



FPT UNIVERSITY

Capstone Project Document

School bus sharing for near-by students

Group 4 - IS	
Group members	Nguyễn Việt Hùng – SE62279 (Leader) Thái Hiếu Trung – SE62768 Tống Văn Giang– SE62256 Ngô Thế Vinh – SE61840
Supervisor	Nguyễn Anh Khoa
Ext. Supervisor	N/A
Project Code	SCB

Table of contents

A. <i>Introduction</i>	5
1. Project Information	5
2. Introduction	5
3. Current Situation	5
4. Problem Definition	5
5. Proposed Solution	6
5.1 Feature functions.....	6
5.2 Advantages and disadvantages	6
6. Functional Requirements	7
6.1 Administrator.....	7
6.2 System	7
6.3 Driver (Service Provider).....	7
6.4 Customer (Service Consumer)	8
7. Roles & Responsibility	8
B. <i>Software Project Management Plan</i>	9
1. Problem Definition	9
1.1 Name of this Capstone Project	9
1.2 Problem Abstract.....	9
1.3 Project Overview	9
2. Problem Organization	12
2.1 Software Process Model	12
2.2 Roles and responsibilities	12
2.3 Tools and techniques.....	13
3. Project Management Plan	14
3.1 Product backlog	14
3.2 Sprint backlog	14
3.3 Deliverables	14
3.4 All meeting minutes.....	14
4. Coding convention	15
C. <i>Software Requirement Specification</i>	16

1. User Requirement Specification	16
1.1 Guest Requirement.....	16
1.2 Driver Requirement	16
1.3 Customer Requirement	16
1.4 Administrator Requirement.....	16
1.5 System Requirement	17
2. System Requirement Specification	17
2.1 External Interface Requirement	17
2.2 System Overview Use Case	18
2.3 List of use case	19
3. Software System Attribute	99
3.1 Usability	99
3.2 Reliability	99
3.3 Availability	99
3.4 Security	99
3.5 Maintainability.....	100
3.6 Portability	100
3.7 Performance	100
4. Conceptual Diagram	100
D. Software Design Description	102
 1. Design Overview.....	102
 2. System Architectural Design	103
 3. Component Diagram	104
 4. Detailed Description	106
4.1 Class Diagram.....	106
4.2 Interaction Diagram.....	118
 5. User Interface Design.....	133
 6. Database Design	163
6.1 Entity Relational Diagram	163
 7. Algorithms	168
7.1 Find matching Driver Service and Customer Requirement	168
7.2 Find shortest routes between multiple stops (based on Traveling Salesman)	170

E. System Implementation & Test.....	172
1. Introduction.....	172
1.1 Overview.....	172
1.2 Test Approach.....	172
2. Data Relationship Diagram.....	173
3. Test plan.....	174
3.1 Features to be tested.....	175
3.2 Features not to be tested	176
4. System testing Test case	177
4.1 Driver Test Case	177
4.2 Customer Test Case	181
4.3 Logic Test Case.....	184
F. Software's User Manual.....	186
1. Installation Guide.....	186
1.1 Setting up environment at server side	186
1.2 Deployment at server side.....	186
1.3 Setting up environment at client side.....	189
2. User Guide.....	190
2.1 For Guest	190
2.2 For Customer	191
2.3 For Driver	194

A. Introduction

1. Project Information

- Project name: **School bus sharing for near-by students**
- Project Code: **SCB**
- Product Type: **Mobile Application**
- Start Date: **January 7th, 2019**
- End Date: **May 6th, 2019**

2. Introduction

Nowadays, life of married couples is very busy, especially when they have young children. In the morning, the wife, or the husband has to take their children to school because kids cannot go to school by themselves. It would be really inconvenient when parents get their kids to school, then go to work, because not every parent has enough time in the morning, due to rush hours, or some other personal reasons.

In this document, we introduce a solution for parents: a mobile application called School Bus. School Bus is a powerful tool to help parents not worry about shuttling their kids to school every day anymore. Kids would be picked up to school and taken back home on time and safely. This document also describes our working process in 4 months including our perspectives on the system, component designs and detailed core workflow.

3. Current Situation

The busy life does not allow parents to take care of their kids properly. One issue of these is that fathers and mothers have to think of many ways to shuttle their kids to school daily regardless of difficult situations such as traffic jam, inflexible working time, etc.

Many temporary approaches have been given to solve the problem, parents can take the kids to school very early (which wastes time of whole families), or rent taxi as well as Grab to take their responsibilities (which could pose a lot of risks from dealing with a strange driver every day without any permanent contract).

4. Problem Definition

Following are the disadvantages of the current situation:

- Parents are very busy that they could not take their kids to school in the morning and pick up their kids in the afternoon.
- Parents' time for work and kids' time for school are not compatible.
- The ways to kids' schools and way to parents' workplace are opposite.
- Hard to find a reliable rental motorbikers in neighborhood or Grab, GoViet, etc.

Above problems may have negative impact on both parents and kids' working days. Kids could be late for school, parents could be late for work because of traffic jam or some unexpected situations caused by unreliable rental motorbikers.

5. Proposed Solution

Our proposed solution is to build a system named "School Bus" to resolve the current problems. With this application, parents wouldn't be worried anymore about taking their kids in the morning or taking them home after school. Another parent with a car would register to be a driver. This driver would take nearby kids to their school and pick them up after class following an arranged schedule based on the service that their parents register before.

School Bus system includes one administration web-based application along with two mobile applications, one for service providers known as drivers, and one for service consumers known as customers, with the functions as follow.

5.1 Feature functions

- Mobile application for drivers
 - Service registration: a parent could register to be a driver. In order to do that, he or she must provide basic information and meet service requirements, including children's information and schedule.
 - Considering the contract: the driver would be notified when he/she is chosen by a customer. Then, they could choose to accept or reject the contract agreement.
 - Journey's planning: everyday, the driver would receive the detailed journey and notifications for picking up customer's kids, taking them to school then taking them back after class.
- Mobile application for customers
 - Requirement registration: parents could register a requirement to use service, he or she must provide basic information and service requirements, including children's information and schedule.
 - Choose the appropriate driver: customers are able to choose a suitable driver from a provided list of available options. Then, there will be notification when their contract agreements are approved by the driver.
 - Journey's notification: everyday, customers would receive their driver's detailed journey and notifications about the time when the driver comes in the morning, comes back again in the afternoon, and when the driver arrives at school.

5.2 Advantages and disadvantages

The advantages and disadvantages of proposed solution:

- Advantages
 - Increased reliability of drivers because they're parents too and their profiles are explicit.

- Drivers have an explicit visual route shown on the map for every day journey.
- The route is calculated to be shortest and most appropriate for all pick-up locations and the drivers.
- Parents could track the position of their drivers.
- Parents could find the most suitable drivers for their children.
- Parents could always receive notifications about their drivers and their kids' location.
- Disadvantages
 - System only supports limited radius between pick-up locations to ensure the quality of service.
 - System only supports contracts in which kids are from same school.
 - System only allows drivers to register one school for one service.

6. Functional Requirements

Functional requirements of the system are listed as following:

6.1 Administrator

- Administrator could manage all user's accounts.
- Administrator could verify driver's registrations.
- Administrator could configure all global specifications for the whole system.
- Administrator could review statistics and make throughout reports of whole system's activities by date.

6.2 System

- System could push notifications to user.
- System could find matching customers and drivers.
- System could calculate money.
- System could process and find the most appropriate route for driver in each journey.

6.3 Driver (Service Provider)

- Driver could register account.
- Driver could login.
- Driver could manage profile.
- Driver could register services.
- Driver could manage his/her services.
- Driver could view notifications.
- Driver could accept or reject contract agreement requests.
- Driver could view his/her contract details.
- Driver could view daily journey's detail and its route explicitly and visually on a map.
- Driver could have bill information after each trip.
- Driver could mark if the kids are picked up or not.

- Driver could call the customer.
- Driver could cancel some trips.
- Driver could cancel contracts.

6.4 Customer (Service Consumer)

- Customer could register account.
- Customer could login.
- Customer could manage profile.
- Customer could register requirements.
- Customer could manage his/her requirements.
- Customer could view notifications.
- Customer could view appropriate drivers then decide to choose or not.
- Customer could view his/her contract details.
- Customer could view the journey's route of their drivers, explicitly and visually on a map.
- Customer could have bill information after each trip.
- Customer could call the driver.
- Customer could cancel some trips.
- Customer could cancel contracts.
- Customer could extend their contracts.

7. Roles & Responsibility

No	Full Name	Role	Position	Contact
1	Nguyễn Anh Khoa	Project Manager	Supervisor	khoa.nguyen@saigontechnology.com
2	Nguyễn Việt Hùng	Developer	Leader	nvhungkt1997@gmail.com
3	Thái Hiếu Trung	Developer	Member	trunghthaihieu93@gmail.com
4	Tống Văn Giang	Developer	Member	giangtvse62256@fpt.edu.vn
5	Ngô Thế Vinh	Developer	Member	Vinhntse61840@fpt.edu.vn

B. Software Project Management Plan

1. Problem Definition

1.1 Name of this Capstone Project

- Official name: School bus sharing for near-by students
- Vietnamese name: Ứng dụng đưa đón học sinh ở gần nhà nhau
- Abbreviation: **Schoolbus**

1.2 Problem Abstract

The system is about how to provide the service to customers to make sure their kids are picked up to school on time daily. By making contracts between the drivers and customers, this can be handled. In details, every day the system will announce and show the direction for driver to pick up the kids and take them to schools. The contracts will enhance the trust between driver side and customer side.

We provide the software to both drivers and customers, in which they can register services and match those suitable services to create contract. We take advantages of mobile application, using geolocation via Map API to provide visual user interface, and also to allow customers to track the driver status while driving their kids to school.

1.3 Project Overview

1.3.1 Current Situation

Below are the problems encountered in the project:

Safety: The children can be in danger while being picked up to school by a strange person.

Availability: The information is exchanged and processed on server, so the server must be available for all of the time.

Framework: Not all the members in team are fluent in the frameworks which applies to both backend and mobile application. An additional time is required for the development team to get familiar with them.

Time: The time for the whole project is limited meanwhile the team has to handle many jobs such as documenting, implementing, testing and deploying system.

1.3.2 The proposed system

According to previous researches, we will apply Google Map API for the great supports and stability. With these APIs, we are capable of processing the coordinates and using alternative algorithms to match suitable drivers and customers as well as suggesting the optimized routes for drivers.

Features in system are divided into smaller tasks and assigned to members every sprint in the project development process. Therefore, if any possible changes in team members arise, the project can keep going on.

Our system includes 3 main subsystems: Web management for administrators, mobile application for drivers and mobile application for customers.

1.3.2.1 Web management

Web management is designed for administrators to:

- Manage users in system
- Configure some attributes for the system
- Have statistics and reports in the system

1.3.2.2 Mobile application (customers)

Customer can use mobile application to book a service for the children, choose the suitable driver and follow their kids during the drive through the devices. Customers can also manage their contracts and adjust information for themselves and their kids, which are critical information for the services.

1.3.2.3 Mobile application (drivers)

Driver can use mobile application to register services for customers. The application will provide the interface for drivers to complete the trip, help them to contact easily with customers. Drivers can manage their contracts and information as well.

1.3.3 Boundaries of the system

The system can:

- Allow driver to register services.
- Allow customer to register requirements.
- Allow customer and driver to match their needs and create a contract.
- Allow driver to have an intuitive view while drive the kids to school.
- Allow both customer and driver to cancel some trips in contract or cancel a contract.

The system cannot:

- Allow driver to have multiple services to multiple schools at the same time.
- Allow customer to make adjustments on some specific days in contract.

1.3.4 Future plans

The system will support multiple ways for payment methods (e-wallet, credit/debit cards).

The system will support service for multiple stops and multiple destinations.

Customers can use the system on website.

1.3.5 Development environment

1.3.5.1 Hardware requirements

For server

Hardware	Minimum Requirement	Recommended
Internet Connection	Cable, Wi-Fi (8 Mbps)	Cable, Wi-Fi (50 Mbps or more)
Operation System	XP, Vista, 7, 10, Window Server 2008	10, Window server 2008
Computer Processor	Intel® Xeon ® 1.4GHz	Intel® Xeon ® Quad Core (12M Cache, 2.50 GHz)
Computer memory	4GB RAM	32 GB RAM or more
Storage space	1GB	5GB or more

Table 1 - Hardware Requirements for Server

For Mobile

PC	Minimum Requirements	Recommended
Internet Connection	Wi-Fi (4 Mbps) / 2G / 3G	Wi-Fi (8 Mbps) / 4G / 5G
Operating System	Android 7 iOS 9	Android 9 iOS 12
Storage Space	100MB	200MB
Memory	1GB RAM	3GB RAM or more

Table 2 - Hardware requirement for Mobile

1.3.5.2 Software requirements

Software	Name / Version	Description
Operation System	Windows 10 build 1809 MacOS 10.14.3	Operating system and platform for development
Environment	C# 7, .NET framework 4.7.2	Specification for developing web services
	Node v10, npm v6	Specification for developing mobile application and web app
Modeling tool	StarUML, draw.io	Used to design diagram
IDE	Visual studio 2017	Programming tools

	VS Code 1.32	
DBMS	Microsoft SQL Server	Used to create & manage the database for system
Source control	Github.com Git-scm Sourcetree	Used for source control
Web browser	Chrome 72 or above	Debugging browser
CI/CD	Jenkins	Deploying and testing rapidly

Table 3 - Software Requirement

2. Problem Organization

2.1 Software Process Model

This project is developed using Scrum framework – a part of the Agile software development for the following reasons:

- The software requirements are not clear enough for us in the beginning, therefore, we have to adapt to new knowledges, new technical challenges and changes in requirements. Scrum is suitable for us to handle these challenges.
- There is no hierarchy in team, which allows everyone to feel free to share and cooperate better.
- The project is complicated and we have many plans in the future, so working in iterative sprints enables us to process everything step by step.

SCRUM FRAMEWORK

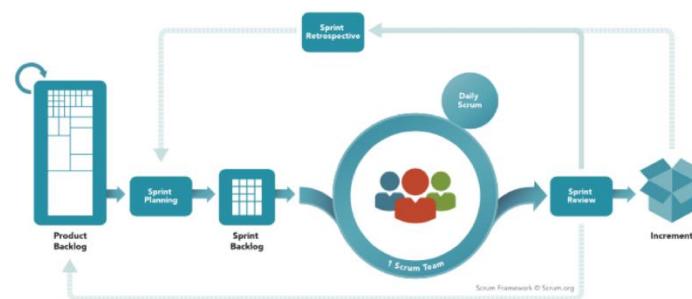


Figure 1 - Scrum Framework

2.2 Roles and responsibilities

No	Full name	Role in Group	Responsibilities
1	Nguyễn Anh Khoa	Product Owner	<ul style="list-style-type: none"> - Specify user requirement

			<ul style="list-style-type: none"> - Control the development process - Give out technique and business analysis support
2	Nguyễn Việt Hùng	Scrum Master	<ul style="list-style-type: none"> - Managing process - Designing database - Clarifying requirements - Prepare documents - GUI Design - Create test plan - Coding - Testing - Arrange Meeting - Risk Management
3	Thái Hiếu Trung	Scrum team member	<ul style="list-style-type: none"> - Designing database - Clarifying requirements - Prepare documents - GUI Design - Create test plan - Coding - Testing
4	Ngô Thế Vinh	Scrum team member	<ul style="list-style-type: none"> - Designing database - Clarifying requirements - Prepare documents - GUI Design - Create test plan - Coding - Testing
5	Tống Văn Giang	Scrum team member	<ul style="list-style-type: none"> - Designing database - Clarifying requirements - Prepare documents - GUI Design - Create test plan - Coding - Testing

Table 4 - Roles and responsibilities

2.3 Tools and techniques

Tool/Technique	Name
----------------	------

Front-end	HTML5, CSS3, JavaScript ES6, ReactJs v16.8, Eslint
Mobile	Javascript ES6, React-native v0.57, Expo SDK v32.0.0, Eslint
Back-end	.NET framework 4.7.2
IDE	Visual Studio 2017, VS Code 1.32
DBMS	Microsoft SQL Server
Source Control	Github.com Git-scm Sourcetree
Modelling tool	StarUML, draw.io

Table 5 - Tools and techniques

3. Project Management Plan

3.1 Product backlog

Product backlog could be found here.

3.2 Sprint backlog

Sprint backlog could be found here.

3.3 Deliverables

No	Deliverable	Note
1	Introduction, Working environment, Source code structure	Sprint 1
2	Requirements, Use case overview, ERD and Database design	Sprint 2
...		

Table 6 - Deliverables

3.4 All meeting minutes

All sprint meeting minutes could be found here.

4. Coding convention

Summary:

- **Naming convention:**
 - Variable names must be meaningful, they must reference to their functions, attributes.
 - For variables, using camel case: Eg: driverService, orderDetail, etc.
 - For functions, using pascal case in C#, using camel case in Javascript.
- **Layout convention:**
 - Each line contains only one statement or declaration at maximum.
 - Indentation: Using tab (4 spaces) in C#, using 2 spaces in Javascript.
 - Having one blank line between each function declaration.
 - Code splitting: each file must contain only one class at maximum, each class references to only one meaning.
- **Commenting convention:**
 - Comment must be in a separate line, before the statement which need to be described.
 - Avoid using TO_DO, TO_FIX in comment, avoid commenting source code.
 - For commenting inside a function, using single-line comment (begins with //)
 - For commenting about a function, using multi-lines comment (wraps with /** *), parameters and return value should be described.
- **Language Guidelines:**
 - Using C# coding convention from here:
<https://docs.microsoft.com/en-us/dotnet/csharp/programming-guide/inside-a-program/coding-conventions>
 - Using Javascript coding convention from here:
<https://github.com/airbnb/javascript>

C. Software Requirement Specification

1. User Requirement Specification

1.1 Guest Requirement

Guest is a person who doesn't have access to the system. Guest can only open the application in login page. To continue, guest must login. These are all functions guest can use:

- Login.
- Register account.

1.2 Driver Requirement

Driver is an authorized user who has permission to login into the driver application. As a driver role, this user can use these functions:

- Update his/her car information.
- Create and manage services.
- Get customers' contract agreement requests and accept or reject them.
- Manage his/her contracts with customers.
- Start trip in his/her service daily and follow up customer as well as children inside the trip.
- Get feedbacks from customers.
- Get trip history and bill from them.
- Get notifications.

1.3 Customer Requirement

Customer is an authorized user who has permission to login into the customer application. As a customer role, this user can use these functions:

- Manage his/her children.
- Create and manage requirements.
- Choose match driver services and make contract agreement.
- Manage his/her contracts with drivers.
- Track driver during the trip.
- Create feedback to drivers.
- Get trip history and bill from them.
- Get notifications.

1.4 Administrator Requirement

Administrator is an authorized user who has permission to login into the web administration. As an administrator role, this user can use these functions:

- Manage all customer and driver accounts.
- Verify driver's registrations.
- Configure all global specifications for the whole system.

- Review statistics and make throughout reports of whole system's activities by date.

1.5 System Requirement

System has responsibility to:

- Send notifications to users.
- Calculate the amount of money for customer requirements.
- Find matching driver services for customers.
- Find the suitable route for driver.

2. System Requirement Specification

2.1 External Interface Requirement

2.1.1 User Interface

The customer's and driver's mobile application user interface will be in multi-language (English and Vietnamese) which is ensured to:

- Have intuitive navigation functions.
- Have clear Button, Text Input, Touchable View for users to avoid misunderstanding.
- Have suitable icon, image and color to make components become simple, reminiscent.
- Have great arrangement for every component to keep information clear, easy to understand and easy to interact.
- Follow interface design iterative process including: designing, sketching, prototyping and user assessment.

2.1.2 Hardware Interface

Smartphone running on Android OS or iOS.

Computer.

2.1.3 Software Interface

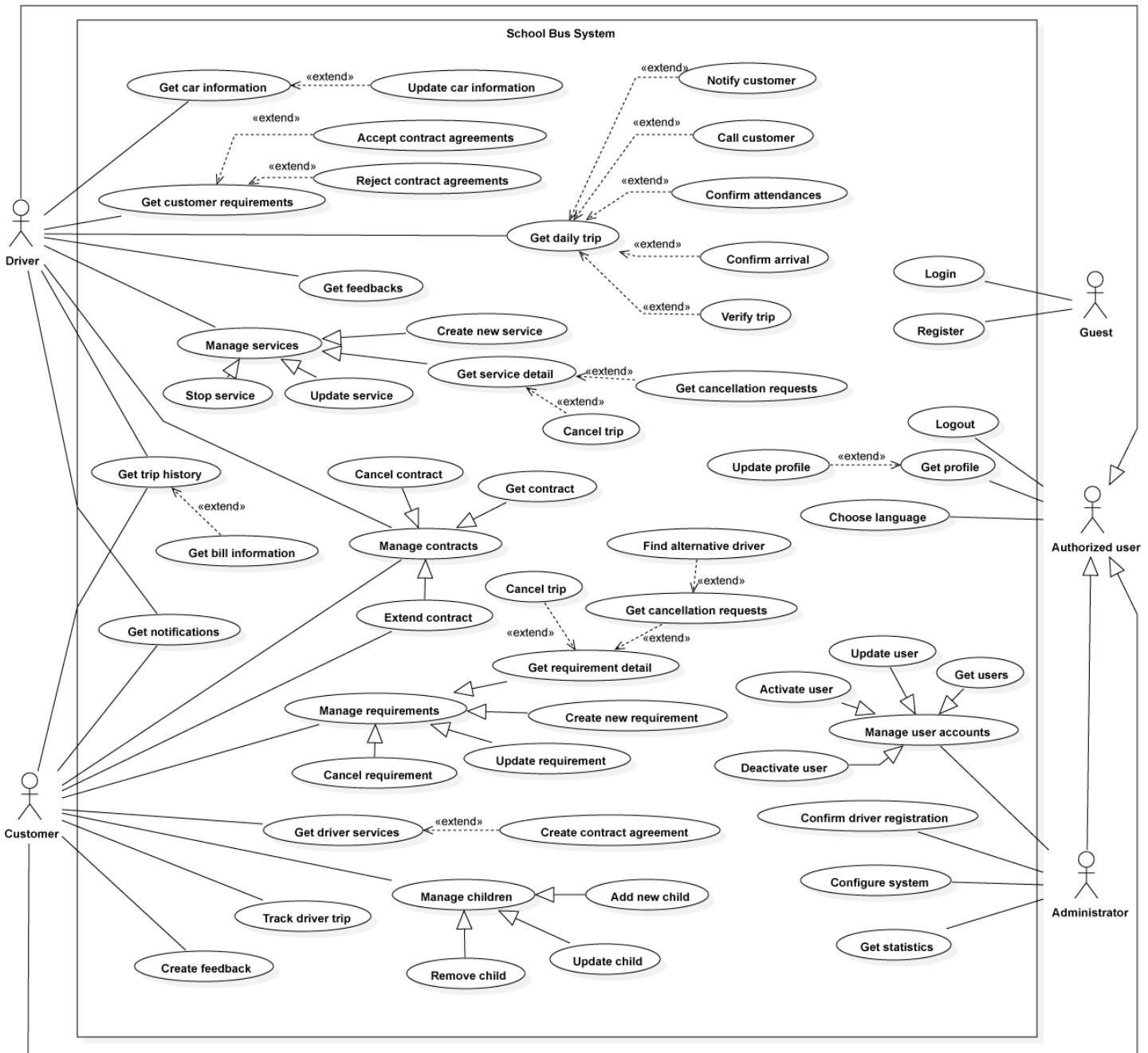
Web administration: support browsers Firefox (from v47), Chrome (from v48), Internet Explorer (from v10), Microsoft Edge (from v14), Safari (from v8).

