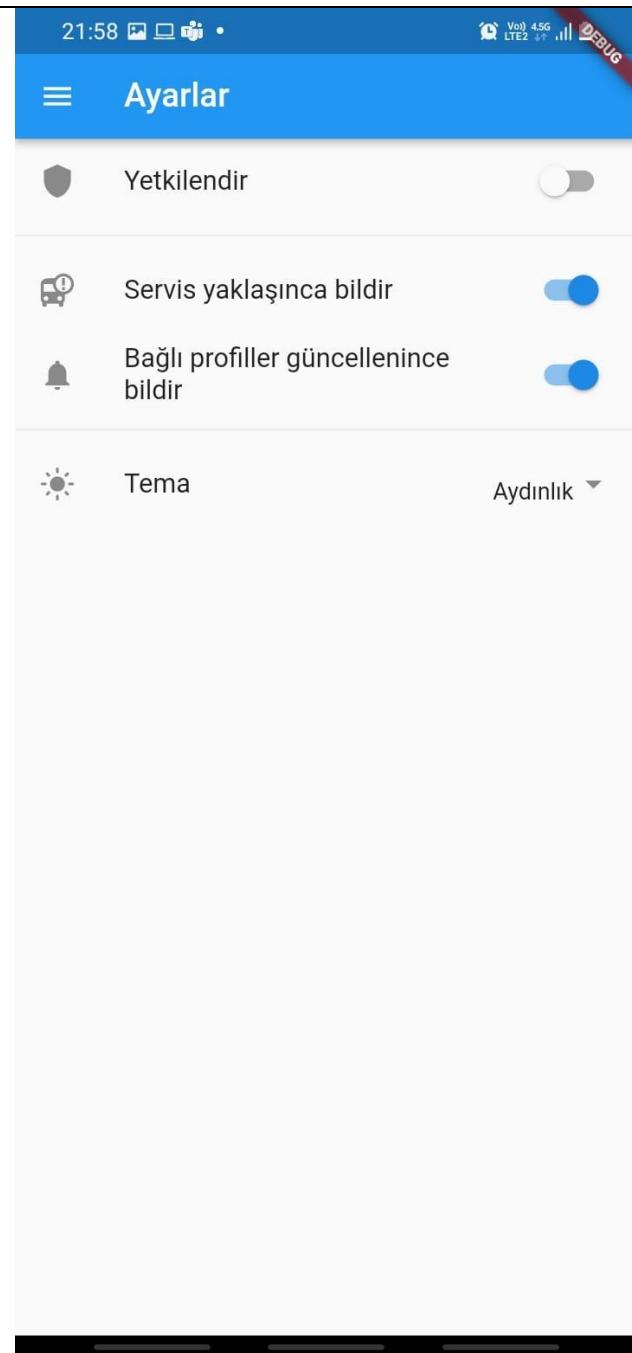
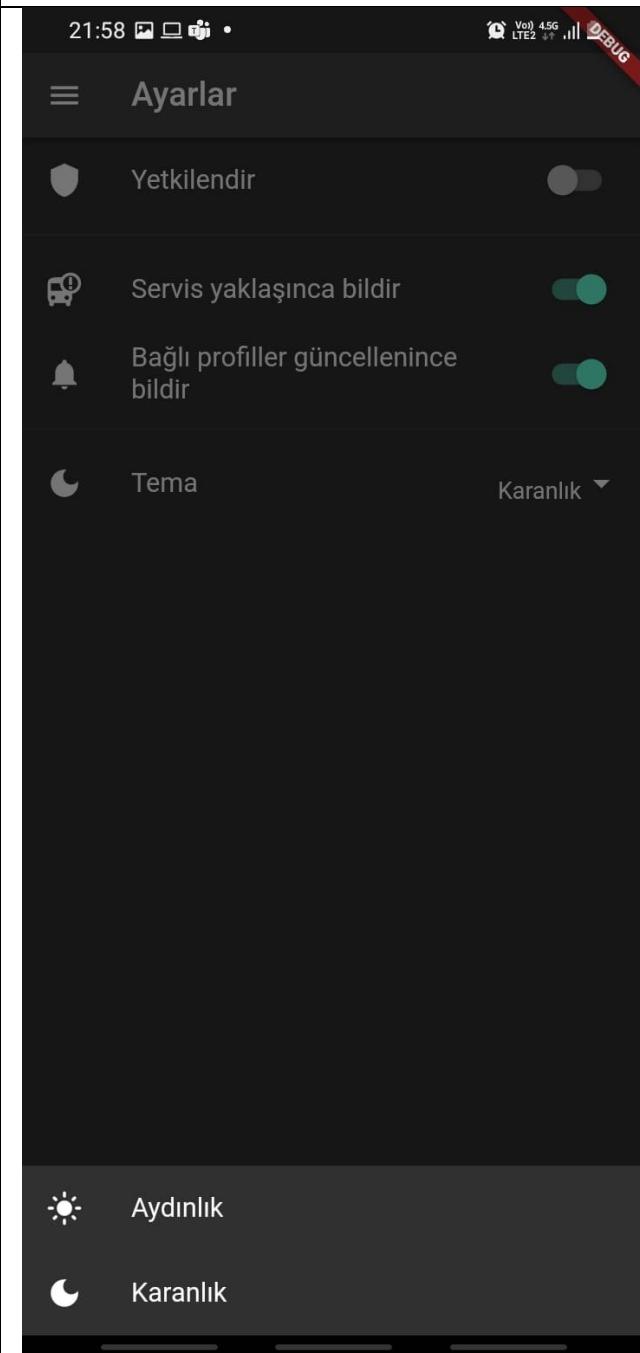
**Result:** "The link has already been used error occurs and validation is not successful for this link

## 6.23. Test Case (Changing Theme and Notifications )

**Description:** Check dark or light theme of application and notifications of application

**Test Steps:**

1. Login Vega App
2. Slide main screen to the right
3. Click setting button
4. Change theme of application and notification



**Result:** The applications theme changes what user wants and notification also arranges according to the user's wishes

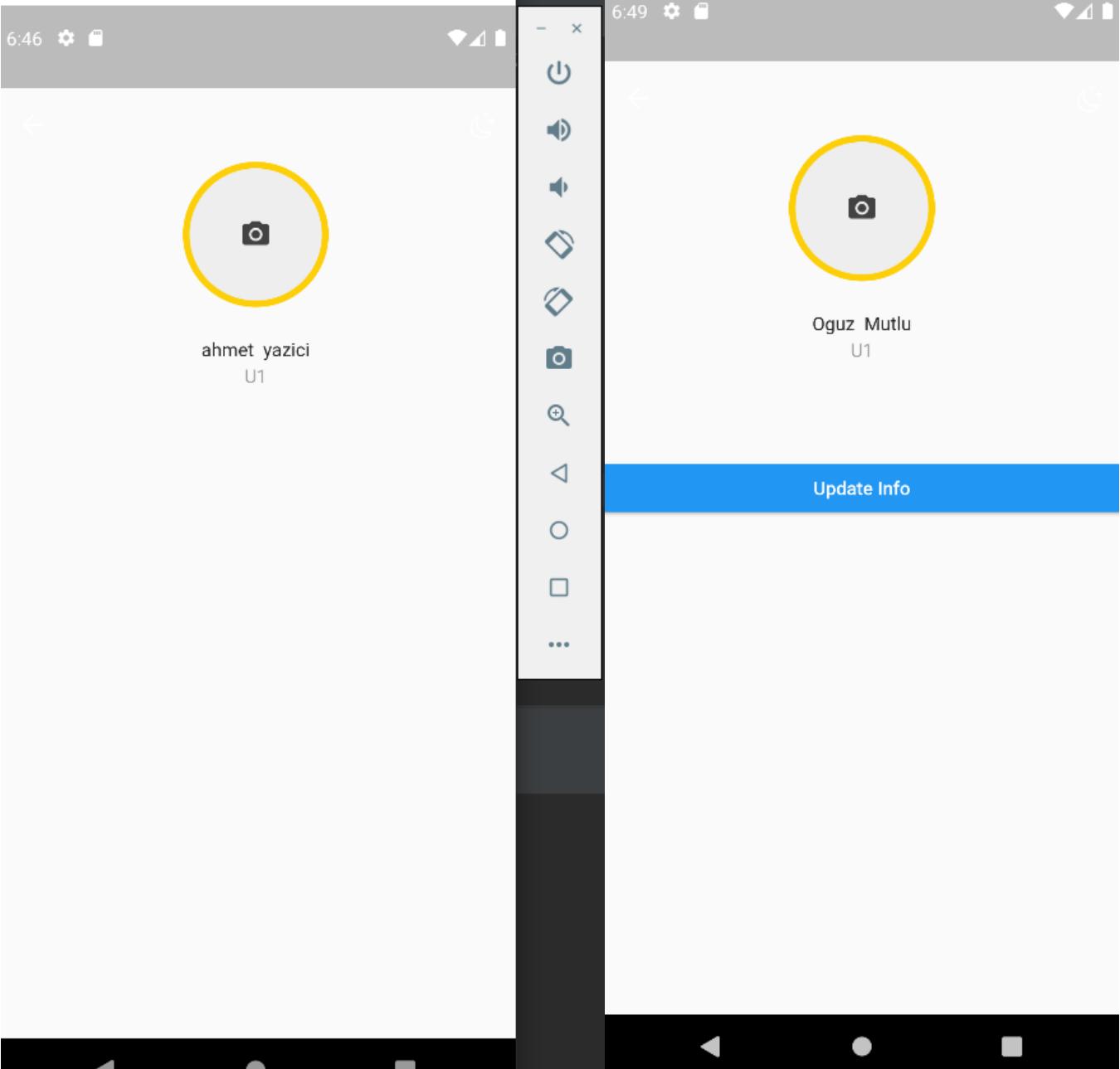
## 6.24. Test Case (Changing Username)

**Description:** Changing User Name

**Test Steps:**

1. Login Vega App
2. Click Profile
3. Click update Info button
4. Write your name

Android Emulator - Pixel\_4\_API\_30:5554



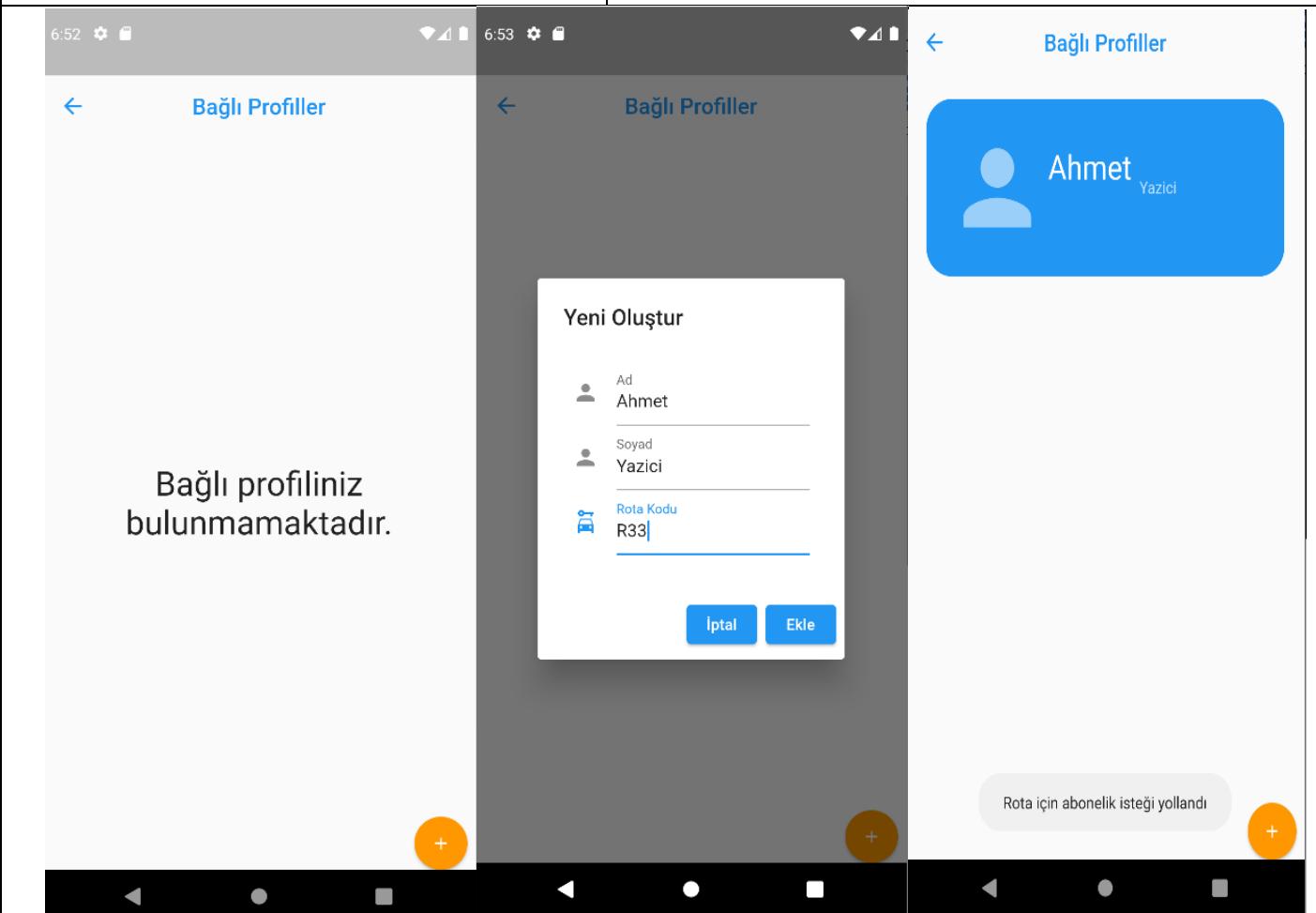
**Result:** The name of user changes what user wants.