Mobile application: work with Android OS (v7.0 – Nougat or above), iOS (v11)

2.1.4 Software Interface

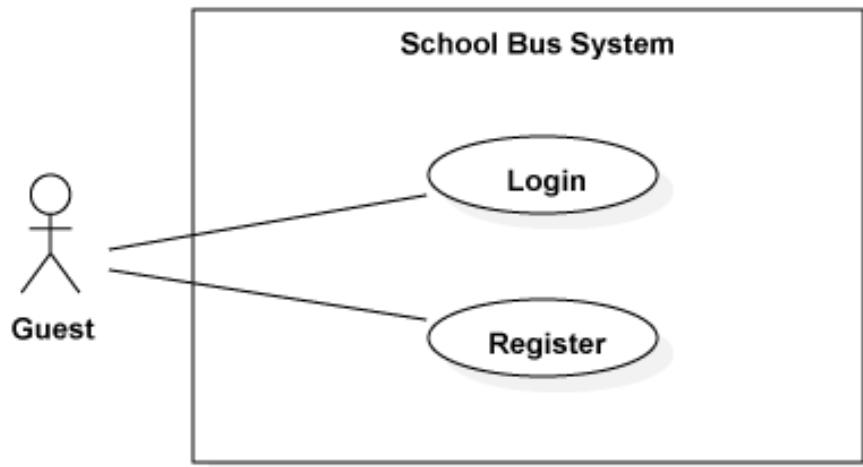
Use HTTP protocol 1.1 for communication between: web application and server, mobile application and server.

2.2 System Overview Use Case

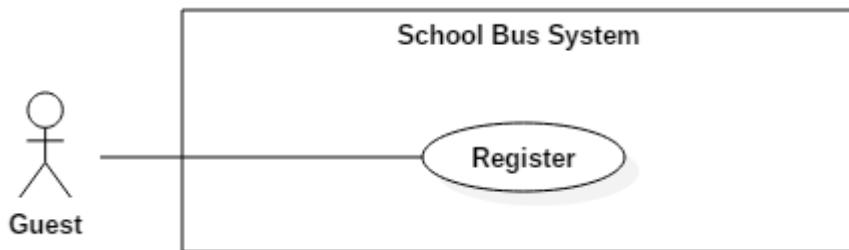


2.3 List of use case

2.3.1 <Guest> Overview Use Case

*Guest Overview Use Case*

2.3.1.1 <Guest> Register

*<Guest> Register*

USE CASE - SCB_UC_1			
Use Case No.	SCB_UC_1	Use Case Version	1.0
Use Case Name	Register		
Author	TrungTH		
Date	13/2/2019	Priority	Normal
Actor:	<ul style="list-style-type: none"> - Guest 		
Summary:	<ul style="list-style-type: none"> - Allow guest to register new account. 		
Goal:	<ul style="list-style-type: none"> - Guest can register new account. 		
Triggers:	<ul style="list-style-type: none"> - Guest sends register command. 		
Preconditions:	N/A		
Post conditions:	<ul style="list-style-type: none"> - Success: Guest registers new account successfully. - Fail: System shows error messages. 		

Main Success Scenario:

Step	Actor Action	System Response
1	Guest goes to register view.	System requires identity information from Guest: - Phone number: text input. - Password: text input - Confirm: text input
2	Guest inputs information.	
3	Guest sends command to register new account.	Guest register new account [Alternative 1] [Alternative 2] [Exception 1]

Alternative Scenario:

No	Actor Action	System Response
1	Guest enter wrong format.	System shows appropriate validating message. Ex: "Phone number must be 10 numbers", "Password not match"
2	Duplicate phone number	System shows error message "This phone number has been used. Please choose another phone number."

Exceptions:

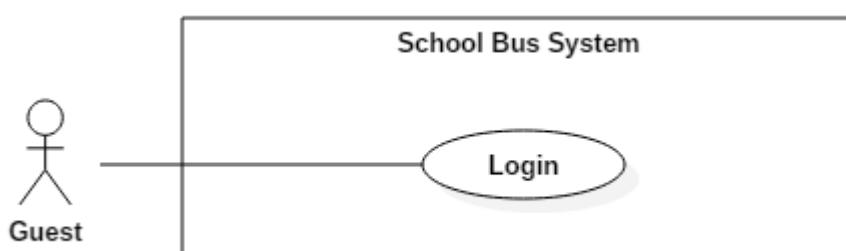
No	Actor Action	System Response
1		System shows message the "Please check your connection!" when the internet is lost.

Relationships: N/A**Business Rules:**

- If success, user will gain an access token to have authorization in other actions.
- After registering, user have to update their profile.
- After registering, the device token is saved for pushing notification.

<Guest> Register Use Case Specification

2.3.1.2 <Guest> Login

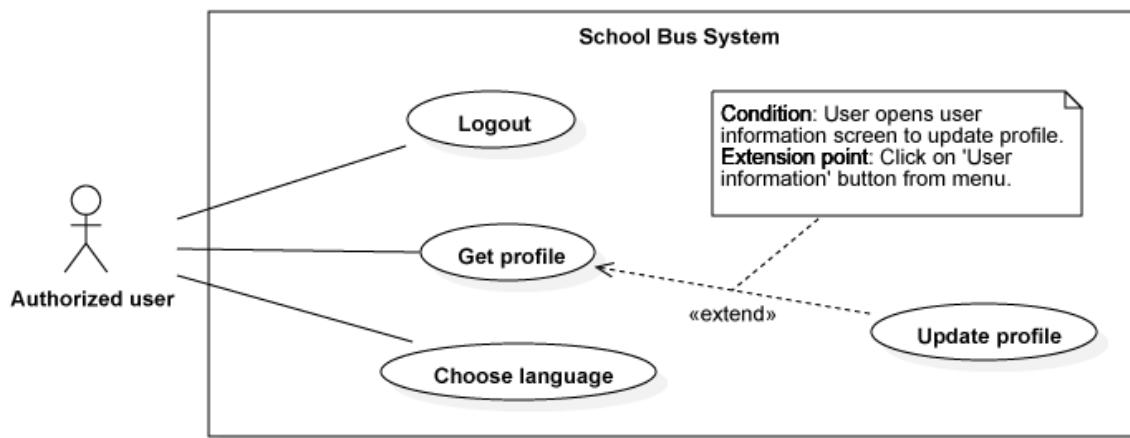


<Guest> Login

USE CASE - SCB_UC_2					
Use Case No.	SCB_UC_2	Use Case Version	1.0		
Use Case Name	Login				
Author	TrungTH				
Date	13/2/2019	Priority	Normal		
Actor:	<ul style="list-style-type: none"> - Guest 				
Summary:	<ul style="list-style-type: none"> - Allow guest to log in the system. 				
Goal:	<ul style="list-style-type: none"> - Guest can log in the system with specific role. 				
Triggers:	<ul style="list-style-type: none"> - Guest sends login command. 				
Preconditions: N/A					
Post conditions:	<ul style="list-style-type: none"> - Success: Guest is exactly authorized with specific role in the system. - Fail: System shows error messages. 				
Main Success Scenario:					
Step	Actor Action	System Response			
1	Guest goes to login view.	System requires identity information from Guest: <ul style="list-style-type: none"> - Phone number: text input. - Password: text input. 			
2	Guest inputs information.				
3	Guest sends command to login to system.	Guest logs system with his specific role. [Alternative 1] [Exception 1]			
Alternative Scenario:					
No	Actor Action	System Response			
1	Guest enter wrong identity information.	System shows error message "Wrong username or password. Please try again."			
Exceptions:					
No	Actor Action	System Response			
1		System shows message the "Please check your connection!" when the internet is lost.			
Relationships: N/A					
Business Rules:					
<ul style="list-style-type: none"> - If success, user will gain an access token to have authorization in other actions. - After logging in, the device token is saved for pushing notification. 					

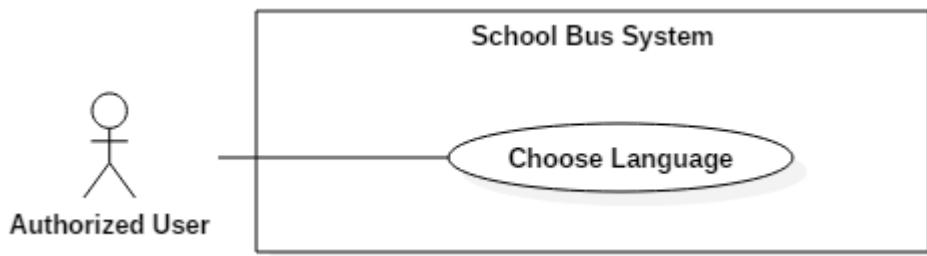
<Guest> Login Use Case Specification

2.3.2 <Authorized User> Overview Use Case



Authorized User Overview Use Case

2.3.2.1 <Authorized User> Choose Language



<Authorized User> Choose Language

USE CASE - SCB_UC_3						
Use Case No.	SCB_UC_3	Use Case Version	1.0			
Use Case Name	Choose Language					
Author	TrungTH					
Date	13/2/2019	Priority	Normal			
Actor:	<ul style="list-style-type: none"> - Authorized User 					
Summary:	<ul style="list-style-type: none"> - Allow authorized user to choose language. 					
Goal:	<ul style="list-style-type: none"> - Authorized user could choose language. 					
Triggers:	<ul style="list-style-type: none"> - Authorized user sends choosing language command. 					
Preconditions:	<ul style="list-style-type: none"> - Authorized user must login first. - Authorized user must be in settings tab. 					
Post conditions:	<ul style="list-style-type: none"> - Success: Authorized user chooses language successfully, chosen language is shown. - Fail: System shows error messages. 					
Main Success Scenario:	<table border="1"> <thead> <tr> <th>Step</th><th>Actor Action</th><th>System Response</th></tr> </thead> </table>			Step	Actor Action	System Response
Step	Actor Action	System Response				

1	Authorized user goes to profile view.	Sidebar with profile's information on top, and some functional links: <ul style="list-style-type: none"> - My request/service - Create new request/service - My children - About us - Settings - Logout
2	Authorized user goes to settings view	Settings view with settings for: <ul style="list-style-type: none"> - Language
3	Authorized user chooses language	Language is changed.

Alternative Scenario:

No	Actor Action	System Response
----	--------------	-----------------

Exceptions:

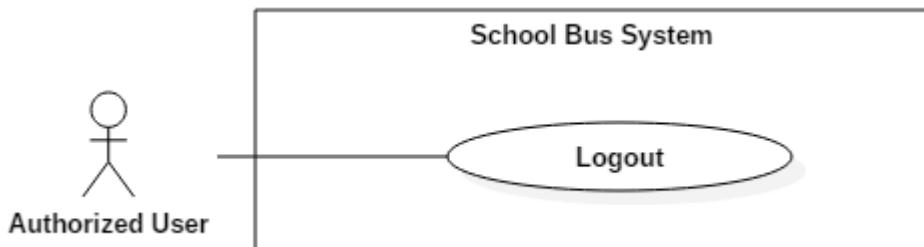
No	Actor Action	System Response
----	--------------	-----------------

Relationships: [Login Use Case]**Business Rules:**

- Language type must be saved in storage for future use.

<Authorized User> Choose Language Case Specification

2.3.2.2 <Authorized User> Logout



<Authorized User> Logout

USE CASE – SCB_UC_4			
Use Case No.	SCB_UC_4	Use Case Version	1.0
Use Case Name	Logout		
Author	TrungTH		
Date	13/2/2019	Priority	Normal
Actor:	<ul style="list-style-type: none"> - Authorized User 		
Summary:	<ul style="list-style-type: none"> - Allow authorized user to logout. 		
Goal:	<ul style="list-style-type: none"> - Authorized user could logout. 		

Triggers:

- Authorized user sends logout command.

Preconditions:

- Authorized user must login first.
- Authorized user must be in view profile tab.

Post conditions:

- Success: Authorized user logs out successfully.
- Fail: System shows error messages.

Main Success Scenario:

Step	Actor Action	System Response
1	Authorized user goes to profile view.	Sidebar with profile's information on top, and some functional links: <ul style="list-style-type: none"> - My request/service - Create new request/service - My children - About us - Settings - Logout
2	Authorized user sends command to logout.	Authorized user logs out. [Exception 1]

Alternative Scenario:

No	Actor Action	System Response

Exceptions:

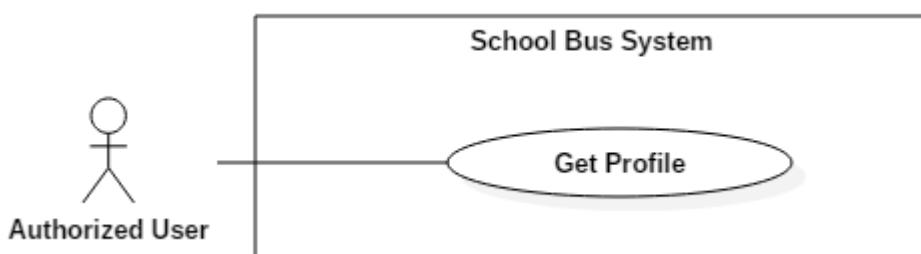
No	Actor Action	System Response
1		System shows message "Please check your connection!" when the internet is lost.

Relationships: [Login Use Case] [Get Profile Use Case]**Business Rules:**

- Access token and other data will be removed from storage.
- Device token will be deleted so that guest won't receive notification.
- After logging out, authorized user becomes guest and will be redirected to login screen.

<Authorized User> Logout Use Case Specification

2.3.2.3 <Authorized User> Get Profile



<Authorized User> Get Profile

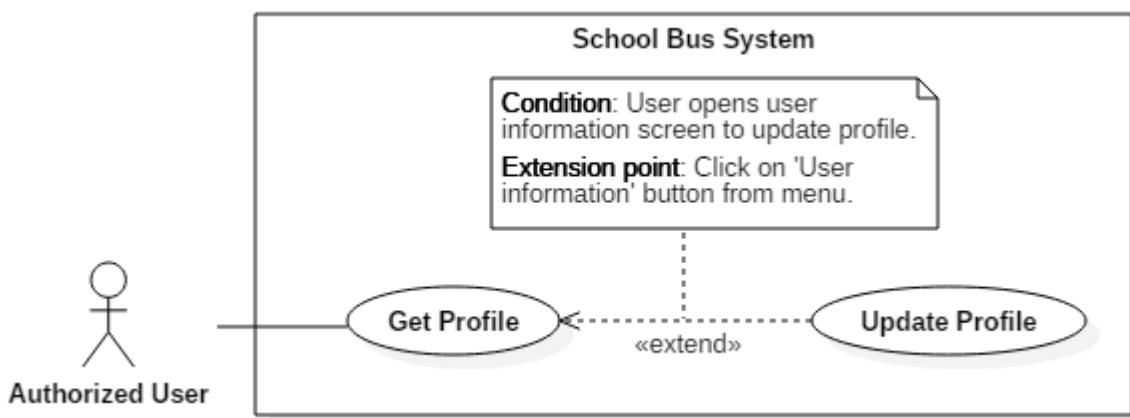
USE CASE – SCB_UC_5			
Use Case No.	SCB_UC_5	Use Case Version	1.0
Use Case Name	Get Profile		
Author	TrungTH		
Date	13/2/2019	Priority	Normal
Actor:	<ul style="list-style-type: none"> - Authorized user 		
Summary:	<ul style="list-style-type: none"> - Allow authorized user to view profile 		
Goal:	<ul style="list-style-type: none"> - Authorized user could view profile. 		
Triggers:	<ul style="list-style-type: none"> - Authorized user sends view profile command. 		
Preconditions:	<ul style="list-style-type: none"> - Authorized user must login first. 		
Post conditions:	<ul style="list-style-type: none"> - Success: Authorized user logout successfully. - Fail: System shows error messages. 		
Main Success Scenario:			
Step	Actor Action	System Response	
1	Authorized user sends get profile command	<p>Sidebar with profile's information on top:</p> <ul style="list-style-type: none"> - Name - Avatar - Feedback score (if driver) <p>And some functional links:</p> <ul style="list-style-type: none"> - My request/service - Create new request/service - My children (if customer) - My car (if driver) - About us - Settings - Logout <p>[Exception 1]</p>	
Alternative Scenario:			
No	Actor Action	System Response	
Exceptions:			
No	Actor Action	System Response	
1		System shows message the "Please check your connection!" when the internet is lost.	
Relationships: [Login Use Case]			

Business Rules:

- Authorized user will be redirected to view profile screen.

<Authorized User> View Profile Use Case Specification

2.3.2.4 <Authorized User> Update Profile



<Authorized User> Update Profile

USE CASE - SCB_UC_6									
Use Case No.	SCB_UC_6	Use Case Version	1.0						
Use Case Name	Update Profile								
Author	TrungTH								
Date	13/2/2019	Priority	Normal						
Actor:	<ul style="list-style-type: none"> - Authorized user 								
Summary:	<ul style="list-style-type: none"> - Allow authorized users to update profile. 								
Goal:	<ul style="list-style-type: none"> - Authorized user could update profile. 								
Triggers:	<ul style="list-style-type: none"> - Authorized user sends view profile command. 								
Preconditions:	<ul style="list-style-type: none"> - Authorized user must login first. - Authorized user must see profile first. 								
Post conditions:	<ul style="list-style-type: none"> - Success: Authorized user updates profile successfully. - Fail: System shows error messages. 								
Main Success Scenario:	<table border="1"> <thead> <tr> <th>Step</th><th>Actor Action</th><th>System Response</th></tr> </thead> <tbody> <tr> <td>1</td><td>Authorized user sends going to update profile view command</td><td> Profile screen with profile's information: <ul style="list-style-type: none"> - Name - Avatar - Address - Identification Number </td></tr> </tbody> </table>			Step	Actor Action	System Response	1	Authorized user sends going to update profile view command	Profile screen with profile's information: <ul style="list-style-type: none"> - Name - Avatar - Address - Identification Number
Step	Actor Action	System Response							
1	Authorized user sends going to update profile view command	Profile screen with profile's information: <ul style="list-style-type: none"> - Name - Avatar - Address - Identification Number 							

		And button "Update" [Exception 1]
2	Authorized user edits profile	
3	Authorized user sends update profile command	<p>Profile editing screen with these input fields:</p> <ul style="list-style-type: none"> - Name - Address - Identification number <p>And avatar image choosing</p>

Alternative Scenario:

No	Actor Action	System Response
1	Authorized user enters wrong format.	System shows appropriate validating message. Ex: "Name must not contain numbers", "All fields are not blank" ...

Exceptions:

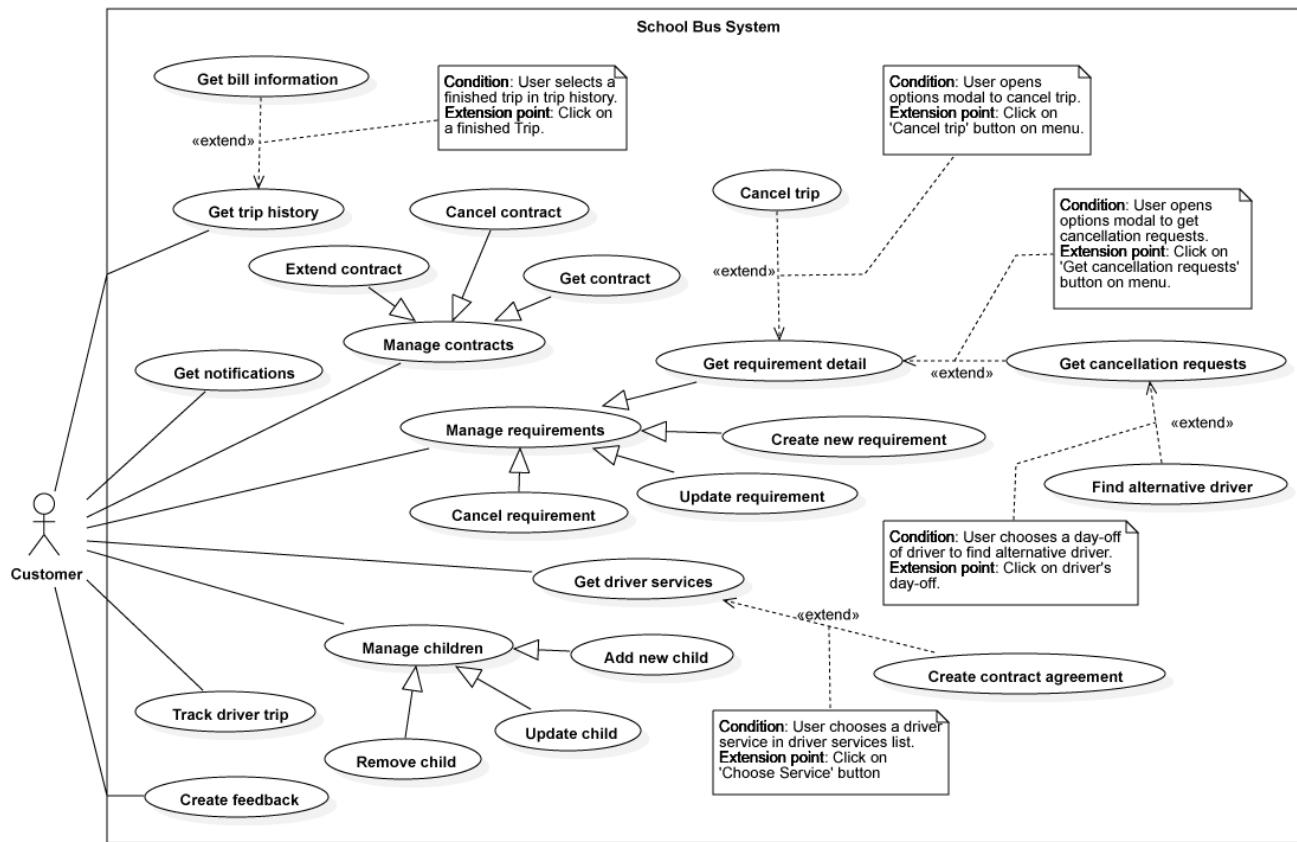
No	Actor Action	System Response
1		System shows message the "Please check your connection!" when the internet is lost.