## 6.25. Test Case (Create Child Profile and Subscribe Route)

**Description:** Users can add her/his child to shuttle route and linked her/his child to own profile

**Test Steps:**

1. Login Vega App
2. Click “Bagli Profiller”
3. Click Plus icon
4. Enters child Name Surname and correct route code info
5. Route subscription send to Driver



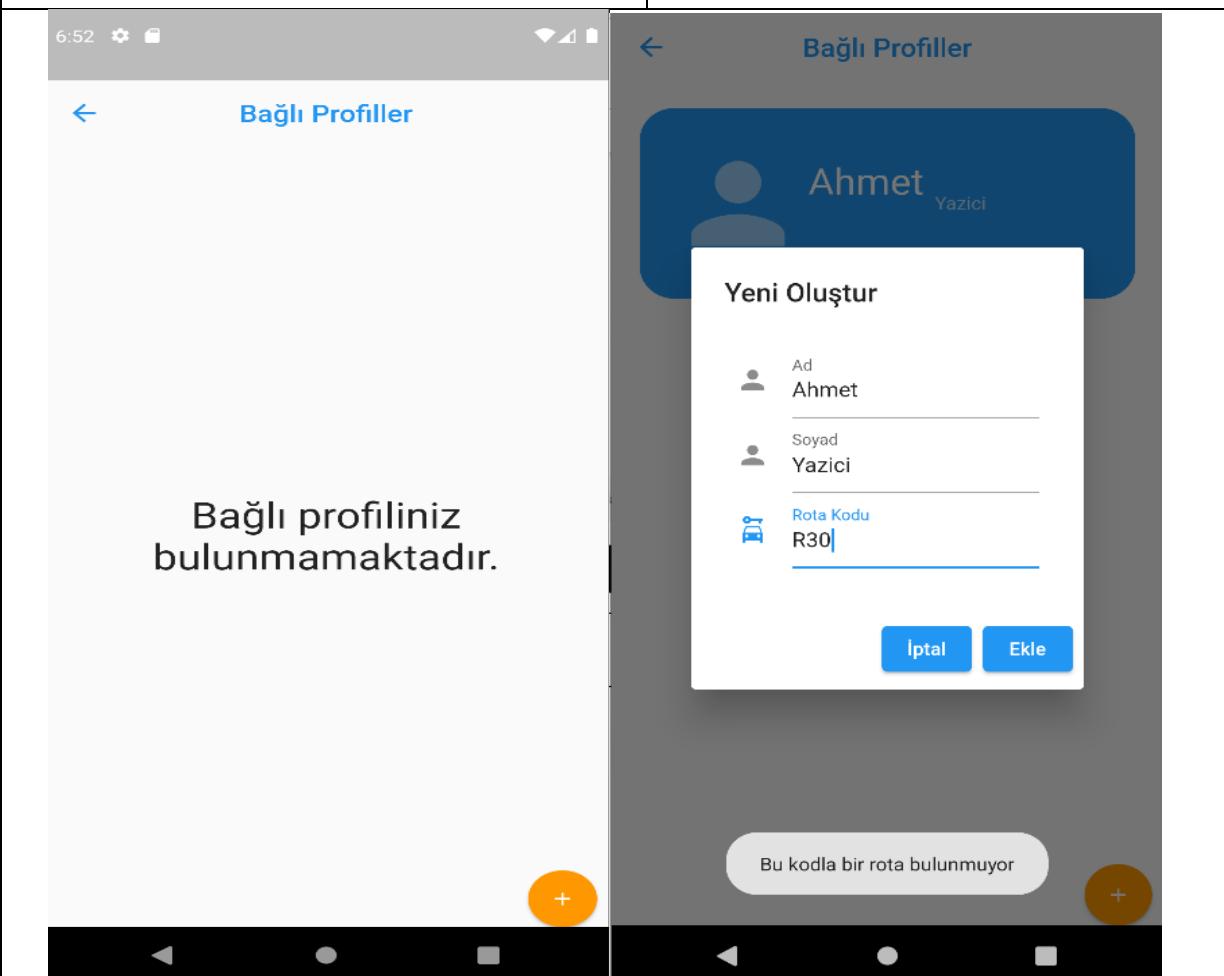
**Result:** Route subscription visible to driver

## 6.26. Test Case (Create Child Profile and Subscribe Route Wrong Route Code)

**Description:** Users can add her/his child to shuttle route and linked her/his child to own profile

### Test Steps:

1. Login Vega App
2. Click “Bagli Profiller”
3. Click Plus icon
4. Enters child Name Surname and wrong route code info
5. Route subscription send to Driver