Relationships: [Login Use Case] [Get Profile Use Case]**Business Rules:**

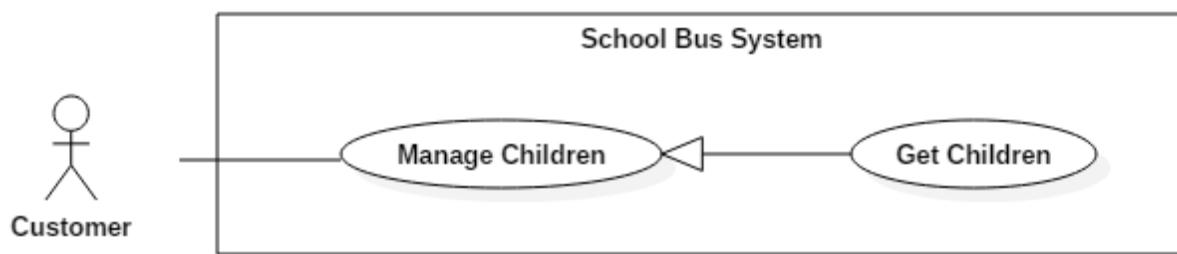
- Driver can only update his/her profile once after registering.

<Authorized User> Update Profile Use Case Specification

2.3.3 <Customer> Overview Use Case



2.3.3.1 <Customer> Get Children



USE CASE - SCB_UC_7			
Use Case No.	SCB_UC_7	Use Case Version	1.0
Use Case Name	Get Children		
Author	TrungTH		
Date	13/2/2019	Priority	Normal
Actor:	<ul style="list-style-type: none"> - Customer 		
Summary:	<ul style="list-style-type: none"> - Allow customers to view their children. 		
Goal:	<ul style="list-style-type: none"> - Customer could view their children. 		

Triggers:

- Customer sends viewing children command.

Preconditions:

- Customer must login first.

Post conditions:

- Success: Customer could view their children
- Fail: System shows error messages.

Main Success Scenario:

Step	Actor Action	System Response
1	Customer sends going to children view command (from sidebar or home screen)	<p>Children screen with list of children and their information:</p> <ul style="list-style-type: none"> - Name - Avatar - School - Class <p>And one adding children button</p> <p>[Exception 1]</p>

Alternative Scenario:

No	Actor Action	System Response

Exceptions:

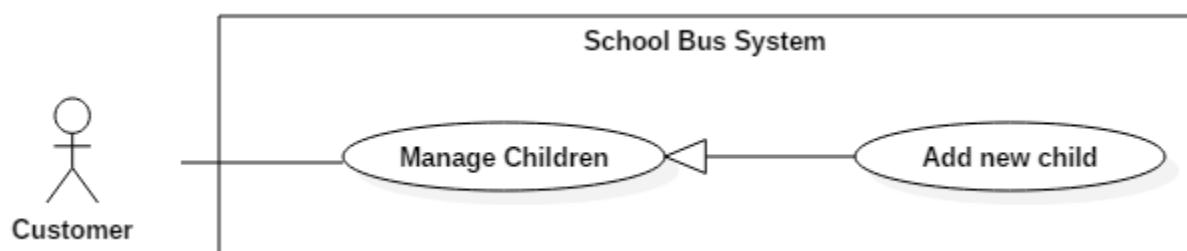
No	Actor Action	System Response
1		System shows message the "Please check your connection!" when the internet is lost.

Relationships: [Login Use Case]**Business Rules:**

- Removed children are not shown.

<Customer> Get Children Use Case Specification

2.3.3.2 <Customer> Add new child



<Customer> Add new child

USE CASE – SCB_UC_8			
Use Case No.	SCB_UC_8	Use Case Version	1.0

Use Case Name	Add new child		
Author	TrungTH		
Date	13/2/2019	Priority	Normal

Actor:

- Customer

Summary:

- Allow customer to add new children

Goal:

- Customer could add new children

Triggers:

- Customer sends adding new children command

Preconditions:

- Customer must login first.
- Customer must be in viewing children screen.

Post conditions:

- Success: Customer add new children successfully.
- Fail: System shows error messages.

Main Success Scenario:

Step	Actor Action	System Response
1	Customer sends going to children view command (from sidebar or home screen)	<p>Children screen with list of children and their information:</p> <ul style="list-style-type: none"> - Name - Avatar - School - Class <p>And one adding children button</p> <p>[Exception 1]</p>
2	Customer sends going to adding children view command	<p>Adding children screen with following fields:</p> <ul style="list-style-type: none"> - Avatar: Image Picker - Name: Text Input - Birthday: Date Picker - Class: Text Input - School: Place Suggestion
3	Customer inputs the child's information	
3	Customer sends adding new child command	Customer added new child [Alternate 1] [Exception 1]

Alternative Scenario:

No	Actor Action	System Response
1	Customer enters wrong format	System shows appropriate validating message. Ex: "Name must not contain numbers", "All fields are not blank" ...

Exceptions:

No	Actor Action	System Response
1		System shows message the "Please check your connection!" when the internet is lost.

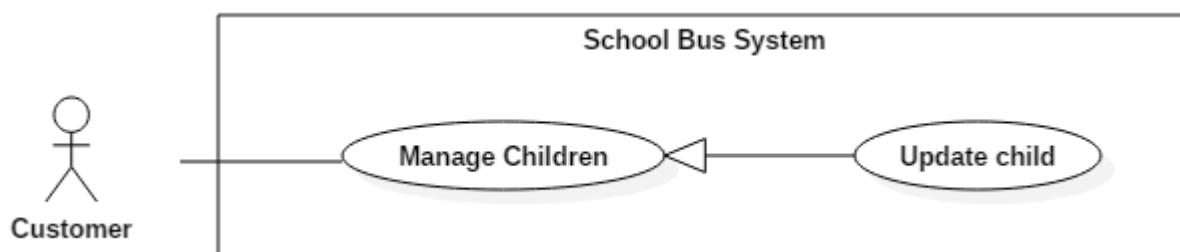
Relationships: [Login Use Case] [Get Children Use Case]

Business Rules:

- User will be redirected to viewing children screen after adding children.

<Customer> Add new child Use Case Specification

2.3.3.3 <Customer> Update child



<Customer> Update child

USE CASE - SCB_UC_9			
Use Case No.	SCB_UC_9	Use Case Version	1.0
Use Case Name	Update child		
Author	TrungTH		
Date	13/2/2019	Priority	Normal
Actor:			
- Customer			
Summary:			
- Allow customer to update children.			
Goal:			
- Customer could update children.			
Triggers:			
- Customer sends updating children command.			
Preconditions:			
- Customer must login first.			
- Customer must be in viewing children screen.			
Post conditions:			
- Success: Customer updates children successfully.			
- Fail: System shows error messages.			
Main Success Scenario:			
Step	Actor Action	System Response	
1	Customer sends going to updating children view command	Updating children screen with children's information: - Name - Avatar	

		<ul style="list-style-type: none"> - Birthday - School - Class <p>And button "Update" [Exception 1]</p>
2	Customer edits child's profile	
3	Customer sends updating children command	Customer updates child's profile [Exception 1]

Alternative Scenario:

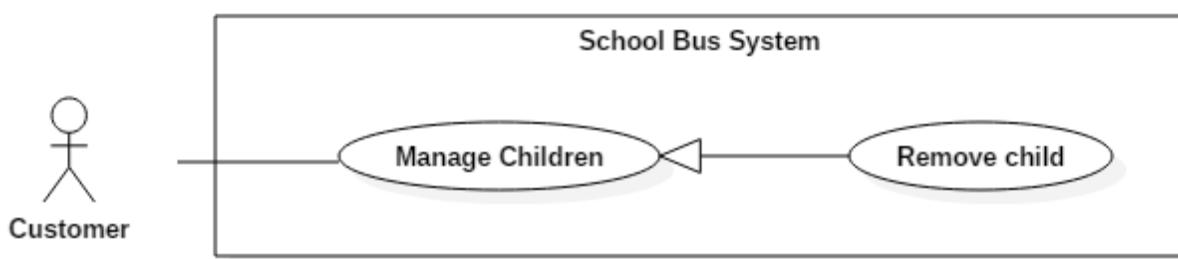
No	Actor Action	System Response
1	Customer enters wrong format	System shows appropriate validating message. Ex: "Name must not contain numbers", "All fields are not blank" ...

Exceptions:

No	Actor Action	System Response
1		System shows message the "Please check your connection!" when the internet is lost.

Relationships: [Login Use Case] [Get Children Use Case]**Business Rules:**

- After updating child, customer will be redirected to viewing children screen.

*<Customer> Update child Use Case Specification***2.3.3.4 <Customer> Remove child***<Customer> Remove child*

USE CASE - SCB_UC_10			
Use Case No.	SCB_UC_10	Use Case Version	1.0
Use Case Name	Remove child		
Author	TrungTH		
Date	13/2/2019	Priority	Normal
Actor:	<ul style="list-style-type: none"> - Customer 		
Summary:			

- Allow customers to remove child.

Goal:

- Customer could remove child.

Triggers:

- Customer sends removing child command.

Preconditions:

- Customer must login first.
- Customer must be in viewing children screen.

Post conditions:

- Success: Customer deletes children successfully.
- Fail: System shows error messages.

Main Success Scenario:

Step	Actor Action	System Response
1	Customer sends removing child command	Popup alert for confirming deletion.
2	Customer confirms removing children.	Customer removes children successfully. [Alternate 1] [Exception 1]

Alternative Scenario:

No	Actor Action	System Response
1	Customer confirms "No" for removing children.	Going back to the viewing children screen

Exceptions:

No	Actor Action	System Response
1		System shows message the "Please check your connection!" when the internet is lost.

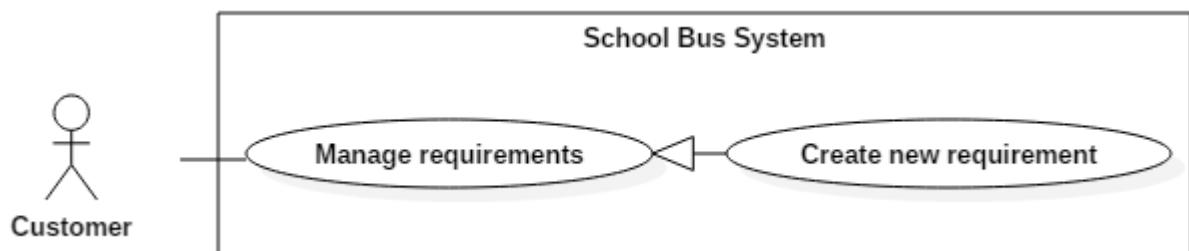
Relationships: [Login Use Case] [Get Profile Use Case]

Business Rules:

- Children which are in requirements having active contracts could not be removed.
- Removed children are not shown in viewing children screen.

<Customer> Remove child Use Case Specification

2.3.3.5 <Customer> Create new requirement



<Customer> Create new requirement

USE CASE - SCB_UC_11

Use Case No.	SCB_UC_11	Use Case Version	1.0
Use Case Name	Create new requirement		
Author	TrungTH		
Date	13/2/2019	Priority	Normal

Actor:

- Customer

Summary:

- Allow customers to create transporting service requirements.

Goal:

- Customer could create transporting service requirements.

Triggers:

- Customer sends creating transporting service requirements command.

Preconditions:

- Customer must login first.

Post conditions:

- Success: Customer create new transporting service requirements successfully.
- Fail: System shows error messages.

Main Success Scenario:

Step	Actor Action	System Response
1	Customer goes to creating transporting service requirements screen	<p>Creating requirements screen with following fields:</p> <ul style="list-style-type: none"> - Days of week (multiple weekday picker) - Pick up address (place suggestion) - Pick up location (place suggestion) - Pick up time (time picker) - Arrival time (time picker) - Return time (time picker) - Start date (date picker) - End date (date picker)
2	Customer inputs transporting service requirements	
3	Customer sends creating transporting service requirements command	Customer register new transporting service requirements [Alternate 1] [Alternate 2] [Exception 1]

Alternative Scenario:

No	Actor Action	System Response
1	Customer leaves fields blank	System shows appropriate validating message. Ex: " You must choose pick up address", ...
2	Datetime logic constraints	Date-Time should be valid. Ex: "Pickup time must before arrival time, arrival

		time must before return time, start date must before end date."
3	Requirement logic constraints	New requirement must not have duplicate properties with old ones, like: "Days of week, children, timeline, ..."

Exceptions:

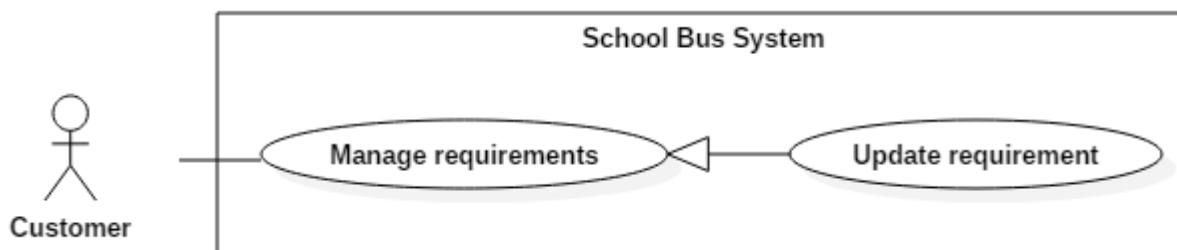
No	Actor Action	System Response
1		System shows message the "Please check your connection!" when the internet is lost.

Relationships: [Login Use Case]**Business Rules:**

- Address coordinates must be provided.
- Pickup address should be suggested as customer's address, school address should be suggested as children's school.
- Unit price and estimated total price must be shown.

<Customer> Create new requirement Use Case Specification

2.3.3.6 <Customer> Update requirement



<Customer> Update requirement

USE CASE - SCB_UC_12			
Use Case No.	SCB_UC_12	Use Case Version	1.0
Use Case Name	Update requirement		
Author	TrungTH		
Date	13/2/2019	Priority	Normal
Actor:	<ul style="list-style-type: none"> - Customer 		
Summary:	<ul style="list-style-type: none"> - Allow customers to update transporting service requirements. 		
Goal:	<ul style="list-style-type: none"> - Customer could update transporting service requirements. 		
Triggers:	<ul style="list-style-type: none"> - Customer sends updating transporting service requirements command. 		
Preconditions:	<ul style="list-style-type: none"> - Customer must login first. 		

- Customer must be in viewing requirement list screen.

Post conditions:

- Success: Customer updates new transporting service requirements successfully.
- Fail: System shows error messages.

Main Success Scenario:

Step	Actor Action	System Response
1	Customer goes to viewing transporting service requirement screen	<p>Viewing requirement list screen with following data:</p> <ul style="list-style-type: none"> - Children - Driver (Name, Avatar) - School - Pickup time <p>Requirement's weekdays</p>
2	Customer goes to viewing transporting service requirement in detail screen.	<p>Created transporting service requirements screen with following information</p> <ul style="list-style-type: none"> - Days of week - Children - School - Pickup Address - Pick up time - Arrival time - Return time - Start date - End date - Status (finding or contract created) - Contract information () (show if driver is chosen)
3	Customer goes to updating transporting service requirements screen	<p>Updating requirements screen with following fields:</p> <ul style="list-style-type: none"> - Days of week (multiple weekday picker) - Pick up address (place suggestion) - Pick up location (place suggestion) - Pick up time (time picker) - Arrival time (time picker) - Return time (time picker) - Start date (date picker) - End date (date picker)
4	Customer edits transporting service requirements	

5	Customer sends updating transporting service requirements command	Customer register new transporting service requirements [Alternate 1] [Alternate 2] [Exception 1]
---	---	---

Alternative Scenario:

No	Actor Action	System Response
1	Customer leaves fields blank	System shows appropriate validating message. Ex: " You must choose pick up address", ...
2	Datetime logic constraints	Date-Time should be valid. Ex: "Pickup time must before arrival time, arrival time must before return time, start date must before end date."
3	Requirement logic constraints	Updated requirement must not have duplicate properties with old ones, like: "Days of week, children, timeline, ..."

Exceptions:

No	Actor Action	System Response
1		System shows message the "Please check your connection!" when the internet is lost.

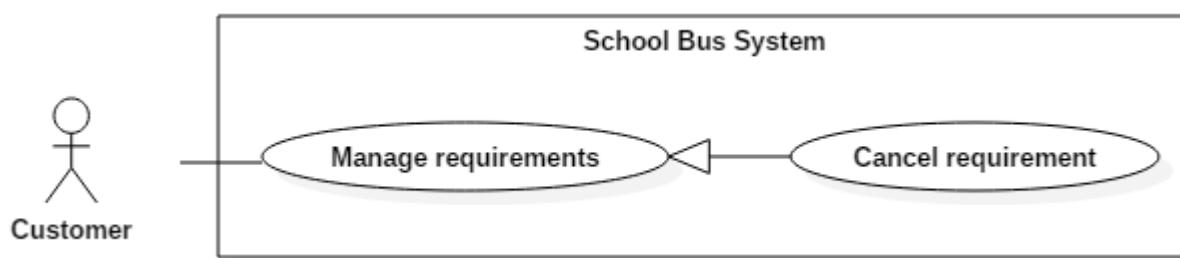
Relationships: [Get Requirement Detail Use Case] [Login Use Case]

Business Rules:

- Requirements which are in active contracts are unable to update.
- Address coordinates must be provided.
- Pickup address should be suggested as customer's address, school address should be suggested as children's school.
- Unit price and estimated total price must be shown.

<Customer> Update requirement Use Case Specification

2.3.3.7 <Customer> Cancel requirement



<Customer> Cancel requirement

USE CASE - SCB_UC_13			
Use Case No.	SCB_UC_13	Use Case Version	1.0
Use Case Name	Cancel requirement		

Author	TrungTH		
Date	13/2/2019	Priority	Normal
Actor:			
- Customer			
Summary:			
- Allow customers to delete transporting service requirements.			
Goal:			
- Customer could delete transporting service requirements.			
Triggers:			
- Customer sends deleting transporting service requirements command.			
Preconditions:			
- Customer must login first.			
- Customer must be in viewing requirement list screen.			
- Requirements don't meet any contract.			
Post conditions:			
- Success: Customer deletes new transporting service requirements successfully.			
- Fail: System shows error messages.			
Main Success Scenario:			
Step	Actor Action	System Response	
1	Customer goes to viewing transporting service requirement screen	Viewing requirements screen with following data: - Children - Driver (Name, Avatar) - School - Pickup time - Requirement's weekdays	
2	Customer goes to viewing transporting service requirement in detail screen.	Created transporting service requirements screen with following information - Days of week - Children - School - Pickup Address - Pick up time - Arrival time - Return time - Start date - End date - Status (finding or contract created) Contract information () (show if driver is chosen)	
3	Customer sends deleting requirement command	Popup alert for confirming deletion.	
4	Customer confirms deleting requirement.	Customer deletes children successfully. [Alternate 1] [Exception 1]	

Alternative Scenario:

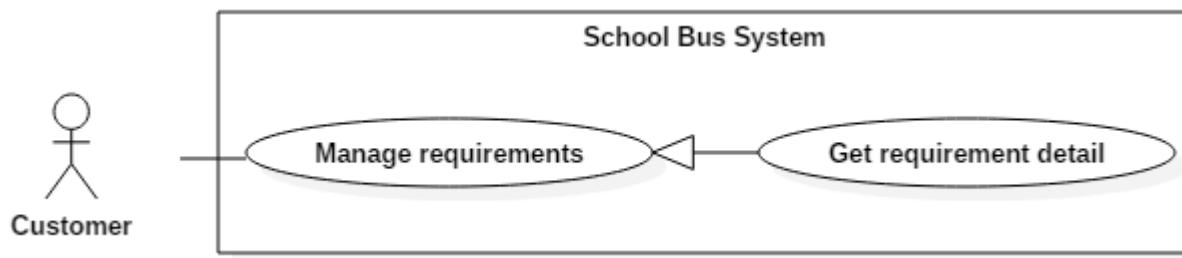
No	Actor Action	System Response
1	Customer confirms "No"	Back to the viewing requirements screen

Exceptions:

No	Actor Action	System Response
1		System shows message the "Please check your connection!" when the internet is lost.

Relationships: [Get Requirement Detail Use Case] [Login Use Case]**Business Rules:**

- Requirements which are in active contracts are unable to delete.
- Confirmation should be shown before deleting.

*<Customer> Cancel requirement Use Case Specification***2.3.3.8 <Customer> Get requirement detail***<Customer> Get requirement detail*

USE CASE - SCB_UC_14			
Use Case No.	SCB_UC_14	Use Case Version	1.0
Use Case Name	Get requirement detail		
Author	TrungTH		
Date	13/2/2019	Priority	Normal
Actor:	<ul style="list-style-type: none"> - Customer 		
Summary:	<ul style="list-style-type: none"> - Allow customers to view transporting service requirement in detail. 		
Goal:	<ul style="list-style-type: none"> - Customer could view transporting service requirement in detail. 		
Triggers:	<ul style="list-style-type: none"> - Customer sends viewing transporting service requirement in detail command. 		
Preconditions:	<ul style="list-style-type: none"> - Customer must login first. - Customer must be in viewing transporting service requirements screen. 		
Post conditions:	<ul style="list-style-type: none"> - Success: Customer could view transporting service requirement in detail. 		

- Fail: System shows error messages.

Main Success Scenario:

Step	Actor Action	System Response
1	Customer goes to viewing transporting service requirement in detail screen.	<p>Created transporting service requirements screen with following information</p> <ul style="list-style-type: none"> - Days of week - Children - School - Pickup Address - Pick up time - Arrival time - Return time - Start date - End date - Status (finding or contract created) - Contract information () (show if driver is chosen) <p>[Exception 1]</p>

Alternative Scenario:

No	Actor Action	System Response
----	--------------	-----------------

Exceptions:

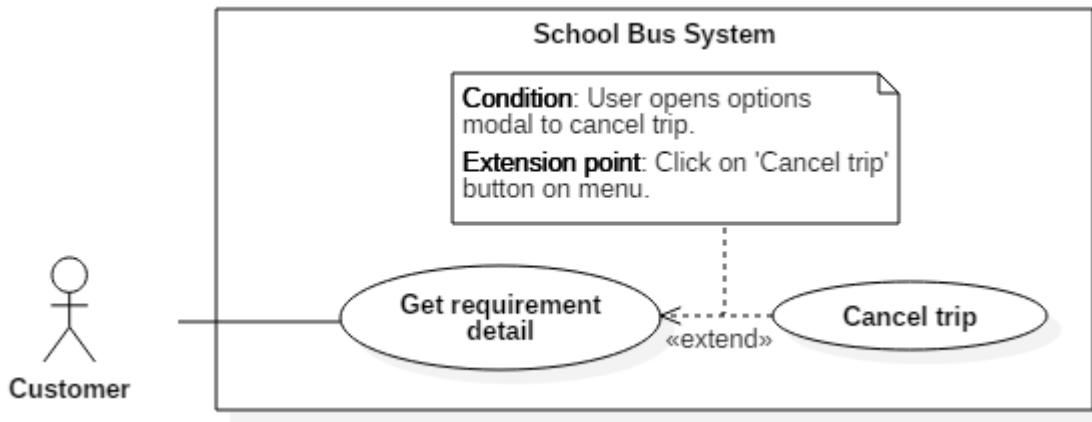
No	Actor Action	System Response
1		System shows message the "Please check your connection!" when the internet is lost.

Relationships: [Login Use Case]**Business Rules:**

- Contract with driver's service is shown with pending or active status.
- Extending detail is shown if there's an extending request.

<Customer> Get requirement detail Use Case Specification

2.3.3.9 <Customer> Cancel Trip



<Customer> Cancel Trip

USE CASE - SCB_UC_15			
Use Case No.	SCB_UC_15	Use Case Version	1.0
Use Case Name	Cancel Trip		
Author	TrungTH		
Date	13/2/2019	Priority	Normal
Actor:			
- Customer			
Summary:			
- Allow customers to cancel some trips.			
Goal:			
- Customer could cancel some trips.			
Triggers:			
- Customer sends cancelling trip command.			
Preconditions:			
- Customer must login first.			
- Customer must be in viewing requirement detail screen.			
Post conditions:			
- Success: Customer cancels trip successfully.			
- Fail: System shows error messages.			
Main Success Scenario:			
Step	Actor Action	System Response	
1	Customer goes to creating trip cancellation request screen.	Creating trip cancellation request screen with following fields: - Children (multiple children picker) - Off day (Date picker) - Reason (List Picker, Text Area)	
2	Customer inputs trip cancellation request	[Alternate 2]	
3	Customer sends creating trip cancellation request command	Customer cancels trip successfully [Alternate 1] [Exception 1]	

Alternative Scenario:

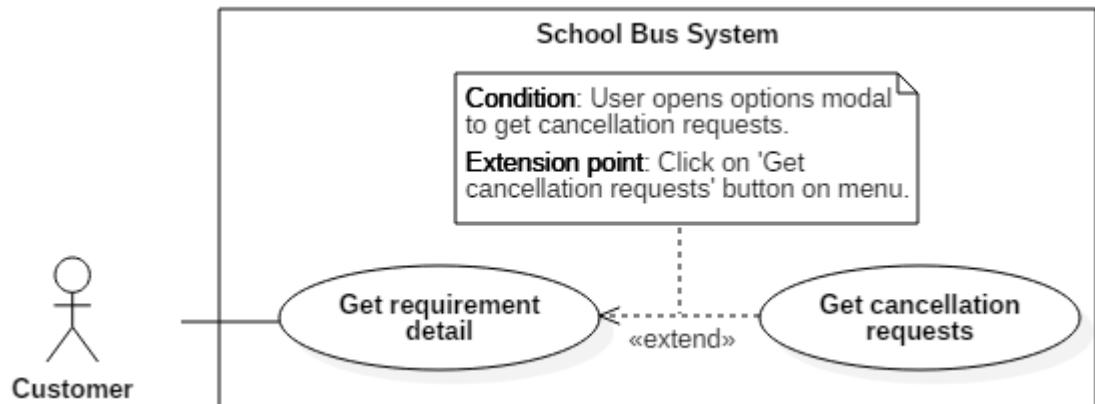
No	Actor Action	System Response
1	Customer enters wrong format	System shows appropriate validating message. Ex: "All fields are not blank"
2	Customer chooses wrong off day	System shows appropriate validating message. Ex: "Your child is not going to school on that day"

Exceptions:

No	Actor Action	System Response
1		System shows message the "Please check your connection!" when the internet is lost.

Relationships: [Get Requirement Detail Use Case] [Login Use Case]**Business Rules:**

- System should notify driver about the cancellation requests.

*<Customer> Cancel Trip Use Case Specification***2.3.3.10 <Customer> Get Cancellation Requests***<Customer> Get Cancellation Requests***USE CASE - SCB_UC_16**

Use Case No.	SCB_UC_16	Use Case Version	1.0
Use Case Name	Get Cancellation Requests		
Author	TrungTH		
Date	13/2/2019	Priority	Normal

Actor:

- Customer

Summary:

- Allow customers to view their trip cancellation requests.

Goal:

- Customer could view their trip cancellation requests.

Triggers:

- Customer sends viewing trip cancellation requests command.

Preconditions:

- Customer must login first.
- Customer must be in viewing transporting service requirement detail screen.

Post conditions:

- Success: Customer could view their trip cancellation requests.
- Fail: System shows error messages.

Main Success Scenario:

Step	Actor Action	System Response
1	Customer goes to viewing transporting service requirement in detail screen.	<p>Created transporting service requirements screen with following information</p> <ul style="list-style-type: none"> - Days of week - Children - School - Pickup Address - Pick up time - Arrival time - Return time - Start date - End date - Status (finding or contract created) - Contract information () (show if driver is chosen)
2	Customer goes to viewing trip cancellation request screen.	<p>Cancellation requests are shown in two kinds: customer's and driver's, with following properties:</p> <ul style="list-style-type: none"> - Children list (customer's) - Off Day - Description - Hyperlink switching 2 types. <p>[Exception 1]</p>

Alternative Scenario:

No	Actor Action	System Response
-----------	---------------------	------------------------

Exceptions:

No	Actor Action	System Response
1		System shows message the "Please check your connection!" when the internet is lost.

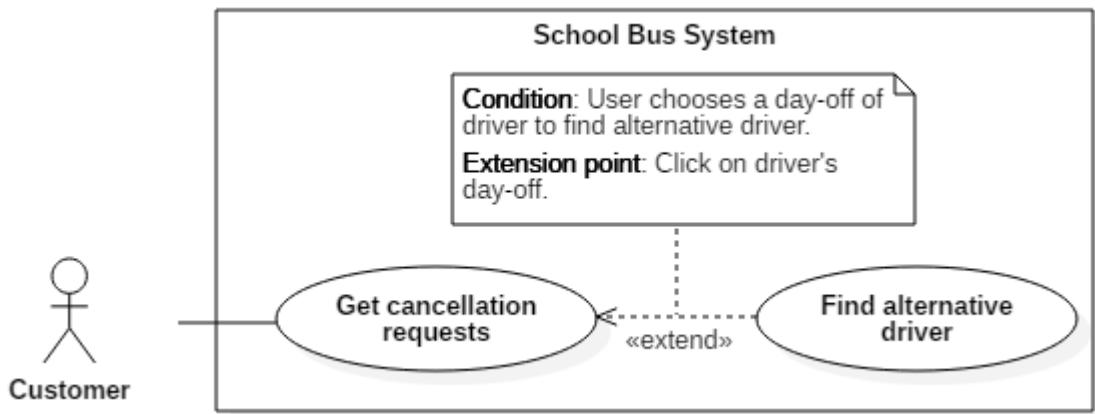
Relationships: [Login Use Case] [Get Requirement Detail Use Case]

Business Rules:

- Cancellation requests should be ordered descending by time.

<Customer> Get Cancellation Requests Use Case Specification

2.3.3.11 <Customer> Find alternative driver



<Customer> Find Alternative Driver

USE CASE - SCB_UC_17									
Use Case No.	SCB_UC_17	Use Case Version	1.0						
Use Case Name	Find Alternative Driver								
Author	TrungTH								
Date	13/2/2019	Priority	Normal						
Actor:	<ul style="list-style-type: none"> - Customer 								
Summary:	<ul style="list-style-type: none"> - Allow customer to find an alternative driver 								
Goal:	<ul style="list-style-type: none"> - Customer could find an alternative driver 								
Triggers:	<ul style="list-style-type: none"> - Customer sends finding alternative driver command. 								
Preconditions:	<ul style="list-style-type: none"> - Customer must login. - There are driver's off days in the current contract. 								
Post conditions:	<ul style="list-style-type: none"> - Success: Customer find temporary driver successfully. - Fail: System shows error messages. 								
Main Success Scenario:	<table border="1"> <thead> <tr> <th>Step</th><th>Actor Action</th><th>System Response</th></tr> </thead> <tbody> <tr> <td>1</td><td>Customer goes to getting cancellation requests screen.</td><td> <p>'View Cancelled Trip' screen with following properties:</p> <ul style="list-style-type: none"> - Customer's cancelled trips: children, off-day, content - Driver's cancelled trips: off-day, content </td></tr> </tbody> </table>			Step	Actor Action	System Response	1	Customer goes to getting cancellation requests screen.	<p>'View Cancelled Trip' screen with following properties:</p> <ul style="list-style-type: none"> - Customer's cancelled trips: children, off-day, content - Driver's cancelled trips: off-day, content
Step	Actor Action	System Response							
1	Customer goes to getting cancellation requests screen.	<p>'View Cancelled Trip' screen with following properties:</p> <ul style="list-style-type: none"> - Customer's cancelled trips: children, off-day, content - Driver's cancelled trips: off-day, content 							

2	Customer chooses to view the driver's cancellation requests	
3	Customer presses a driver cancel request for opening two options	<ul style="list-style-type: none"> - Alternative driver information if chosen - Option to go to appropriate driver's services for a temporary contract <p>[Exception 1]</p>

Alternative Scenario:

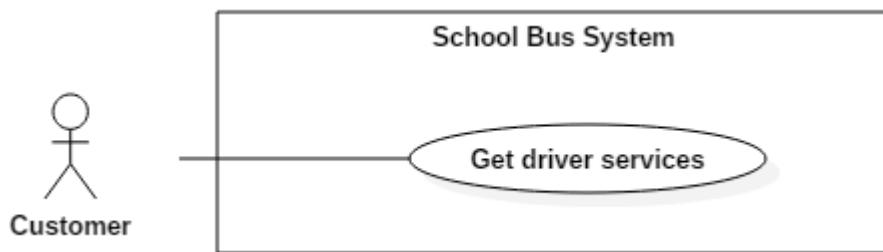
No	Actor Action	System Response
----	--------------	-----------------

Exceptions:

No	Actor Action	System Response
1		System shows message the "Please check your connection!" when the internet is lost.

Relationships: N/A**Business Rules:**

- There are driver's off days in the current contract.

*<Customer> Find Alternative Driver Use Case Specification***2.3.3.12 <Customer> Get Driver Services***<Customer> Get Driver Services*

USE CASE – SCB_UC_18			
Use Case No.	SCB_UC_18	Use Case Version	1.0
Use Case Name	Get driver services		
Author	TrungTH		
Date	13/2/2019	Priority	Normal
Actor:	<ul style="list-style-type: none"> - Customer 		
Summary:	<ul style="list-style-type: none"> - Allow customers to get driver's transporting service. 		
Goal:	<ul style="list-style-type: none"> - Customer could get driver's transporting service. 		

Triggers:

- Customer sends getting driver's transporting service command.

Preconditions:

- Customer must login first.
- Customer must create transporting service requirement.

Post conditions:

- Success: Customer gets driver's transporting service successfully.
- Fail: System shows error messages.

Main Success Scenario:

Step	Actor Action	System Response
1	Customer goes to getting driver services screen	<p>Getting driver services screen with following information:</p> <ul style="list-style-type: none"> - Driver's name - Driver's avatar - Driver's phone - Car's info: Plate number, Brand, Model, Color - Driver's average score based on previous feedbacks <p>[Alternate 1] [Exception 1]</p>

Alternative Scenario:

No	Actor Action	System Response
1	Driver service is not found	Display some announcement, like "Driver service is not found yet"

Exceptions:

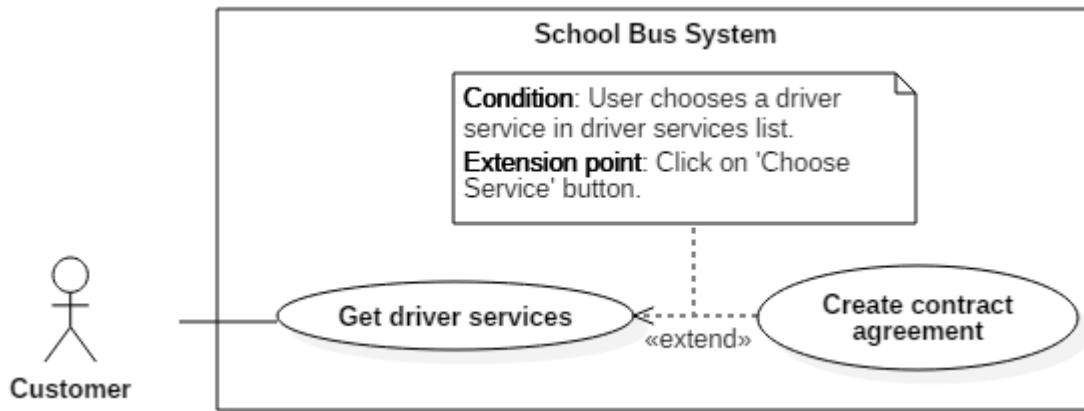
No	Actor Action	System Response
1		System shows message the "Please check your connection!" when the internet is lost.

Relationships: [Login Use Case] [Create Requirement Use Case]**Business Rules:**

- Appropriate driver services are matched based on matching percentage calculated with following conditions: in-range start point and same destination, days in week, timeline, start and end date, available capacity.

<Customer> Get Driver Services Use Case Specification

2.3.3.13 <Customer> Create Contract Agreement



<Customer> Create Contract Agreement

USE CASE - SCB_UC_19			
Use Case No.	SCB_UC_19	Use Case Version	1.0
Use Case Name	Create Contract Agreement		
Author	TrungTH		
Date	13/2/2019	Priority	Normal
Actor:	<ul style="list-style-type: none"> - Customer 		
Summary:	<ul style="list-style-type: none"> - Allow customers to create contract agreement with driver. 		
Goal:	<ul style="list-style-type: none"> - Customer could create contract agreement with driver. 		
Triggers:	<ul style="list-style-type: none"> - Customer sends creating contract agreement with driver command. 		
Preconditions:	<ul style="list-style-type: none"> - Customer must login first. - Customer must create transporting service requirement. - Customer must be in viewing driver services screen. 		
Post conditions:	<ul style="list-style-type: none"> - Success: Customer create contract agreement with driver successfully. - Fail: System shows error messages. 		
Main Success Scenario:			
Step	Actor Action	System Response	
1	Customer goes to getting driver services screen	Getting driver services screen with following information: <ul style="list-style-type: none"> - Driver's name - Driver's avatar - Driver's phone - Car's info: Plate number, Brand, Model, Color 	
2	Customer chooses driver	Alert box for confirming contract with driver	

3	Customer confirms creating contract agreement	Customer creates contract agreement successfully. [Alternate 1] [Exception 1]
---	---	---

Alternative Scenario:

No	Actor Action	System Response
1	Customer choose "No"	Continue to the getting driver services screen

Exceptions:

No	Actor Action	System Response
1		System shows message the "Please check your connection!" when the internet is lost.

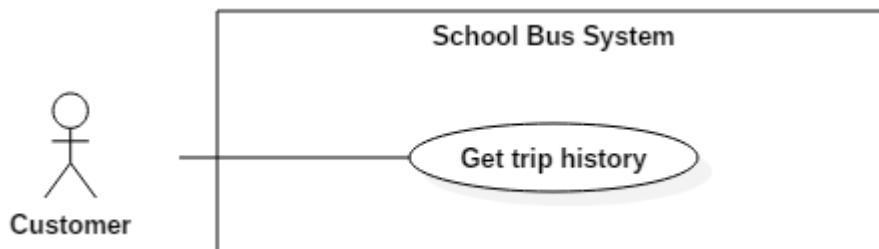
Relationships: [Login Use Case] [Create Requirement Use Case] [Get Driver Service Use Case]

Business Rules:

- After contract agreement is created, we consider that available capacity in driver's provided service is occupied by children in requirement.
- After creating contract agreement, customer would be redirected to viewing transporting service requirements screen.
- System should push notification to driver to let them know about the contract.

<Customer> Create Contract Agreement Use Case Specification

2.3.3.14 <Customer> Get Trip History



<Customer> Get Trip History

USE CASE – SCB_UC_20			
Use Case No.	SCB_UC_20	Use Case Version	1.0
Use Case Name	Get Trip History		
Author	TrungTH		
Date	13/2/2019	Priority	Normal
Actor:	<ul style="list-style-type: none"> - Customer 		
Summary:	<ul style="list-style-type: none"> - Allow customers to get history of all trips. 		
Goal:			

- Customer could get history of all trips.

Triggers:

- Customer sends getting history of all trips command.

Preconditions:

- Customer must login first.

Post conditions:

- Success: Customer gets history of all trips successfully.
- Fail: System shows error messages.

Main Success Scenario:

Step	Actor Action	System Response
1	Customer goes to getting history of all trips screen	<p>Getting history of all trips screen with following information:</p> <ul style="list-style-type: none"> - Trip's date - Trip's status - Driver's avatar - Driver's name - Children's avatar - Children's name <p>Trip status is separated into 2 kinds: 'on going' and 'history'</p> <p>[Exception 1]</p>

Alternative Scenario:

No	Actor Action	System Response
----	--------------	-----------------

Exceptions:

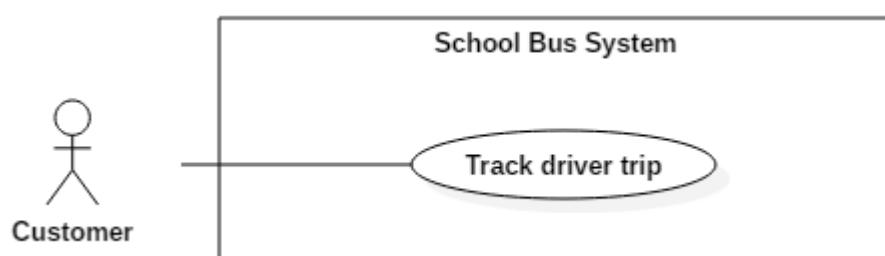
No	Actor Action	System Response
1		System shows message the "Please check your connection!" when the internet is lost.

Relationships: [Login Use Case]**Business Rules:**

- All finished or cancelled or even current trips must be listed.
- Trip must be ordered descending by time.