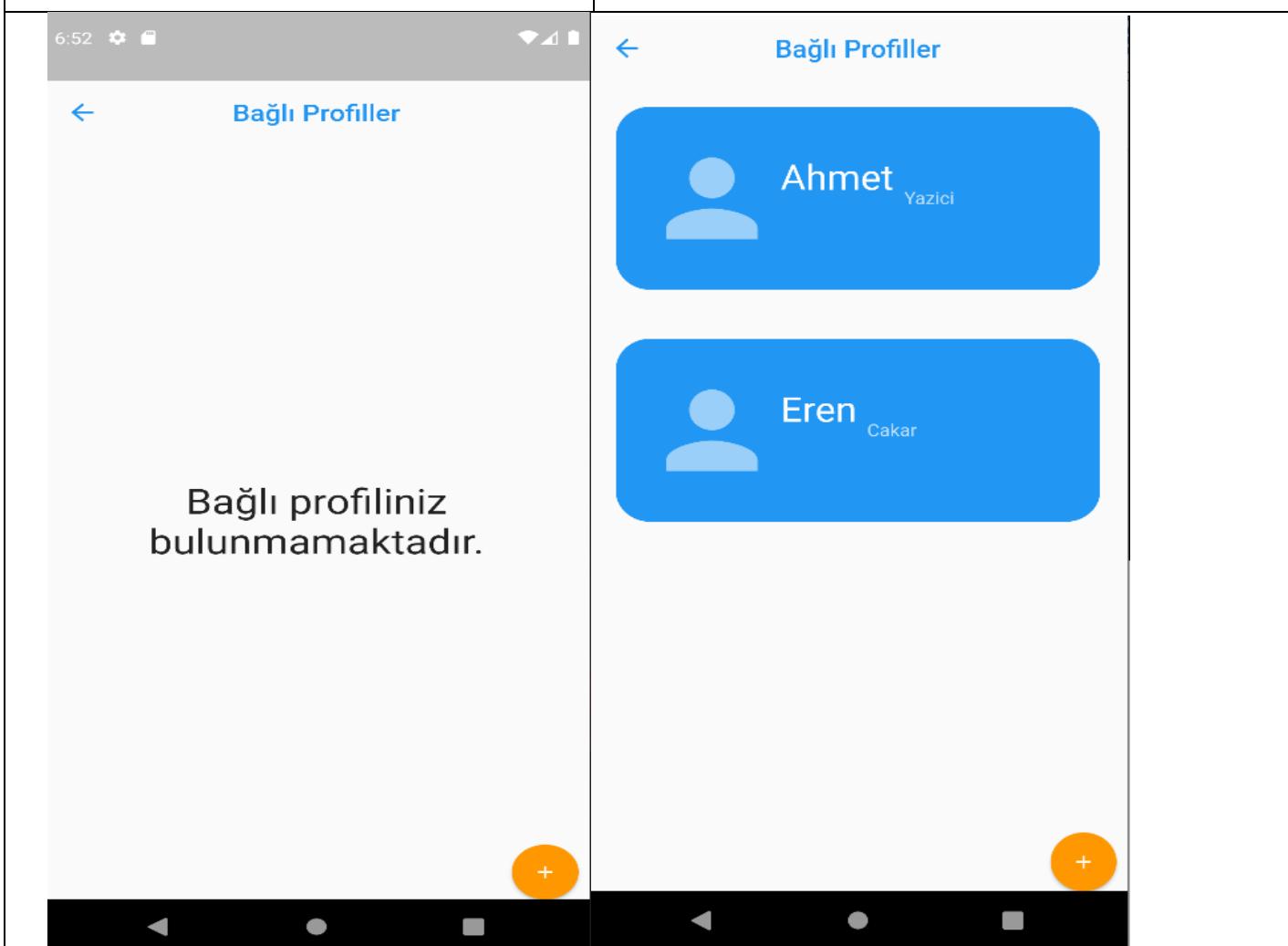
**Result:** Route subscription is not acceptable because of wrong route code

## 6.27. Test Case (Create Child Profile and Subscribe Route more than one child)

**Description:** Users can add her/his second or third child to shuttle route and linked her/his child to own profile

### Test Steps:

1. Login Vega App
2. Click “Bagli Profiller”
3. Click Plus icon
4. Enters child Name Surname and route code info
5. Route subscription send to Driver



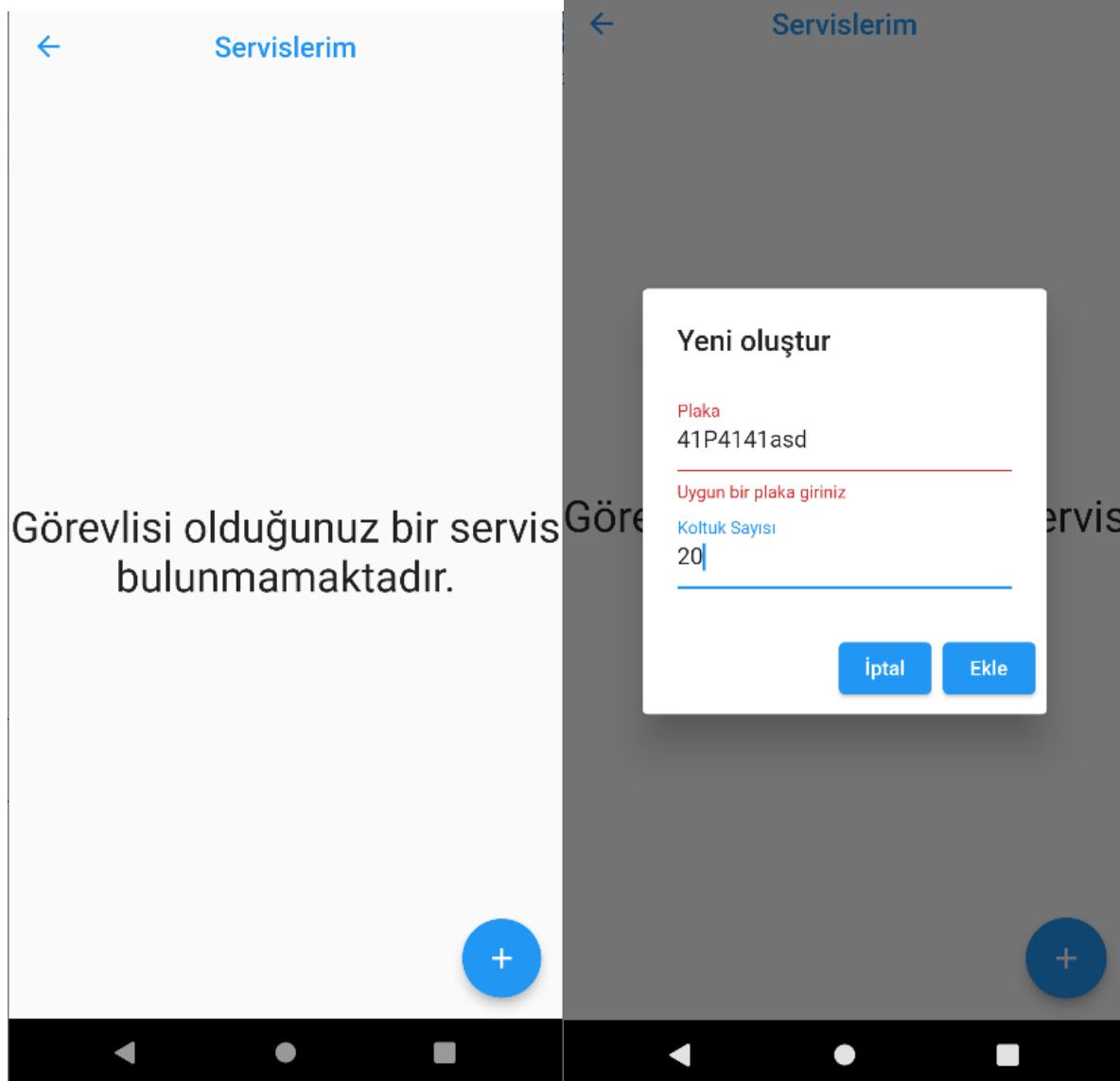
**Result:** Route subscription visible to driver