<Customer> Get Trip History Use Case Specification

2.3.3.15 <Customer> Track Driver Trip



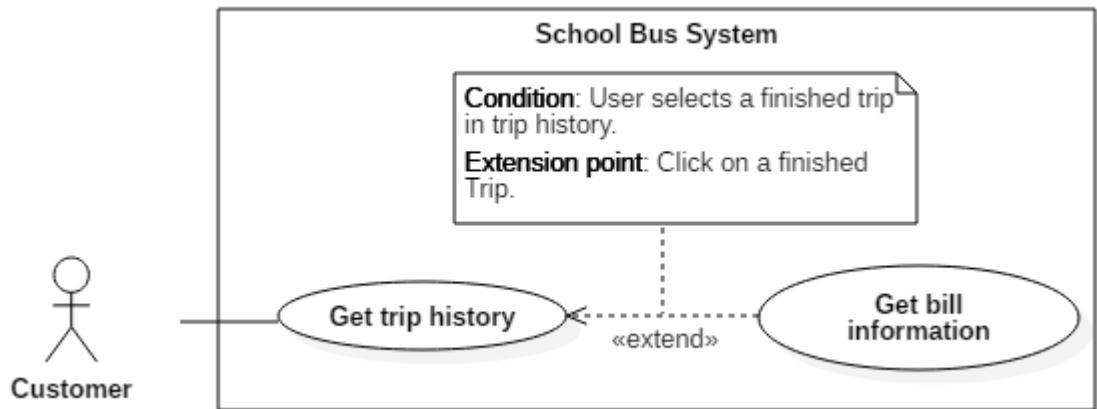
<Customer> Track Driver Trip

USE CASE - SCB_UC_21												
Use Case No.	SCB_UC_21	Use Case Version	1.0									
Use Case Name	Track Driver Trip											
Author	TrungTH											
Date	13/2/2019	Priority	Normal									
Actor:	<ul style="list-style-type: none"> - Customer 											
Summary:	<ul style="list-style-type: none"> - Allow customers to track driver's current trip. 											
Goal:	<ul style="list-style-type: none"> - Customer could track driver's current trip. 											
Triggers:	<ul style="list-style-type: none"> - Customer sends tracking driver's current trip command. 											
Preconditions:	<ul style="list-style-type: none"> - Customer must login first. - Customer must be in viewing trip history screen/getting requirement detail screen - There is a contract. - The trip must be in "On going" status. 											
Post conditions:	<ul style="list-style-type: none"> - Success: Customer tracks driver's current trip successfully. - Fail: System shows error messages. 											
Main Success Scenario:	<table border="1"> <thead> <tr> <th>Step</th><th>Actor Action</th><th>System Response</th></tr> </thead> <tbody> <tr> <td>1</td><td>Customer goes to history of all trips screen or notification screen.</td><td></td></tr> <tr> <td>2</td><td>Customer goes to tracking driver's current trip screen</td><td> Integrated map with following information: <ul style="list-style-type: none"> - Driver's current position by marker - Driver's current route [Exception 1] </td></tr> </tbody> </table>			Step	Actor Action	System Response	1	Customer goes to history of all trips screen or notification screen.		2	Customer goes to tracking driver's current trip screen	Integrated map with following information: <ul style="list-style-type: none"> - Driver's current position by marker - Driver's current route [Exception 1]
Step	Actor Action	System Response										
1	Customer goes to history of all trips screen or notification screen.											
2	Customer goes to tracking driver's current trip screen	Integrated map with following information: <ul style="list-style-type: none"> - Driver's current position by marker - Driver's current route [Exception 1] 										
Alternative Scenario:	<table border="1"> <thead> <tr> <th>No</th><th>Actor Action</th><th>System Response</th></tr> </thead> </table>			No	Actor Action	System Response						
No	Actor Action	System Response										
Exceptions:	<table border="1"> <thead> <tr> <th>No</th><th>Actor Action</th><th>System Response</th></tr> </thead> <tbody> <tr> <td>1</td><td></td><td>System shows message the "Please check your connection!" when the internet is lost.</td></tr> </tbody> </table>			No	Actor Action	System Response	1		System shows message the "Please check your connection!" when the internet is lost.			
No	Actor Action	System Response										
1		System shows message the "Please check your connection!" when the internet is lost.										
Relationships:	[Login Use Case] [Get Trip History Use Case] [Get Requirement Detail Use Case]											
Business Rules:												

- The location and route of driver's trip must be received repeatedly and automatically.

<Customer> Track Driver Trip Use Case Specification

2.3.3.16 <Customer> Get Bill Information



<Customer> Get Bill Information

USE CASE - SCB_UC_22									
Use Case No.	SCB_UC_22	Use Case Version	1.0						
Use Case Name	Get Bill Information								
Author	TrungTH								
Date	13/2/2019	Priority	Normal						
Actor:	<ul style="list-style-type: none"> - Customer 								
Summary:	<ul style="list-style-type: none"> - Allow customers to get trip's bill information. 								
Goal:	<ul style="list-style-type: none"> - Customer could get trip's bill information. 								
Triggers:	<ul style="list-style-type: none"> - Customer sends getting trip's bill information command. - Or after finishing trip. 								
Preconditions:	<ul style="list-style-type: none"> - Customer must login first. - Customer must be in viewing trip history screen screen - There is a contract. - The trip must be in "Finished" status. 								
Post conditions:	<ul style="list-style-type: none"> - Success: Customer gets trip's bill information successfully. - Fail: System shows error messages. 								
Main Success Scenario:	<table border="1"> <thead> <tr> <th>Step</th><th>Actor Action</th><th>System Response</th></tr> </thead> <tbody> <tr> <td>1</td><td>Customer goes to getting history of all trips screen (if not in tracking driver screen)</td><td> <p>Getting history of all trips screen with following information:</p> <ul style="list-style-type: none"> - Trip's date - Trip's status </td></tr> </tbody> </table>			Step	Actor Action	System Response	1	Customer goes to getting history of all trips screen (if not in tracking driver screen)	<p>Getting history of all trips screen with following information:</p> <ul style="list-style-type: none"> - Trip's date - Trip's status
Step	Actor Action	System Response							
1	Customer goes to getting history of all trips screen (if not in tracking driver screen)	<p>Getting history of all trips screen with following information:</p> <ul style="list-style-type: none"> - Trip's date - Trip's status 							

		<ul style="list-style-type: none"> - Driver's avatar - Driver's name - Children's avatar <p>Children's name</p>
2	Customer goes to getting trip's bill information screen	<p>Getting trip's bill information screen with following information:</p> <ul style="list-style-type: none"> - Children's information - Trip's detail information: pick-up/drop-off addresses, timeline - Total charge fee. <p>And a Confirm Button [Alternative 1] [Exception 1]</p>

Alternative Scenario:

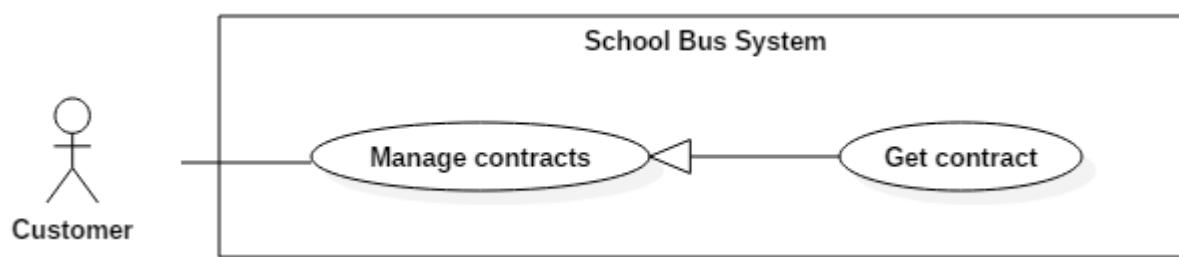
No	Actor Action	System Response
1	Customer confirms bill	Back to the viewing history of all trips screen

Exceptions:

No	Actor Action	System Response
1		System shows message the "Please check your connection!" when the internet is lost.

Relationships: [Login Use Case] [Get Trip History Use Case]**Business Rules:**

- The order detail must have list of kids and the corresponding price.
- Total price for the trip must be calculated.

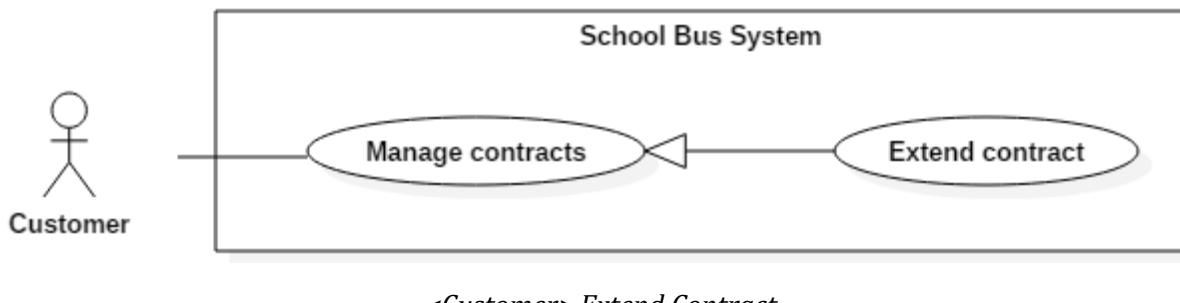
*<Customer> Get Bill Information Use Case Specification***2.3.3.17 <Customer> Get Contract***<Customer> Get Contract*

USE CASE – SCB_UC_23			
Use Case No.	SCB_UC_23	Use Case Version	1.0
Use Case Name	Get Contract		

Author	TrungTH				
Date	13/2/2019	Priority	Normal		
Actor:					
- Customer					
Summary:					
- Allow customers to get contract's detail.					
Goal:					
- Customer could get contract's detail.					
Triggers:					
- Contract is shown with requirement detail.					
Preconditions:					
- Customer must login first.					
- Customer must be in viewing requirement detail screen.					
Post conditions:					
- Success: Customer gets trip's bill information successfully.					
- Fail: System shows error messages.					
Main Success Scenario:					
Step	Actor Action	System Response			
1	Customer goes to getting requirement detail screen	Contract is on the bottom of getting requirement detail screen with following information: - Children's information - Trip's detail information: pick-up/drop-off addresses, timeline - Total charge fee. [Exception 1]			
Alternative Scenario:					
No	Actor Action	System Response			
Exceptions:					
No	Actor Action	System Response			
1		System shows message the "Please check your connection!" when the internet is lost.			
Relationships: [Login Use Case] [Get Requirement Detail Use Case]					
Business Rules:					
- If there's no active contract or pending request, a 'Find now' button is display for customer choosing matching driver's services.					

<Customer> Get Contract Use Case Specification

2.3.3.18 <Customer> Extend Contract



<Customer> Extend Contract

USE CASE - SCB_UC_24			
Use Case No.	SCB_UC_24	Use Case Version	1.0
Use Case Name	Extend Contract		
Author	TrungTH		
Date	13/2/2019	Priority	Normal
Actor:			
- Customer			
Summary:			
- Allow customers to extend transporting contract.			
Goal:			
- Customer could extend transporting contract.			
Triggers:			
- Contract sends extending transporting contract command.			
Preconditions:			
- Customer must login first.			
- Customer must be in viewing requirement detail screen.			
Post conditions:			
- Success: Customer extends transporting contract successfully.			
- Fail: System shows error messages.			
Main Success Scenario:			
Step	Actor Action	System Response	
1	Customer goes to getting requirement detail screen	Contract is on the bottom of getting trip's bill information screen with following information: - Children's information - Trip's detail information: pick-up/drop-off addresses, timeline - Total charge fee.	
2	Customer goes to extending transporting contract screen	Extending transporting contract screen with following fields: - Extended Date (Date Picker)	
3	Customer chooses extending date		

4	Customer sends extending transporting contract command	Customer extends the transporting contract period [Exception 1] [Alternate 1]
---	--	--

Alternative Scenario:

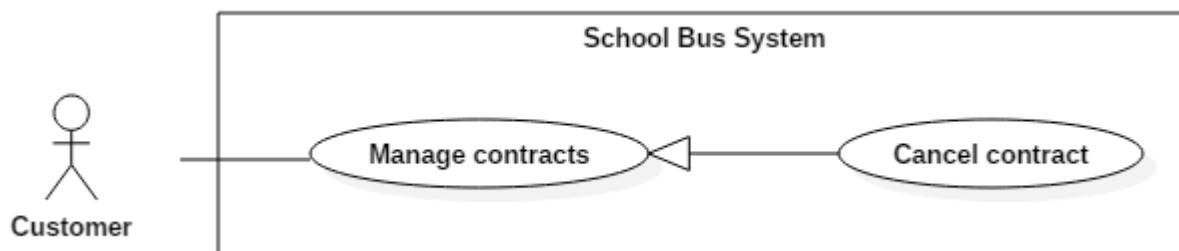
No	Actor Action	System Response
1	Out of capacity	The driver service is out of available capacity at extending time.

Exceptions:

No	Actor Action	System Response
1		System shows message the "Please check your connection!" when the internet is lost.

Relationships: [Login Use Case] [Get Requirement Detail Use Case]**Business Rules:**

- Extending date must be suitable with driver's service's available capacity.
- Customer would be redirected to getting requirement detail screen.
- System should push notification to driver to let he/she know about the request.

*<Customer> Extend Contract Use Case Specification***2.3.3.19 <Customer> Cancel Contract***<Customer> Cancel Contract*

USE CASE - SCB_UC_25			
Use Case No.	SCB_UC_25	Use Case Version	1.0
Use Case Name	Cancel Contract		
Author	TrungTH		
Date	13/2/2019	Priority	Normal
Actor:	<ul style="list-style-type: none"> - Customer 		
Summary:	<ul style="list-style-type: none"> - Allow customers to cancel transporting contract. 		
Goal:	<ul style="list-style-type: none"> - Customer could cancel transporting contract. 		
Triggers:	<ul style="list-style-type: none"> - Contract sends canceling transporting contract command. 		

Preconditions:

- Customer must login first.
- Customer must be in viewing requirement detail screen.

Post conditions:

- Success: Customer cancels transporting contract successfully.
- Fail: System shows error messages.

Main Success Scenario:

Step	Actor Action	System Response
1	Customer goes to getting requirement detail screen	Contract is on the bottom of getting trip's bill information screen with following information: <ul style="list-style-type: none"> - Children's information - Trip's detail information: pick-up/drop-off addresses, timeline - Total charge fee.
2	Customer goes to canceling transporting contract screen	Extending transporting contract screen with following fields: <ul style="list-style-type: none"> - Description (Text Area)
3	Customer edits description	
4	Customer sends canceling transporting contract command	Customer cancels the transporting contract. [Alternate 1] [Exception 1]

Alternative Scenario:

No	Actor Action	System Response
1	Customer inputs wrong format	System shows validating message: "Description must not be blank"

Exceptions:

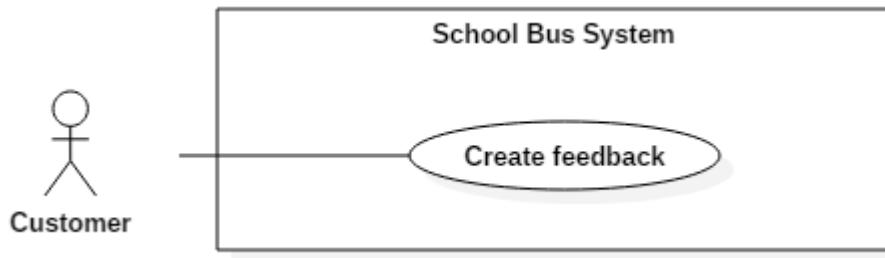
No	Actor Action	System Response
1		System shows message the "Please check your connection!" when the internet is lost.

Relationships: [Login Use Case] [Get Requirement Detail Use Case]**Business Rules:**

- There must be at least a contract of current requirement.
- If requests for service are still being in waiting for driver's confirmation, the cancelling contract link becomes cancelling request link.
- Customer would be redirected to getting requirement detail screen.
- System should push notification to driver.

<Customer> Cancel Contract Use Case Specification

2.3.3.20 <Customer> Create Feedback



<Customer> Create Feedback

USE CASE – SCB_UC_26			
Use Case No.	SCB_UC_26	Use Case Version	1.0
Use Case Name	Create Feedback		
Author	TrungTH		
Date	13/2/2019	Priority	Normal
Actor:	<ul style="list-style-type: none"> - Customer 		
Summary:	<ul style="list-style-type: none"> - Allow customers to create feedback to driver's service. 		
Goal:	<ul style="list-style-type: none"> - Customer could create feedback to driver's service. 		
Triggers:	<ul style="list-style-type: none"> - Contract sends creating feedback to driver's service command. 		
Preconditions:	<ul style="list-style-type: none"> - Customer must login first. - Customer must be in viewing requirement detail screen. 		
Post conditions:	<ul style="list-style-type: none"> - Success: Customer creates feedback to driver's service successfully. - Fail: System shows error messages. 		
Main Success Scenario:			
Step	Actor Action	System Response	
1	Customer goes to getting requirement detail screen	Contract is on the bottom of getting trip's bill information screen with following information: <ul style="list-style-type: none"> - Children's information - Trip's detail information: pick-up/drop-off addresses, timeline - Total charge fee. 	
2	Customer goes to creating feedback to driver's service screen	Extending transporting contract screen with following fields: <ul style="list-style-type: none"> - Description (Text Area) - Star Rating (Star Rating) 	
3	Customer edits feedback		

4	Customer sends creating feedback to driver's service command	Customer creates feedback to driver's service. [Exception 1]
---	--	---

Alternative Scenario:

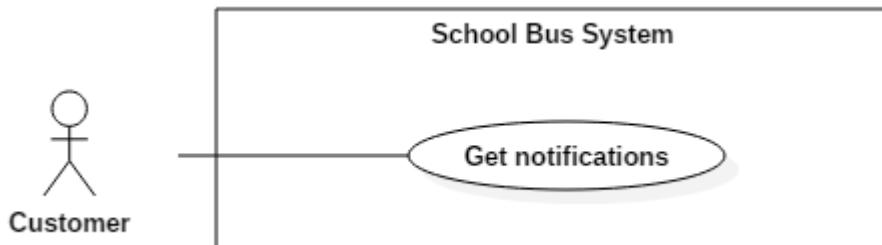
No	Actor Action	System Response
----	--------------	-----------------

Exceptions:

No	Actor Action	System Response
1		System shows message the "Please check your connection!" when the internet is lost.

Relationships: [Login Use Case] [Get Requirement Detail Use Case]**Business Rules:**

- An active contract in requirement is required for doing feedback.
- Customer can feedback for one service once only. The next feedback will modify the previous one.

*<Customer> Create Feedback Use Case Specification***2.3.3.21 <Customer> Get Notifications***<Customer> Get Notifications*

USE CASE - SCB_UC_27			
Use Case No.	SCB_UC_27	Use Case Version	1.0
Use Case Name	Get Notifications		
Author	TrungTH		
Date	13/2/2019	Priority	Normal
Actor:			
- Customer			
Summary:			
- Allow customer to get notifications			
Goal:			
- Customer could get notifications			
Triggers:			
- Customer sends get notifications command.			
- Server push notifications.			
Preconditions:			

- Customer must login.
- There are push notifications from server.

Post conditions:

- Success: Customer gets notifications successfully.
- Fail: System shows error messages.

Main Success Scenario:

Step	Actor Action	System Response
1	Customer goes to getting notifications screen.	<p>List of notifications [Exception 1]</p> <p>Type of notifications:</p> <ul style="list-style-type: none"> - Found matching driver. - Driver arrived. - Children picked up. - Children arrived at school. - Children is back. - Driver accepts requirement. - Driver rejects requirement. - Driver accepts extending. - Driver rejects extending. - Driver cancels trip. - Driver cancels contract.
2	Customer receives push notifications	

Alternative Scenario:

No	Actor Action	System Response

Exceptions:

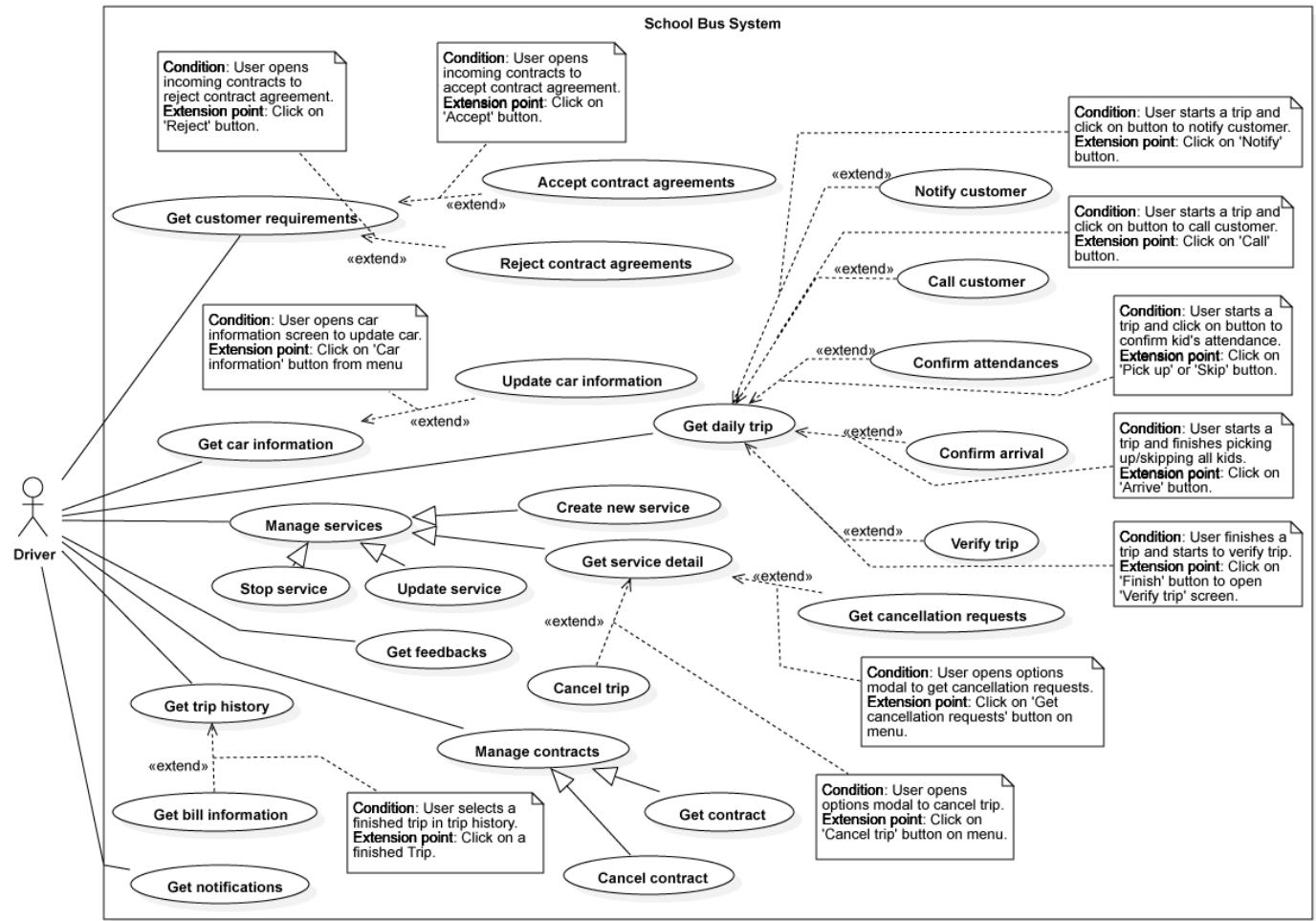
No	Actor Action	System Response
1		System shows message the "Please check your connection!" when the internet is lost.

Relationships: N/A**Business Rules:**

- Vital notifications should be saved and displayed.
- A notification should redirect to an action when the user chooses.
- Notifications should be ordered descending by time.

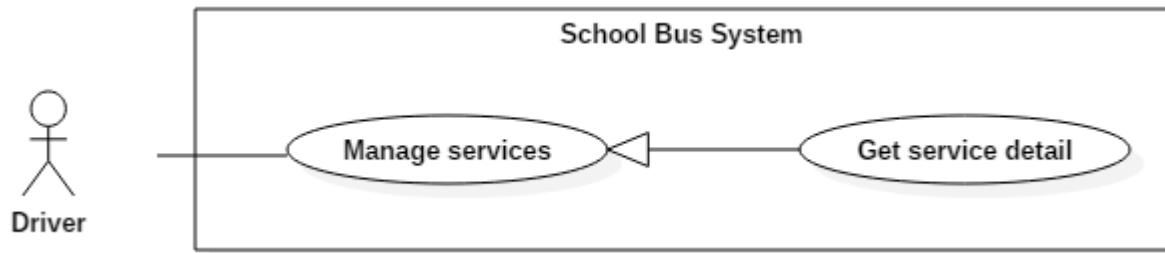
<Customer> Get Notifications Use Case Specification

2.3.4 <Driver> Overview Use Case



Driver Overview Use Case

2.3.4.1 <Driver> Get Service Detail



USE CASE - SCB_UC_28			
Use Case No.	SCB_UC_28	Use Case Version	1.0
Use Case Name	Get service detail		
Author	GiangTV		
Date	17/02/2019	Priority	High
Actor:	<ul style="list-style-type: none"> - Driver 		
Summary:	<ul style="list-style-type: none"> - This use case allows driver to view registered service. 		

Goal:

- Review registered services.

Triggers:

- Driver selects a service from 'Services' screen.

Preconditions:

- Driver must be logged in.

Post conditions:

- Success: show 'Service Detail' screen.
- Fail: Show error message.

Main Success Scenario:

Step	Actor Action	System Response
1	Driver goes to 'Services' scene	<p>Open 'Services' screen with following properties:</p> <ul style="list-style-type: none"> - School - Going trip – Returning trip - Days of week - Start Trip Button
2	Driver touches a service in 'Services' screen.	<p>Open 'Service Detail' screen with following properties:</p> <ul style="list-style-type: none"> - Start Address - School - Days of week - Going trip – Returning trip - Contracts: active, pending, history <p>[Exception 1]</p>

Alternative Scenario:**Exceptions:**

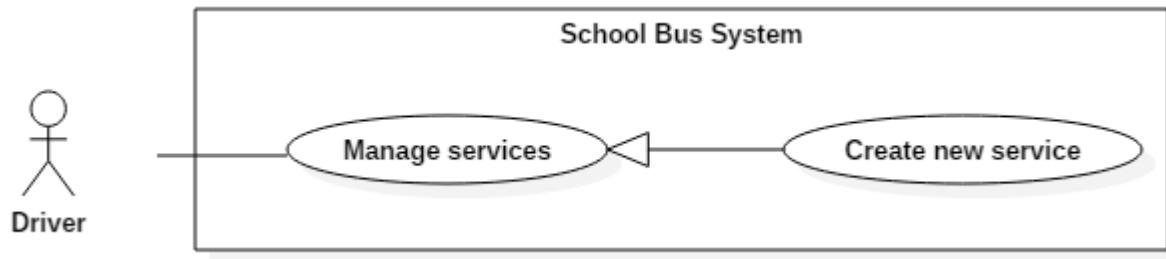
No	Actor Action	System Response
1	Internet connection error	Show error message with error code.

Relationships: [Login Use Case]**Business rules:**

- Contracts must be shown inside service detail.
- There are 3 type of contracts: active, pending and history.

<Driver> Get Service Detail Specification

2.3.4.2 <Driver> Create new service



USE CASE - SCB_UC_29

Use Case No.	SCB_UC_29	Use Case Version	1.0
Use Case Name	Create new service		
Author	Giangtv		
Date	17/02/2019	Priority	High

Actor:

- Driver

Summary:

- This use case allows driver to create new shuttling service.

Goal:

- Create new shuttling service.

Triggers:

- Driver touches 'Create' button on 'Service Registration' screen.

Preconditions:

- Driver must login in.

Post conditions:

- Success: Show "Service Detail" screen with created service information.
- Fail: Show error message.

Main Success Scenario:

Step	Actor Action	System Response
1	Driver opens 'Service Registration' screen.	<p>Application shows 'Service Registration' screen with following properties:</p> <ul style="list-style-type: none"> - Days of week - Start Time - Arrival Time - Return Time - Start Address - School - Class - Available Capacity
2	Driver inputs necessary information.	
3	Driver touches 'Create' button.	<p>Show success notification.</p> <p>[Alternate 1] [Alternate 2] [Exception 1]</p>

Alternative Scenario:

No	Actor Action	System Response
1	Some fields are blank	All fields must not be blank
2	Time constraints are not right	Return time must be greater than arrival time, arrival time must be greater than start time

Exceptions:

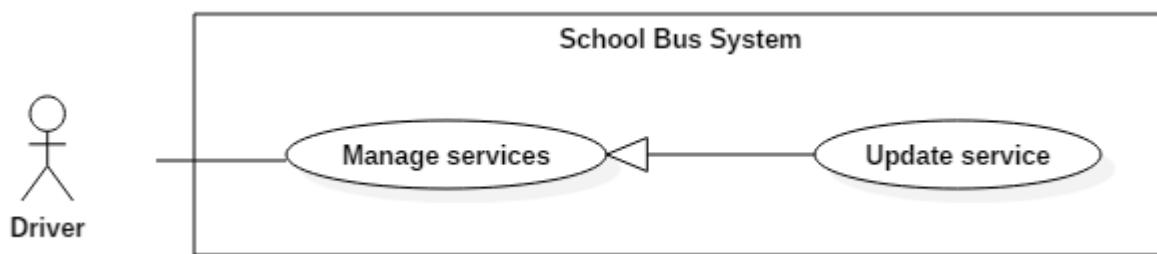
No	Actor Action	System Response
1	No internet connection	Show error message with error code.

Relationships: [Login Use Case]**Business rules:**

- Registered time must not overlap any others existing service.
- Address coordinates must be provided.
- Start address should be suggested as driver's address

<Driver> Create new service Specification

2.3.4.3 <Driver> Update service

**USE CASE – SCB_UC_30**

Use Case No.	SCB_UC_30	Use Case Version	1.0
Use Case Name	Update Service		
Author	Giangtv		
Date	17/02/2019	Priority	High

Actor:

- Driver

Summary:

- Change service information.

Goal:

- Update current services.

Triggers:

- Driver opens 'Update Service' screen.

Preconditions:

- Driver must be logged in.
- Selected service must have no ACTIVE or PENDING contract.

Post conditions:

- Success: Show 'Service Detail' screen.
- Fail: Show error message.

Main Success Scenario:

Step	Actor Action	System Response
1	Driver goes to 'Services' scene	<p>Open 'Services' screen with following properties:</p> <ul style="list-style-type: none"> - School - Going trip – Returning trip - Days of week - Start Trip Button
2	Driver touches a service in 'Services' screen.	<p>Open 'Service Detail' screen with following properties:</p> <ul style="list-style-type: none"> - Start Address - School - Days of week - Going trip – Returning trip - Contracts: active, pending, history <p>[Exception 1]</p>
3	Driver open options	<p>Options are shown:</p> <ul style="list-style-type: none"> - Update service - Delete service
4	Driver opens 'Update Service' screen	<p>Application show 'Update Service' screen with following properties:</p> <ul style="list-style-type: none"> - Days of week - Start Time - Arrival Time - Return Time - Start Address - School - Class <p>Available Capacity</p>
5	Driver changes desired information	
6	Driver touches 'Update' button	<p>Display success notification.</p> <p>[Alternate 1] [Alternate 2] [Exception 1]</p>

Alternative Scenario:

No	Actor Action	System Response
1	Some fields are blank	All fields must not be blank

2	Time constraints are not right	Return time must be greater than arrival time, arrival time must be greater than start time
---	--------------------------------	---

Exceptions:

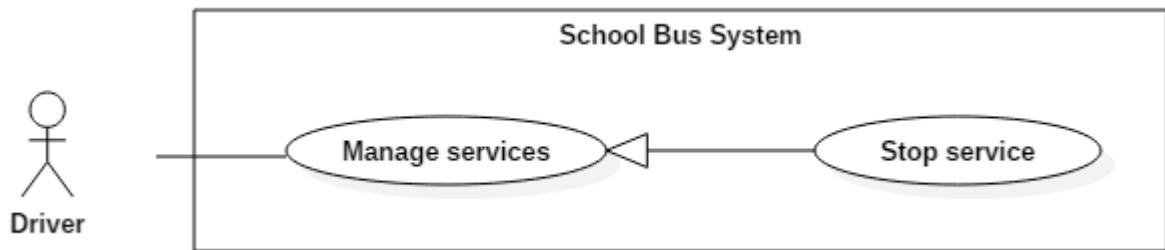
No	Actor Action	System Response
1	No internet connection	Show error message with error code.

Relationships: [Login Use Case]**Business rules:**

- A service can be updated only when it has no active or waiting-for-response contract.
- Changed time must not overlap any others existing service.

<Driver> Update service Specification

2.3.4.4 <Driver> Stop service



USE CASE - SCB_UC_31			
Use Case No.	SCB_UC_31	Use Case Version	1.0
Use Case Name	Stop service		
Author	Giangtv		
Date	17/02/2019	Priority	High
Actor:			
- Driver			
Summary:			
- Request to stop a current active service with no customer.			
Goal:			
- Stop a shuttling service.			
Triggers:			
- Driver clicks 'Stop' button on 'Service Detail' screen.			
Preconditions:			
- Driver must be logged in.			
- Selected service must have no contract.			
Post conditions:			
- Success: Show success message.			
- Fail: Show error message.			
Main Success Scenario:			
Step	Actor Action	System Response	

1	Driver goes to 'Services' scene	Open 'Services' screen with following properties: <ul style="list-style-type: none"> - School - Going trip – Returning trip - Days of week - Start Trip Button
2	Driver touches a service in 'Services' screen.	Open 'Service Detail' screen with following properties: <ul style="list-style-type: none"> - Start Address - School - Days of week - Going trip – Returning trip - Contracts: active, pending, history <p>[Exception 1]</p>
3	Driver presses option button	Options are shown: <ul style="list-style-type: none"> - Update service - Delete service
4	Driver presses 'Delete'	Confirming window is shown for deleting
5	Driver chooses 'OK'	System notifies that service was deleted successfully. <p>[Exception 1]</p>

Alternative Scenario:

No	Actor Action	System Response
1	Driver choose 'No' in deleting alert	Back to service's detail

Exceptions:

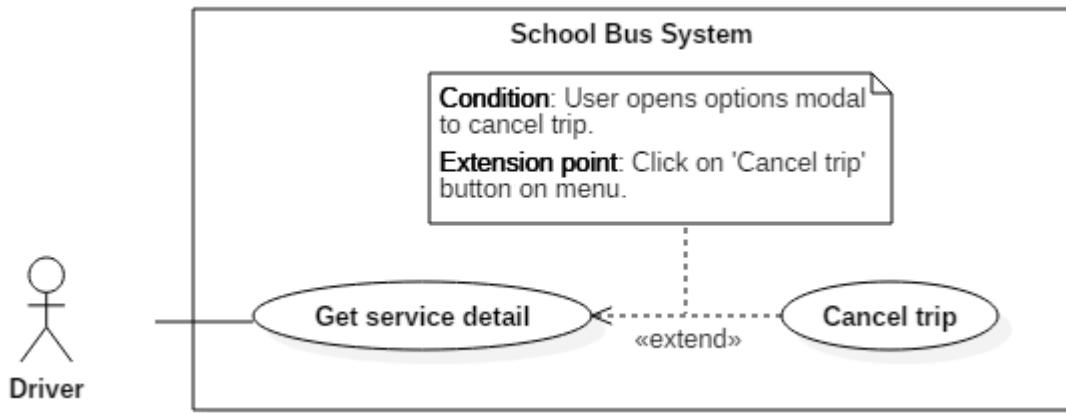
No	Actor Action	System Response
1	Internet connection error.	Show error message with error code.

Relationships: [Login Use Case] [Get Service Detail Use Case]**Business rules:**

- Driver can only stop the service that have no active contract or no contract agreement requests.

<Driver> Stop service Specification

2.3.4.5 <Driver> Cancel trip



USE CASE - SCB_UC_32			
Use Case No.	SCB_UC_32	Use Case Version	1.0
Use Case Name	Cancel trip		
Author	Giangtv		
Date	17/02/2019	Priority	High
Actor:	<ul style="list-style-type: none"> - Driver 		
Summary:	<ul style="list-style-type: none"> - Cancel service for a specific day and give reason. 		
Goal:	<ul style="list-style-type: none"> - Register a day-off. 		
Triggers:	<ul style="list-style-type: none"> - Driver touches 'Cancel' on 'Service Detail' screen. 		
Preconditions:	<ul style="list-style-type: none"> - Driver must be logged in 		
Post conditions:	<ul style="list-style-type: none"> - Success: Show success notification. - Fail: Show error message. 		
Main Success Scenario:			
Step	Actor Action	System Response	
1	Driver goes to 'Services' scene	Open 'Services' screen with following properties: <ul style="list-style-type: none"> - School - Going trip – Returning trip - Days of week - Start Trip Button 	
2	Driver touches a service in 'Services' screen.	Open 'Service Detail' screen with following properties: <ul style="list-style-type: none"> - Start Address - School - Days of week - Going trip – Returning trip 	

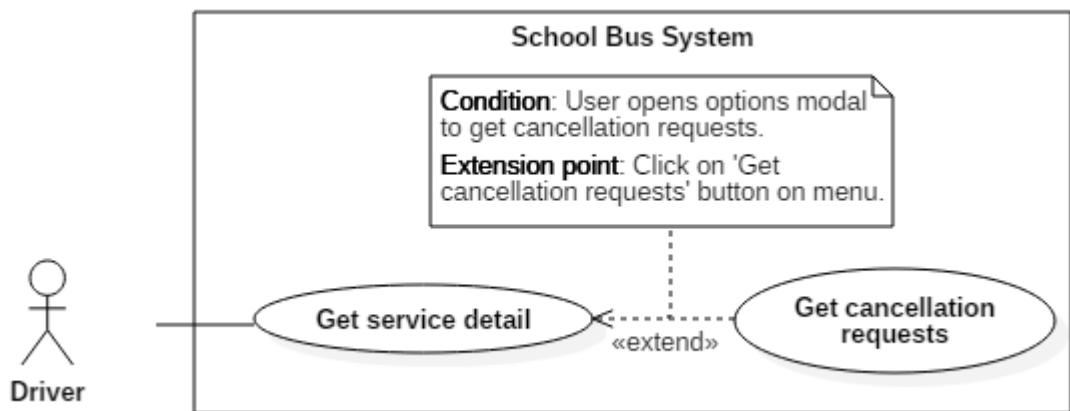
		<ul style="list-style-type: none"> - Contracts: active, pending, history <p>[Exception 1]</p>
3	Driver presses option button	<p>Options:</p> <ul style="list-style-type: none"> - Cancel trip - View cancelled trips
4	Driver goes to 'Cancel Trip' screen	<p>'Cancel Contract' screen with following properties:</p> <ul style="list-style-type: none"> - OffDate - Description
5	Driver presses 'Cancel' button	<p>System notifies that request for cancelling trip was created successfully!</p> <p>[Exception 1]</p>

Alternative Scenario:**Exceptions:**

No	Actor Action	System Response
1	Internet connection error	Show error message with error code.

Relationships: [Login Use Case] [Get Service Detail Use Case]**Business rules:**

- Only accept the day included in driver service.
- System should send notification to all customers that are affected in that day.

*<Driver> Cancel trip Specification***2.3.4.6 <Driver> Get cancellation requests**

USE CASE - SCB_UC_33			
Use Case No.	SCB_UC_33	Use Case Version	1.0
Use Case Name	Get cancellation requests		
Author	Giangtv		
Date	17/02/2019	Priority	High
Actor:	<ul style="list-style-type: none"> - Driver 		
Summary:	<ul style="list-style-type: none"> - View all Driver + Customers cancellation requests for a Service. 		
Goal:	<ul style="list-style-type: none"> - Let driver know the days he and his customer's children were off. 		
Triggers:	<ul style="list-style-type: none"> - Driver opens 'Service Detail' Screen. 		
Preconditions:	<ul style="list-style-type: none"> - Driver must be logged in. 		
Post conditions:	<ul style="list-style-type: none"> - Success: show days off. - Fail: Show error message. 		
Main Success Scenario:			
Step	Actor Action	System Response	
1	Driver goes to 'Services' scene	Open 'Services' screen with following properties: <ul style="list-style-type: none"> - School - Going trip – Returning trip - Days of week - Start Trip Button 	
2	Driver touches a service in 'Services' screen.	Open 'Service Detail' screen with following properties: <ul style="list-style-type: none"> - Start Address - School - Days of week - Going trip – Returning trip - Contracts: active, pending, history <p>[Exception 1]</p>	
3	Driver presses option button	Options: <ul style="list-style-type: none"> - Cancel trip - View cancelled trips 	
4	Driver goes to 'View Cancelled Trip' screen	'View Cancelled Trip' screen with following properties: <ul style="list-style-type: none"> - Customer's cancelled trips: children, off-day, content 	

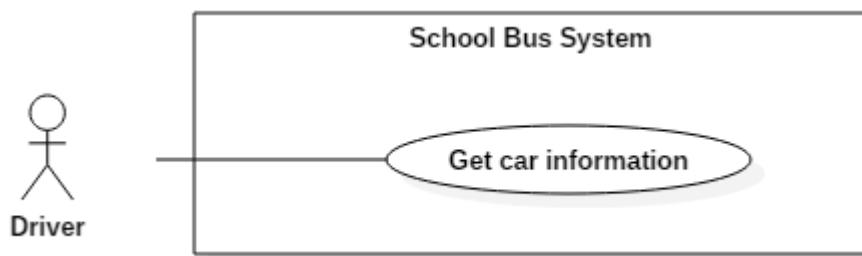
		- Driver's cancelled trips: off-day, content
--	--	--

Alternative Scenario:**Exceptions:**

No	Actor Action	System Response
1	Internet connection error.	Show error message with error code.

Relationships: [Login Use Case] [Get Service Detail Use Case]**Business rules:**

- Cancellation requests should be ordered descending by time.

*<Driver> Get cancellation requests Specification***2.3.4.7 <Driver> Get car information**

USE CASE - SCB_UC_34			
Use Case No.	SCB_UC_34	Use Case Version	1.0
Use Case Name	Get car information		
Author	Giangtv	Priority	High
Date	17/02/2019	Priority	High
Actor:	<ul style="list-style-type: none"> - Driver 		
Summary:	<ul style="list-style-type: none"> - Let driver review his car 		
Goal:	<ul style="list-style-type: none"> - Show driver car information. 		
Triggers:	<ul style="list-style-type: none"> - Driver touches 'My Car' entry in left menu. 		
Preconditions:	<ul style="list-style-type: none"> - Driver must be logged in. 		
Post conditions:	<ul style="list-style-type: none"> - Success: Show "Car information" screen. - Fail: Show error message. 		
Main Success Scenario:			
Step	Actor Action	System Response	

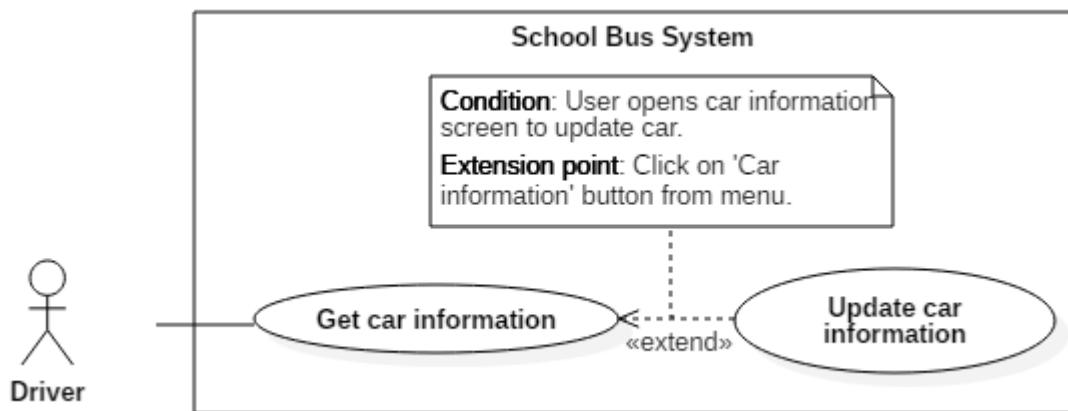
1	Driver touches 'My Car' entry in the left menu	Show "My Car" screen with following properties: <ul style="list-style-type: none"> - Plate Number - Brand - Model - Color - Capacity
---	--	--

Alternative Scenario:**Exceptions:**

No	Actor Action	System Response
1	Internet connection error.	Show error message with error code.

Relationships: [Login Use Case]**Business rules:**

<Driver> Get car information Specification

2.3.4.8 <Driver> Update car information

USE CASE - SCB_UC_35			
Use Case No.	SCB_UC_35	Use Case Version	1.0
Use Case Name	Update car information		
Author	Giangtv		
Date	17/02/2019	Priority	High
Actor:	<ul style="list-style-type: none"> - Driver 		
Summary:	<ul style="list-style-type: none"> - Let driver change his car information. 		
Goal:	<ul style="list-style-type: none"> - Change car information. 		
Triggers:	<ul style="list-style-type: none"> - Driver touches 'Update' button in 'Car information' screen. 		
Preconditions:	<ul style="list-style-type: none"> - Driver must be logged in. 		

Post conditions:

- Success: Show car information after updated.
- Fail: Show error message.

Main Success Scenario:

Step	Actor Action	System Response
1	Driver go to 'View car' screen	Show "My Car" screen with following properties: <ul style="list-style-type: none"> - Plate Number - Brand - Model - Color - Capacity
2	Driver changes the desired information	
3	Driver touches 'Update' button	Show 'Success notification' then show "Car information" screen [Exception 1]

Alternative Scenario:**Exceptions:**

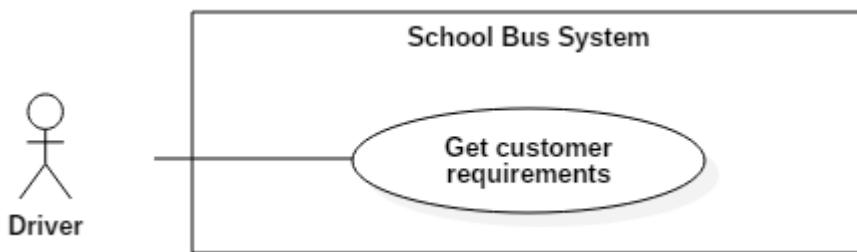
No	Actor Action	System Response
1	Internet connection error.	Show error message with error code.