## 6.28. Test Case (Create Shuttle Enter Wrong Plate)

**Description:** Users can add their shuttle to saved to the system

**Test Steps:**

1. Login Vega App
2. Click “Servislerim”
3. Click ‘+’ icon
4. Enters wrong plate
5. Click “Ekle”



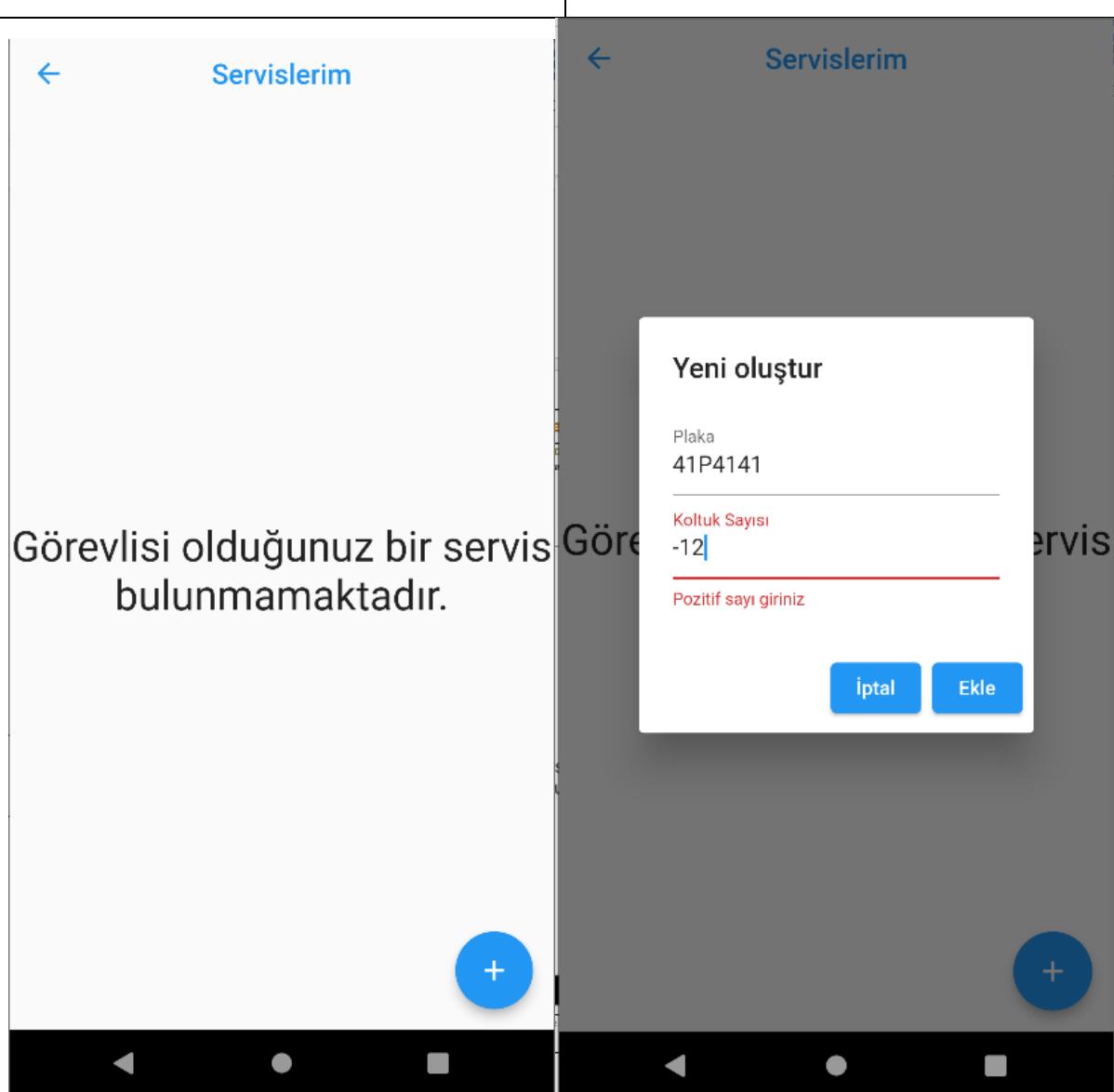
**Result:** Wrongly typed plate is not accepted