Relationships: [Login Use Case]**Business rules:**

- Car information cannot duplicate with other drivers.
- Driver can only update car information once after registering.

<Driver> Update car information Specification

2.3.4.9 <Driver> Get customer requirements

**USE CASE - SCB_UC_36**

Use Case No.	SCB_UC_36	Use Case Version	1.0
Use Case Name	Get customer requirements		
Author	Giangtv		

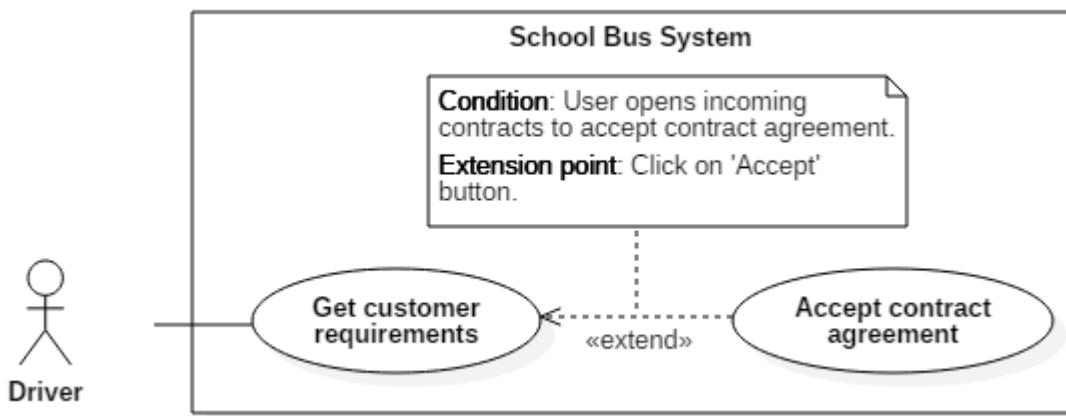
Date	17/02/2019	Priority	High
Actor:			
- Driver			
Summary:			
- View who are registered to use driver service and give driver UI option to reject or accept that requirement.			
Goal:			
- View registered requirements.			
Triggers:			
- Driver opens 'Service Detail' screen and choose Incoming Contracts.			
Preconditions:			
- Driver must be logged in.			
Post conditions:			
- Success: Show Pending Contract Agreements.			
- Fail: Show error message.			
Main Success Scenario:			
Step	Actor Action	System Response	
1	Driver touches "Service Detail" entry in "Services" screen	Show detail of service with following properties: <ul style="list-style-type: none"> - Start Address - School - Days of week - Going trip – Returning trip - Contracts: active, pending, history 	
2	Driver touches Incoming tab inside service detail	Show list of pending contract agreements for driver to response, with following properties: <ul style="list-style-type: none"> - Customer name - Customer avatar - Children List - Going trip – Returning trip - Two buttons for accepting or rejecting <p>[Exception 1] [Alternate 1]</p>	
Alternative Scenario:			
No	Actor Action	System Response	
1	No pending requirements	Show no pending requirements	
Exceptions:			
No	Actor Action	System Response	
1	Internet connection error.	Show error message with error code.	
Relationships: [Login Use Case] [Choose Service Use Case] [Extend Contract Use Case]			

Business rules:

- The requirements must be in some pending contracts waiting for driver's confirmation

<Driver> Get customer requirements Specification

2.3.4.10 <Driver> Accept contract agreement

**USE CASE - SCB_UC_37**

Use Case No.	SCB_UC_37	Use Case Version	1.0
Use Case Name	Accept contract agreement		
Author	Giangtv		
Date	17/02/2019	Priority	High

Actor:

- Driver

Summary:

- Accept an agreement in the request list.

Goal:

- Accept agreement and create contract.

Triggers:

- Driver touches 'Accept' button in contract agreement.

Preconditions:

- Driver must be logged in.

Post conditions:

- Success: Show success message.
- Fail: Show error message.

Main Success Scenario:

Step	Actor Action	System Response
1	Driver opens list of contract agreements	<p>Show list of pending contract agreements for driver to response, with following properties:</p> <ul style="list-style-type: none"> - Customer name - Customer avatar - Children List - Going trip – Returning trip

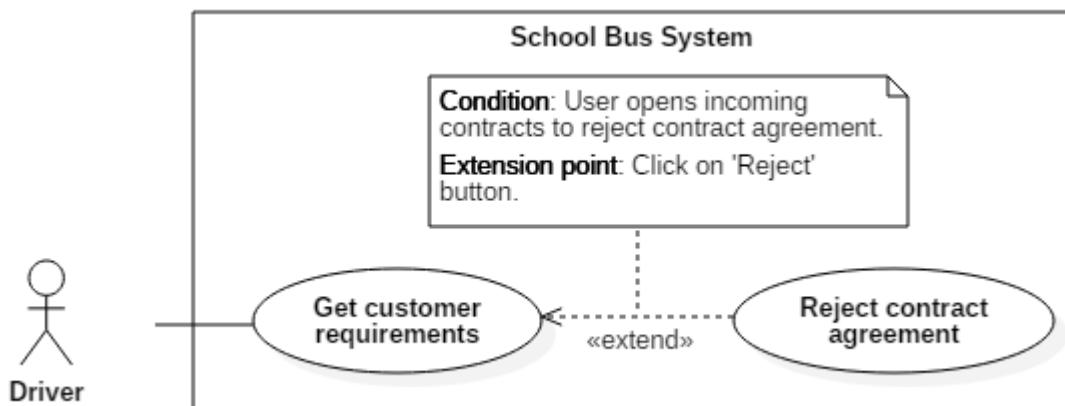
		<ul style="list-style-type: none"> - Two buttons for accepting or rejecting
2	Driver touches 'Accept' button	Update data and display message. [Exception 1]

Alternative Scenario:**Exceptions:**

No	Actor Action	System Response
1	Internet connection error.	Show error message with error code.

Relationships: [Login Use Case] [Choose Service Use Case] [Extend Contract Use Case]**Business rules:**

- Driver capacity is checked before processing Accept.
- If there are any conflicts between the days of week of driver service and customer requirement, the missing days of week in customer will not be removed out of contract.
- Notification is sent to customer when driver accepted.

*<Driver> Accept contract agreement Specification***2.3.4.11 <Driver> Reject contract agreement**

USE CASE – SCB_UC_38			
Use Case No.	SCB_UC_38	Use Case Version	1.0
Use Case Name	Reject contract agreement		
Author	Giangtv		
Date	17/02/2019	Priority	High
Actor:	<ul style="list-style-type: none"> - Driver 		
Summary:	<ul style="list-style-type: none"> - Reject an agreement in the request list. 		
Goal:	<ul style="list-style-type: none"> - Reject agreement and create contract. 		

Triggers:

- Driver touches 'Reject' button in contract agreement.

Preconditions:

- Driver must be logged in.

Post conditions:

- Success: Show message
- Fail: Show error message.

Main Success Scenario:

Step	Actor Action	System Response
1	Driver opens list of contract agreements	Show list of pending contract agreements for driver to response, with following properties: <ul style="list-style-type: none">- Customer name- Customer avatar- Children List- Going trip – Returning trip- Two buttons for accepting or rejecting
2	Driver touches 'Reject' button	Update data and display message. [Exception 1]

Alternative Scenario:**Exceptions:**

No	Actor Action	System Response
1	Internet connection error.	Show error message with error code.

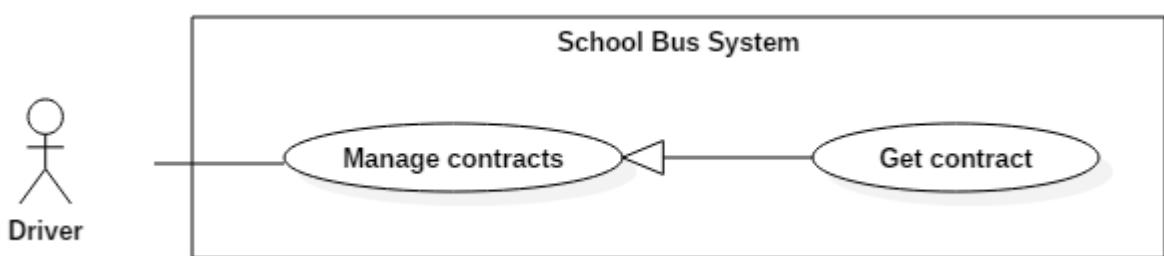
Relationships: [Login Use Case] [Choose Service Use Case] [Extend Contract Use Case]

Business rules:

- Notification is sent to customer when driver rejected.

<Driver> Reject contract agreement Specification

2.3.4.12 <Driver> Get contract

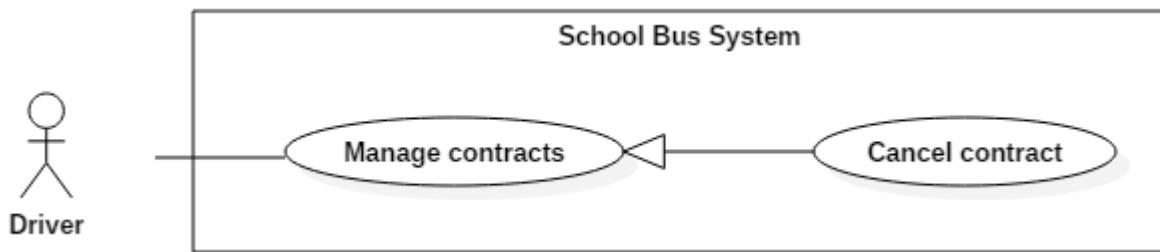
**USE CASE - SCB_UC_39**

Use Case No.	SCB_UC_39	Use Case Version	1.0
<hr/>			

Use Case Name	Get contract		
Author	Giangtv		
Date	17/02/2019	Priority	High
Actor:			
<ul style="list-style-type: none"> - Driver 			
Summary:			
<ul style="list-style-type: none"> - Let driver overview his contracts that he has with customer. He can review history contract from there. 			
Goal:			
<ul style="list-style-type: none"> - View driver contracts. 			
Triggers:			
<ul style="list-style-type: none"> - Driver views a service. 			
Preconditions:			
<ul style="list-style-type: none"> - Driver must be logged in. 			
Post conditions:			
<ul style="list-style-type: none"> - Success: Show “Service Detail” screen. - Fail: Show error message. 			
Main Success Scenario:			
Step	Actor Action	System Response	
1	Driver goes to ‘Services’ scene	Open ‘Services’ screen with following properties: <ul style="list-style-type: none"> - School - Going trip – Returning trip - Days of week - Start Trip Button 	
2	Driver touches a service in ‘Services’ screen.	Open ‘Service Detail’ screen with following properties: <ul style="list-style-type: none"> - Start Address - School - Days of week - Going trip – Returning trip - Contracts: active, pending, history <p>[Exception 1]</p>	
3	Contracts are shown within service’s detail screen	Contracts are shown with following properties: <ul style="list-style-type: none"> - Customer’s name - Customer’s avatar - Customer’s phone - Children - Unit price - Total price - Pickup address - Pickup time 	

Alternative Scenario:**Exceptions:**

No	Actor Action	System Response
1	Internet connection error.	Show error message with error code.

Relationships: [Login Use Case] [Get Service Detail Use Case]**Business rules:***<Driver> Get contract Specification***2.3.4.13 <Driver> Cancel contract****USE CASE - SCB_UC_40**

Use Case No.	SCB_UC_40	Use Case Version	1.0
Use Case Name	Cancel contract		
Author	Giangtv		
Date	17/02/2019	Priority	High

Actor:

- Driver

Summary:

- Driver can cancel contract but he must give a reason.

Goal:

- Cancel a contract that the driver is not able to support the customer.

Triggers:

- Driver views a service.

Preconditions:

- Driver must be logged in.

Post conditions:

- Success: Show message
- Fail: Show error message.

Main Success Scenario:

Step	Actor Action	System Response
1	Driver goes to 'Services' scene	<p>Open 'Services' screen with following properties:</p> <ul style="list-style-type: none"> - School - Going trip – Returning trip - Days of week - Start Trip Button

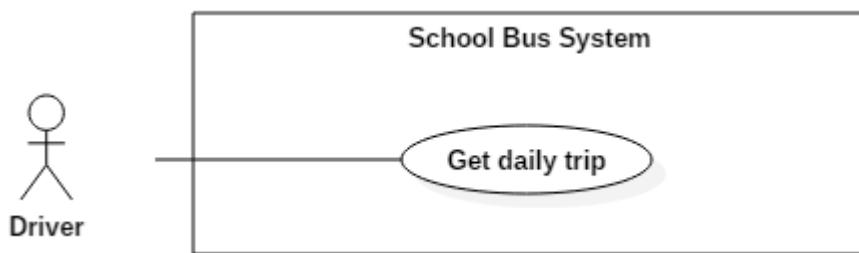
2	Driver touches a service in 'Services' screen.	Open 'Service Detail' screen with following properties: <ul style="list-style-type: none"> - Start Address - School - Days of week - Going trip – Returning trip - Contracts: active, pending, history [Exception 1]
3	Driver presses an active contract long enough to show options	Cancel contract option
4	Driver goes to 'Cancel Contract' screen	'Cancel Contract' screen with following properties: <ul style="list-style-type: none"> - Description
5	Driver presses 'Cancel' button	System notifies that contract was deleted successfully! [Exception 1]

Alternative Scenario:**Exceptions:**

No	Actor Action	System Response
1	Internet connection error.	Show error message with error code.

Relationships: [Login Use Case] [Get Service Detail Use Case]**Business rules:**

- Driver can cancel contract at any time as well as he/she has to give a justifiable reason.
- The contract must be in active status.
- System should send notification to customer

*<Driver> Cancel contract Specification***2.3.4.14 <Driver> Get daily trip**

USE CASE - SCB_UC_41			
Use Case No.	SCB_UC_41	Use Case Version	1.0

Use Case Name	Get daily trip		
Author	Giangtv		
Date	17/02/2019	Priority	High

Actor:

- Driver

Summary:

- Get Trip detail for upcoming trip.

Goal:

- Get Trip detail.

Triggers:

- Driver touches 'Start Trip' button.

Preconditions:

- Driver must be logged in.
- Selected trip must have suitable time.

Post conditions:

- Success: Open "Driving" screen.
- Fail: Show error message.

Main Success Scenario:

Step	Actor Action	System Response
1	Driver touches "Start Trip" button	<p>Open "Driving" screen with following properties:</p> <ul style="list-style-type: none"> - Map (with direction) - Picking-up List: Customer name, avatar, pickup address, children - Buttons: Notify, pickup, skip, call, arrive at school <p>[Exception 1] [Alternate 1]</p>

Alternative Scenario:

Step	Actor Action	System Response
1	Trip has no suitable time	Show message

Exceptions:

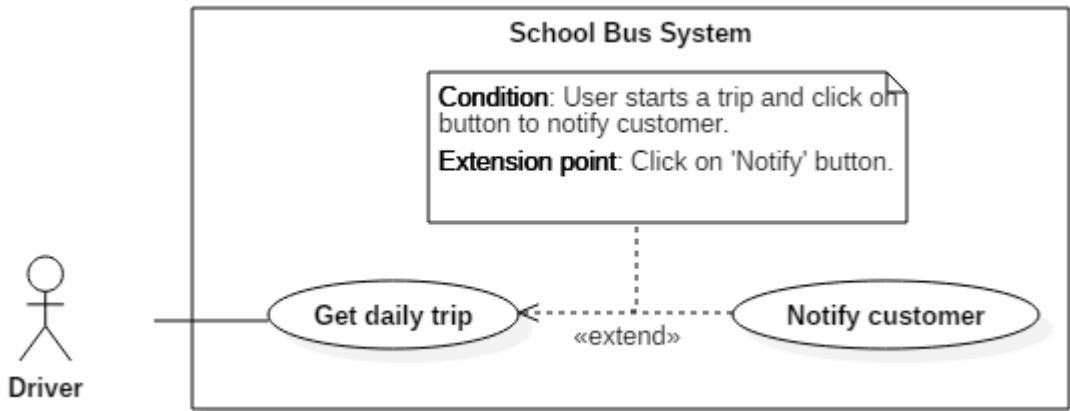
No	Actor Action	System Response
1	Internet connection error.	Show error message with error code.

Relationships: [Login Use Case] [Get Trip History Use Case]**Business rules:**

- Create new trip for driver if not existed.
- Return current trip if the trip is existed.
- Only get contract that have the matching day of week.
- Absent children are not shown in trip.
- Driver's location and route must be continuously sent to server after getting trip.

<Driver> Get daily trip Specification

2.3.4.15 <Driver> Notify customer



USE CASE – SCB_UC_42

Use Case No.	SCB_UC_42	Use Case Version	1.0
Use Case Name	Notify customer		
Author	Giangtv		
Date	17/02/2019	Priority	High

Actor:

- Driver

Summary:

- Send a push notification to customer when he is arriving to pick up their child.

Goal:

- Notify customer when the driver comes.

Triggers:

- Driver starts a trip.

Preconditions:

- Driver must be logged in.
- Driver must be in a Driving trip.

Post conditions:

- Success: none.
- Fail: Show error message.

Main Success Scenario:

Step	Actor Action	System Response
1	Driver touches 'Start Trip' button	Show 'Driving' screen
2	Driver touches 'Notify' button in map or picking up list.	[Exception 1]

Alternative Scenario:

Exceptions:

No	Actor Action	System Response
1	Internet connection error.	Show error message with error code.

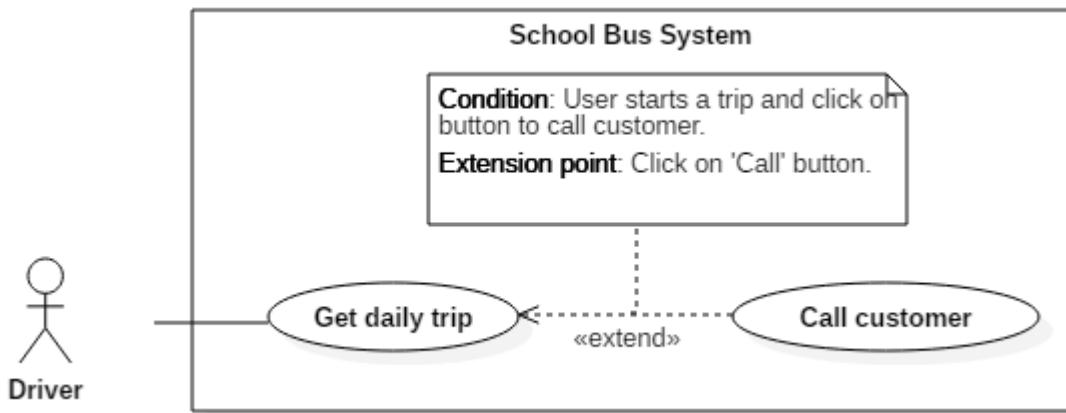
Relationships: [Login Use Case] [Get Daily Trip]

Business rules:

- All of the customer devices must receive notification.

<Driver> Notify customer Specification

2.3.4.16 <Driver> Call customer



USE CASE - SCB_UC_43

Use Case No.	SCB_UC_43	Use Case Version	1.0
Use Case Name	Call customer		
Author	Giangtv		
Date	17/02/2019	Priority	High

Actor:

- Driver

Summary:

- Change to OS dialer to with customer phone number inserted.

Goal:

- An easier way to make a call to customer.

Triggers:

- Driver touches the dialer-shaped icon in the 'Driving' screen.

Preconditions:

- Driver must be logged in.
- Driver must be in a Driving trip.

Post conditions:

- Success: Open OS dialer.
- Fail: Show error message.

Main Success Scenario:

Step	Actor Action	System Response
1	Driver touches 'Start Trip' button	Show 'Driving' screen
2	Driver touches 'Call' button in map or picking up list.	Open OS dialer with inserted customer number.

Alternative Scenario:

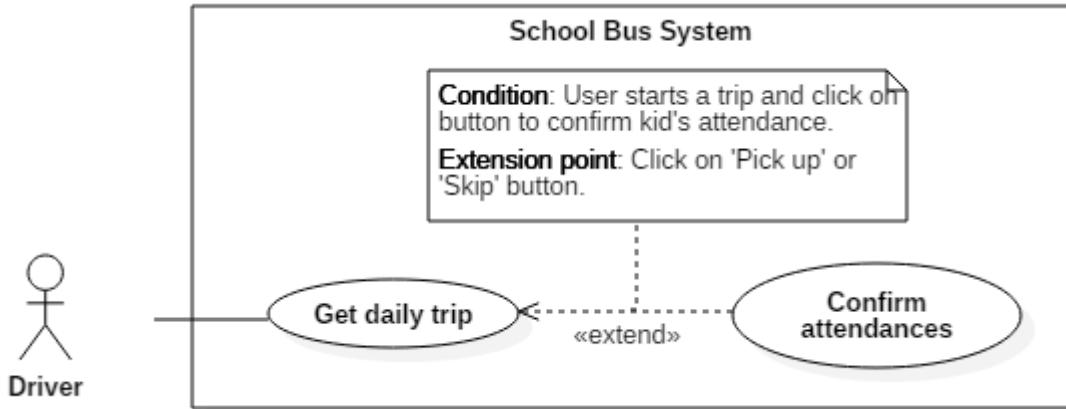
Exceptions:

Relationships: [Login Use Case] [Get Daily Trip]

Business rules:

<Driver> Call customer Specification

2.3.4.17 <Driver> Confirm attendances



USE CASE - SCB_UC_44

Use Case No.	SCB_UC_44	Use Case Version	1.0
Use Case Name	Confirm attendances		
Author	Giangtv		
Date	17/02/2019	Priority	High

Actor:

- Driver

Summary:

- Update the status of that kid inside the daily trip. Notify customer to let them know.

Goal:

- Record the data for the daily trip.

Triggers:

- Driver touch 'Picked up' or 'Skip' button.

Preconditions:

- Driver must be logged in.
- Driver must be in a Driving trip.