## 6.29. Test Case (Create Shuttle Enter Wrong Seat)

**Description:** Users can add their shuttle to save to the system

**Test Steps:**

1. Login Vega App
2. Click “Servislerim”
3. Click ‘+’ icon
4. Enters wrong seat number
5. Click “Ekle”



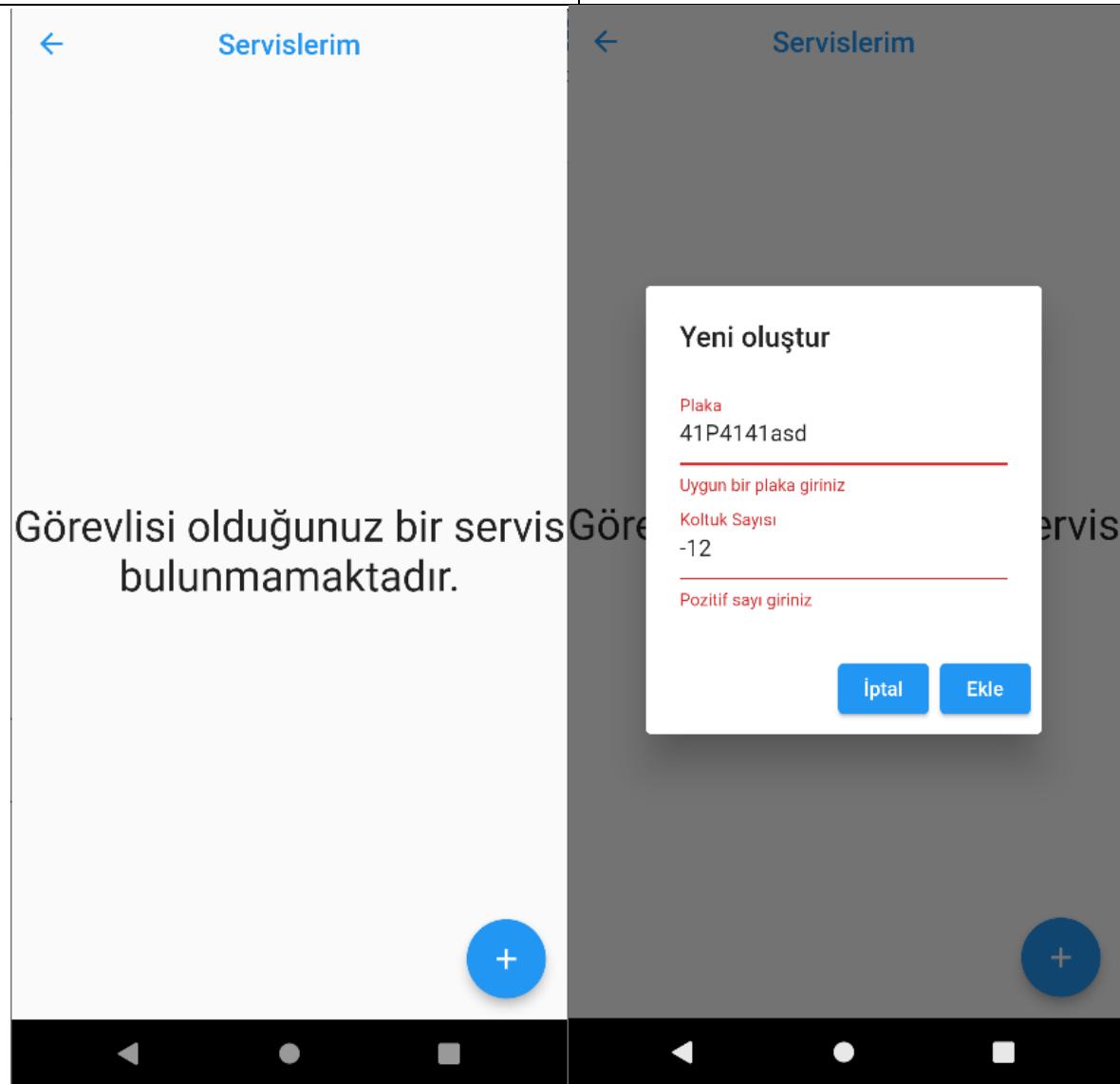
**Result:** Wrongly typed seat is not accepted

## 6.30. Test Case (Create Shuttle Enter Wrong Plate and Seat)

**Description:** Users can add their shuttle to save to the system

**Test Steps:**

1. Login Vega App
2. Click “Servislerim”
3. Click ‘+’ icon
4. Enter wrong seat number
5. Enter wrong plate
6. Click “Ekle”



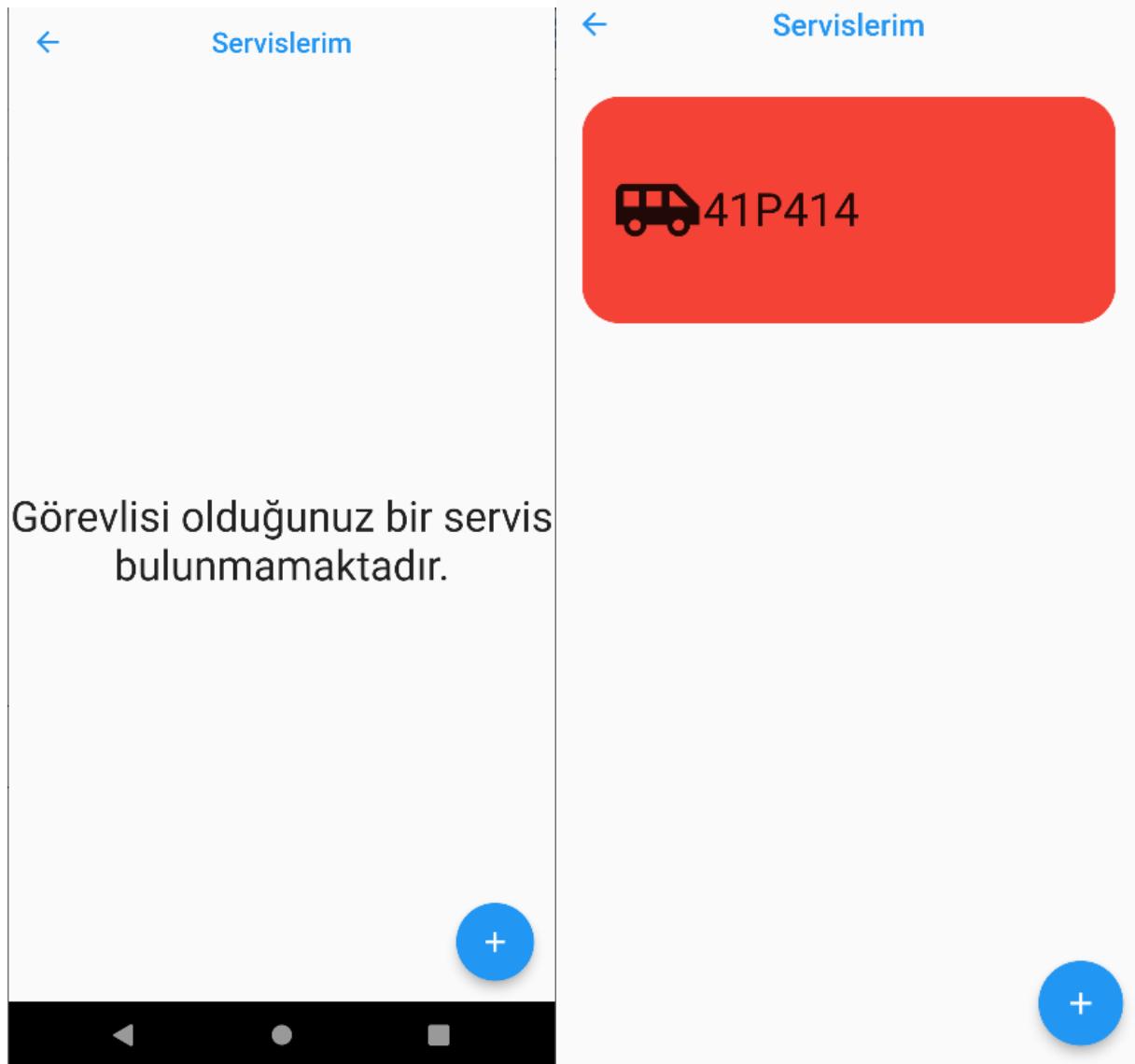
**Result:** Wrongly typed plate and seat is not accepted

## 6.31. Test Case (Create Shuttle Successfully)

**Description:** Users can add their shuttle to save to the system

**Test Steps:**

1. Login Vega App
2. Click “Servislerim”
3. Click ‘+’ icon
4. Enter correct seat number
5. Enter correct plate
6. Click “Ekle”



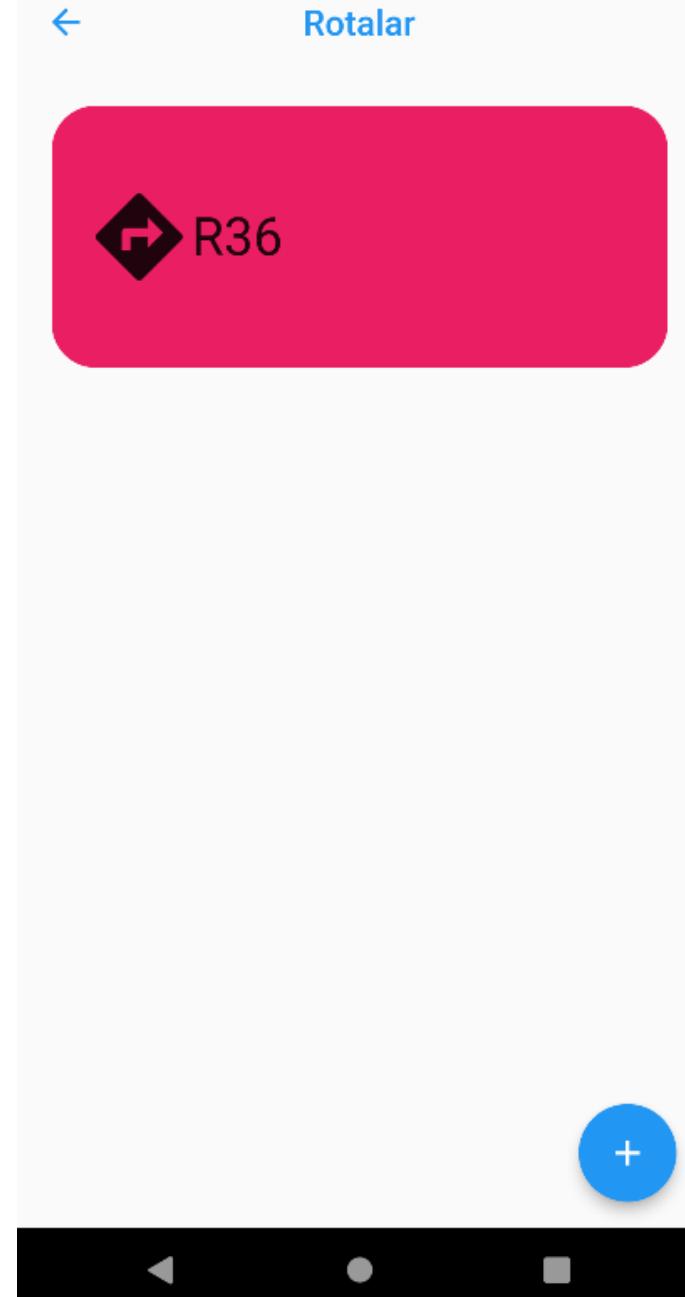
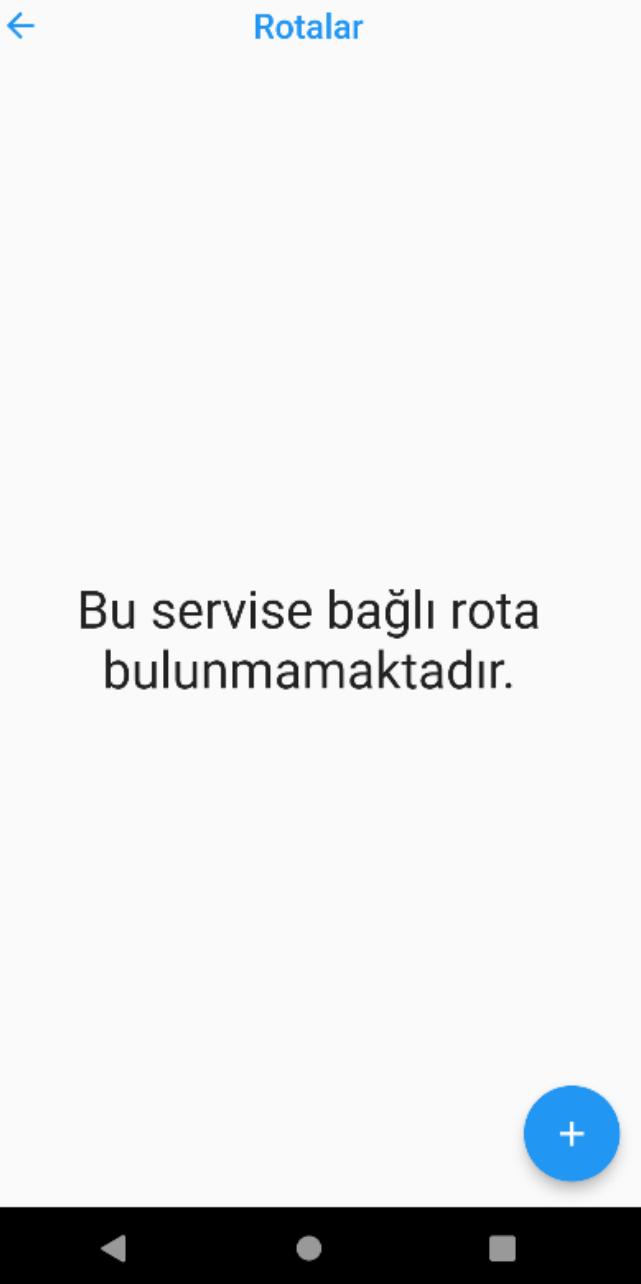
**Result:** Correctly typed plate and seat is visible on database

## 6.32. Test Case (Enter One Route)

**Description:** Drivers can create routes for their shuttles

**Test Steps:**

1. Login Vega App
2. Click “Servislerim”
3. Click service plate
4. Enter wanted route name
6. Click “Ekle”



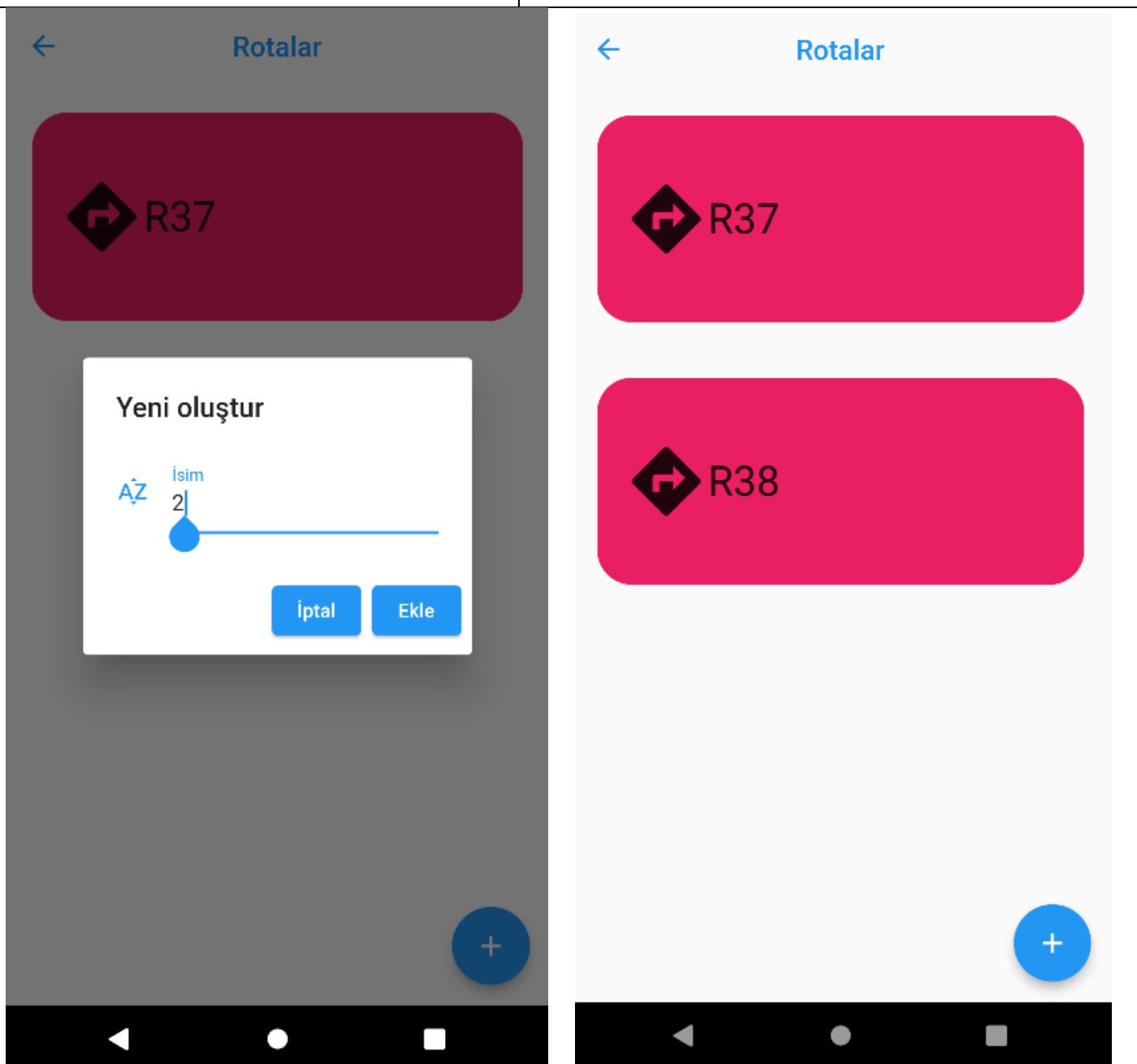
**Result:** Route added to database

## 6.33. Test Case (Enter more than one Route)

**Description:** Drivers can create routes for their shuttles

**Test Steps:**

1. Login Vega App
2. Click “Servislerim”
3. Click service plate
4. Enter wanted route name
6. Click “Ekle”



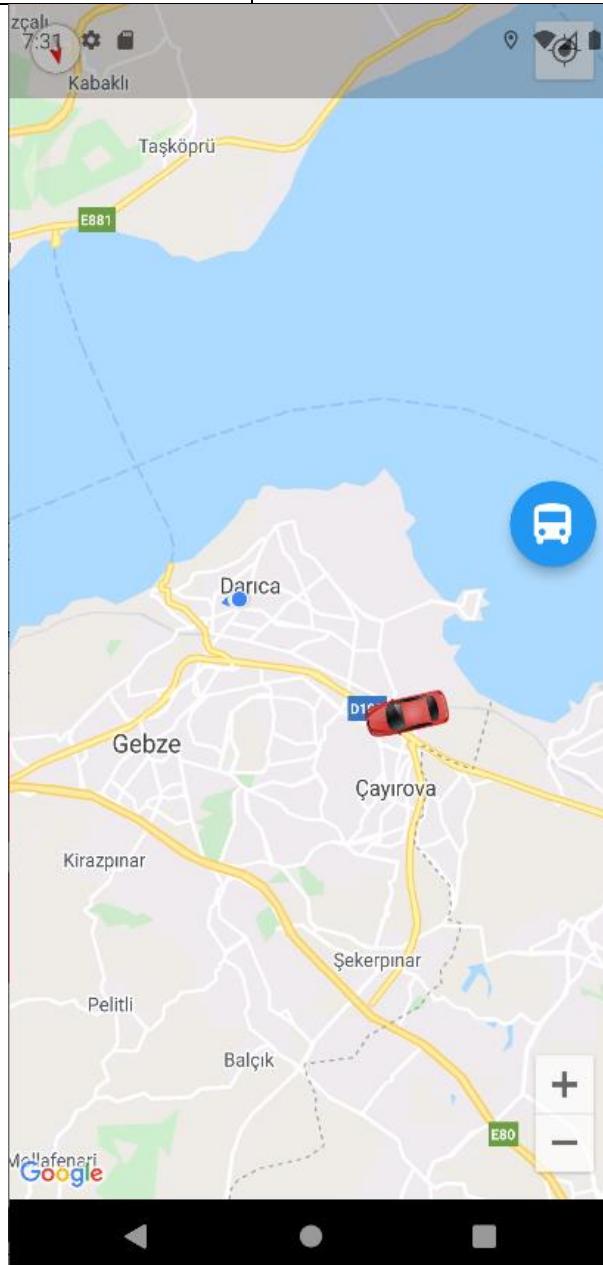
**Result:** If driver work for different company may she/he wants to create new route for new company. Database allows this situation

## 6.34. Test Case (Shuttle's Live Location)

**Description:** Users can see their shuttle's live location

**Test Steps:**

1. Login Vega App
2. Click “Harita”
3. Click “Bus” icon



**Result:** When users clicks the ‘Bus’ icon shuttle marker will we visible by user .