Post conditions:

- Success: none.
- Fail: Show error message.

Main Success Scenario:

Step	Actor Action	System Response
1	Driver touches 'Start Trip' button	Show 'Driving' screen.
2	Driver touches 'check-mark' or 'dismiss' icon in map or picking up list.	[Exception 1]

Alternative Scenario:

Exceptions:

No	Actor Action	System Response
----	--------------	-----------------

1	Internet connection error.	Show error message with error code.
---	----------------------------	-------------------------------------

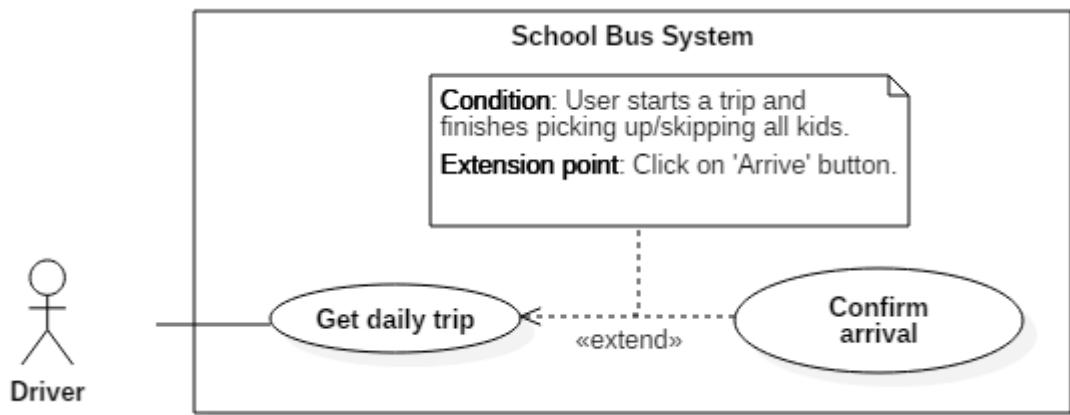
Relationships: [Login Use Case] [Get Daily Trip]

Business rules:

- Driver can only confirm attendances for kids who are not absent that day.
- Notification is sent to customer when driver picked-up or skip his/her child.

<Driver> Confirm attendances Specification

2.3.4.18 <Driver> Confirm arrival



USE CASE - SCB_UC_45									
Use Case No.	SCB_UC_45	Use Case Version	1.0						
Use Case Name	Confirm arrival								
Author	Giangtv								
Date	17/02/2019	Priority	High						
Actor:	<ul style="list-style-type: none"> - Driver 								
Summary:	<ul style="list-style-type: none"> - Update trip status and notify customers. 								
Goal:	<ul style="list-style-type: none"> - Record the data for the daily trip. 								
Triggers:	<ul style="list-style-type: none"> - Driver touch 'Arrive' button. 								
Preconditions:	<ul style="list-style-type: none"> - Driver must be logged in. - Driver must be in a Driving trip. 								
Post conditions:	<ul style="list-style-type: none"> - Success: Open "Trip Completion" screen. - Fail: Show error message. 								
Main Success Scenario:	<table border="1"> <thead> <tr> <th>Step</th><th>Actor Action</th><th>System Response</th></tr> </thead> <tbody> <tr> <td>1</td><td>Driver touches 'Arrive' button</td><td>Open "Trip Completion" screen on success.</td></tr> </tbody> </table>			Step	Actor Action	System Response	1	Driver touches 'Arrive' button	Open "Trip Completion" screen on success.
Step	Actor Action	System Response							
1	Driver touches 'Arrive' button	Open "Trip Completion" screen on success.							

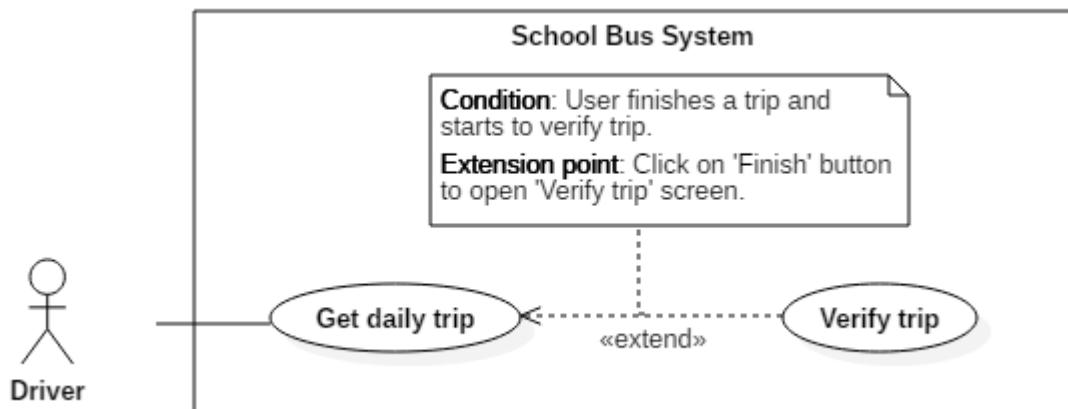
		[Exception 1]
--	--	---------------

Alternative Scenario:**Exceptions:**

No	Actor Action	System Response
1	Internet connection error.	Show error message with error code.

Relationships: [Login Use Case] [Get Daily Trip]**Business rules:**

- Send notification to customer with notification information according to trip information.
- Customer whose children are all skipped or absent will not receive the notification.

*<Driver> Confirm arrival Specification***2.3.4.19 <Driver> Verify trip**

USE CASE – SCB_UC_46			
Use Case No.	SCB_UC_46	Use Case Version	1.0
Use Case Name	Verify Trip		
Author	Giangtv		
Date	17/02/2019	Priority	High
Actor:	<ul style="list-style-type: none"> - Driver 		
Summary:	<ul style="list-style-type: none"> - Driver has to take a picture of the children and message to announce parents after completing trip. 		
Goal:	<ul style="list-style-type: none"> - Make sure the driver has driven the kids to school successfully. 		
Triggers:	<ul style="list-style-type: none"> - Driver touches 'Submit' button. 		
Preconditions:	<ul style="list-style-type: none"> - Driver must be logged in. - Driver must arrive after Driving trip. 		

Post conditions:

- Success: Open “Bill” screen.
- Fail: Show error message.

Main Success Scenario:

Step	Actor Action	System Response
1	Driver goes to the ‘Verification’ screen	‘Verification’ screen with following properties: <ul style="list-style-type: none"> - Image for evidence - Content
2	Driver touches “Send” button	Open bill screen [Exception 1] [Alternate 1]

Alternative Scenario:

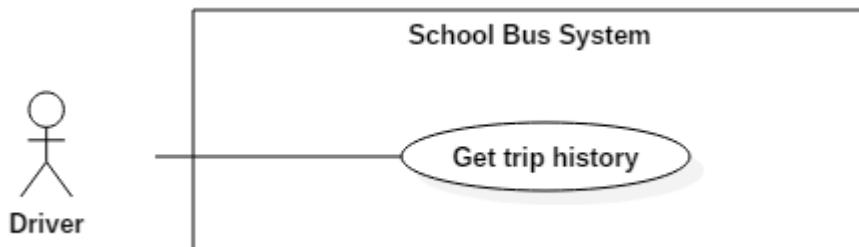
No	Actor Action	System Response
1	No empty fields	All fields must not be blank

Exceptions:

No	Actor Action	System Response
1	Internet connection error.	Show error message with error code.

Relationships: [Login Use Case] [Get Daily Trip]**Business rules:**

- Confirm image can only be taken using camera. No any others method accepted.
- Customer whose children are all skipped or absent will not receive the notification.

*<Driver> Verify trip Specification***2.3.4.20 <Driver> Get trip history****USE CASE – SCB_UC_47**

Use Case No.	SCB_UC_47	Use Case Version	1.0
Use Case Name	Get trip history		
Author	Giangtv		
Date	17/02/2019	Priority	High
Actor:			
- Driver			

Summary:

- This use case allows driver to view all trips he have driven.

Goal:

- Tracking driver work.

Triggers:

- Driver touches the 'History' entry in the Home Screen.

Preconditions:

- Driver must be logged in.

Post conditions:

- Success: Display trip history screen.
- Fail: Show error message.

Main Success Scenario:

Step	Actor Action	System Response
1	Driver touches "History" entry in the Home Screen	Show history screen with following properties: <ul style="list-style-type: none"> - School - Day - Time - Status [Exception 1]

Exceptions:

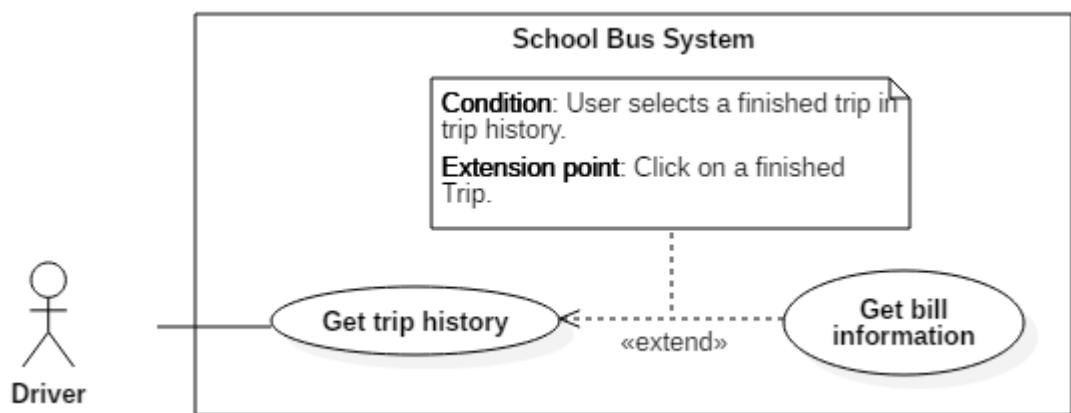
No	Actor Action	System Response
1	API cannot connect to server.	Show error message with error code.

Relationships: [Login Use Case]**Business rules:**

- All finished or cancelled or even current trips must be listed.
- Trip must be ordered descending by time.

<Driver> Get trip history Specification

2.3.4.21 <Driver> Get bill information



USE CASE - SCB_UC_48			
Use Case No.	SCB_UC_48	Use Case Version	1.0
Use Case Name	Get bill information		
Author	Giangtv		
Date	17/02/2019	Priority	High
Actor:	<ul style="list-style-type: none"> - Driver 		
Summary:	<ul style="list-style-type: none"> - This use case allows driver to view bill of a finished trip. 		
Goal:	<ul style="list-style-type: none"> - View the bill of a specific trip. 		
Triggers:	<ul style="list-style-type: none"> - Driver touches Trip record on 'History screen' or after completing trip. 		
Preconditions:	<ul style="list-style-type: none"> - Driver must be logged in. - Trip must be finished. 		
Post conditions:	<ul style="list-style-type: none"> - Success: Show bill information - Fail: Show error message. 		
Main Success Scenario:			
Step	Actor Action	System Response	
1	Driver opens 'History' screen.	<p>Show history screen with following properties:</p> <ul style="list-style-type: none"> - School - Day - Time - Status 	
2	Driver touches a trip in the list	<p>Show bill information with following properties:</p> <ul style="list-style-type: none"> - Day - Start Time - Trip id - Driver name - Driver avatar - Driver phone - Car's plate number - Children - Pickup address - Pickup time - Arrival time - Unit Price - Total Price <p>[Exception 1]</p>	
Exceptions:			

No	Actor Action	System Response
1	API cannot connect to server.	Show error message with error code.

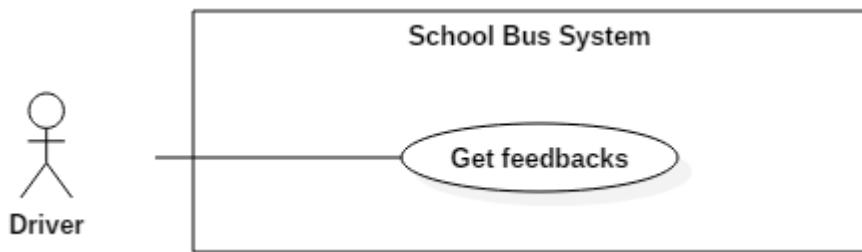
Relationships: [Login Use Case] [Get Trip History Use Case]

Business rules:

- The order detail must have list of kids and the corresponding price.
- Total price for the trip must be calculated.

<Driver> Get bill information Specification

2.3.4.22 <Driver> Get feedbacks



USE CASE - SCB_UC_49

Use Case No.	SCB_UC_49	Use Case Version	1.0
Use Case Name	Get feedback		
Author	Giangtv		
Date	17/02/2019	Priority	High

Actor:

- Driver

Summary:

- Let driver view all of his feedback.

Goal:

- View customer feedbacks.

Triggers:

- Driver touches ‘Feedbacks’ entry in side bar menu.

Preconditions:

- Driver must be logged in.

Post conditions:

- Success: Open “Feedback” screen.
- Fail: Show error message.

Main Success Scenario:

Step	Actor Action	System Response
1	Driver touches ‘Review Feedbacks’ entry in side bar menu.	<p>Open “Review Feedbacks” screen with following properties:</p> <ul style="list-style-type: none"> - Driver’s name - Driver’s avatar - Driver’s average score - Feedback’s score - Feedback’s content

		- Feedbacks' time [Exception 1]
--	--	---------------------------------------

Alternative Scenario:**Exceptions:**

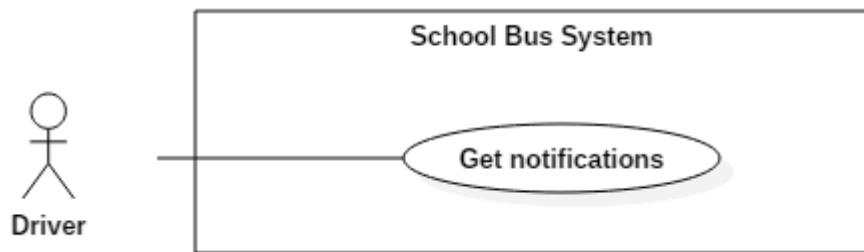
No	Actor Action	System Response
1	Internet connection error.	Show error message with error code.

Relationships: [Login Use Case]**Business rules:**

- Feedback can be score only. Comment is optional.
- Average score should be provided.

<Driver> Get feedbacks Specification

2.3.4.23 <Driver> Get notifications



USE CASE - SCB_UC_50						
Use Case No.	SCB_UC_50	Use Case Version	1.0			
Use Case Name	Get notifications					
Author	Giangtv					
Date	17/02/2019	Priority	High			
Actor:	<ul style="list-style-type: none"> - Driver 					
Summary:	<ul style="list-style-type: none"> - This use case allows driver to view his all notifications. 					
Goal:	<ul style="list-style-type: none"> - List all available notifications that driver has. 					
Triggers:	<ul style="list-style-type: none"> - Driver touches the 'Notifications' icon in the top right navigation bar. 					
Preconditions:	<ul style="list-style-type: none"> - Driver must be logged in. 					
Post conditions:	<ul style="list-style-type: none"> - Success: Show screen list all notifications. - Fail: Show error message. 					
Main Success Scenario:	<table border="1"> <thead> <tr> <th>Step</th> <th>Actor Action</th> <th>System Response</th> </tr> </thead> </table>			Step	Actor Action	System Response
Step	Actor Action	System Response				

1	Driver touches the “Notifications” icon in the top right navigation bar.	<p>Show “Notifications” screen with following properties:</p> <ul style="list-style-type: none"> - Title - Content - Day - Time <p>[Exception 1]</p>
---	--	--

Exceptions:

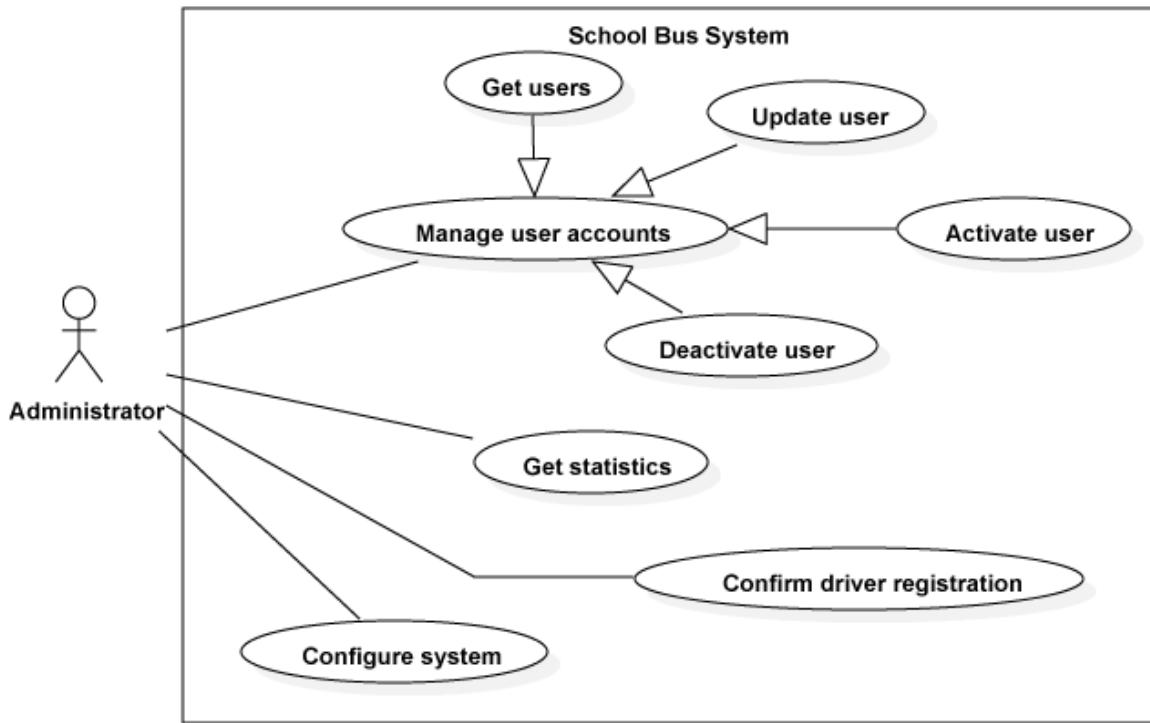
No	Actor Action	System Response
1	API cannot connect to server.	Show error message with error code.

Relationships: [Login Use Case]**Business rules:**

- Vital notifications should be saved and displayed.
- A notification should redirect to an action when the user chooses.
- Notifications should be ordered descending by time.

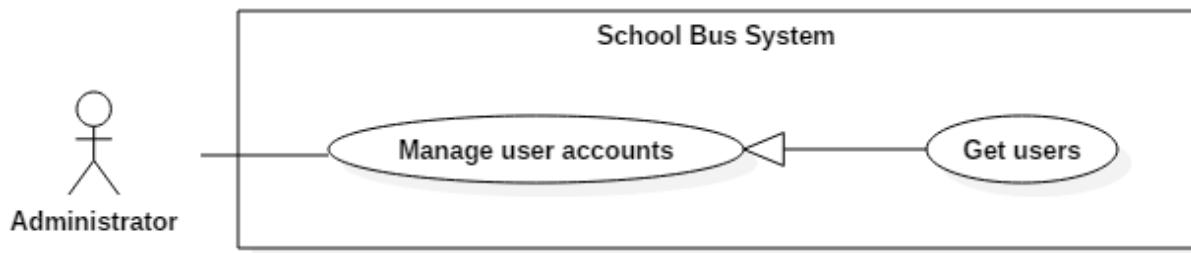
<Driver> Get notifications Specification

2.3.5 <Administrator> Overview Use Case



Administrator Overview Use Case

2.3.5.1 <Administrator> Get users



USE CASE - SCB_UC_51

Use Case No.	SCB_UC_51	Use Case Version	1.0
Use Case Name	Get users		
Author	VinhNT		
Date	17/02/2019	Priority	Normal

Actor:

- Administrator

Summary:

- View user account include driver and customer.

Goal:

- Overview user account with some statistic information.

Triggers:

- Administrator chooses an account from list in "User account" page.

Preconditions:

- Administrator must be logged in.

Post conditions:

- Success: Show user account table.
- Fail: Show error message.

Main Success Scenario:

Step	Actor Action	System Response
1	Administrator chooses an account from list in "User account" page.	Show "Account Detail" page.

Alternative Scenario:

Exceptions:

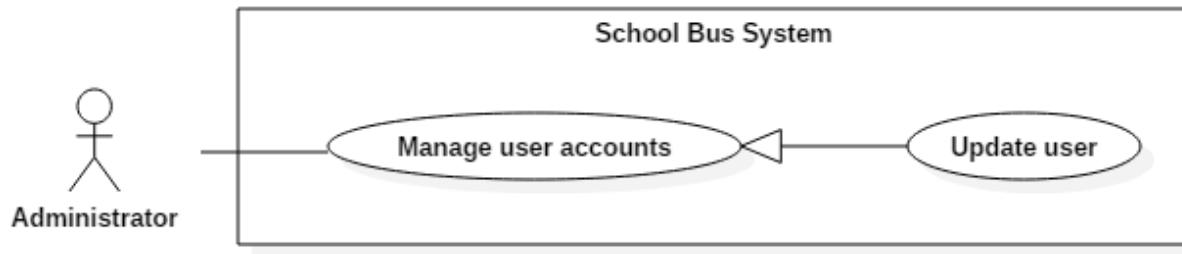
No	Actor Action	System Response
1	Internet connection error.	Show error message with error code.

Relationships:

Business rules:

<Administrator> Get users Specification

2.3.5.2 <Administrator> Update user



USE CASE - SCB_UC_52

Use Case No.	SCB_UC_52	Use Case Version	1.0
Use Case Name	Update user		
Author	VinhHNT		
Date	17/02/2019	Priority	Normal

Actor:

- Administrator

Summary:

- Change customer account as required.

Goal:

- Control user account.

Triggers:

- Administrator clicks “Update” button on “Update user account” page.

Preconditions:

- Administrator must be logged in.

Post conditions:

- Success: Show user account table.
- Fail: Show error message.

Main Success Scenario:

Step	Actor Action	System Response
1	Administrator clicks “Update” button on “Update user account” page.	Show “Account Detail” page

Alternative Scenario:

Exceptions:

No	Actor Action	System Response
1	Internet connection error.	Show error message with error code.

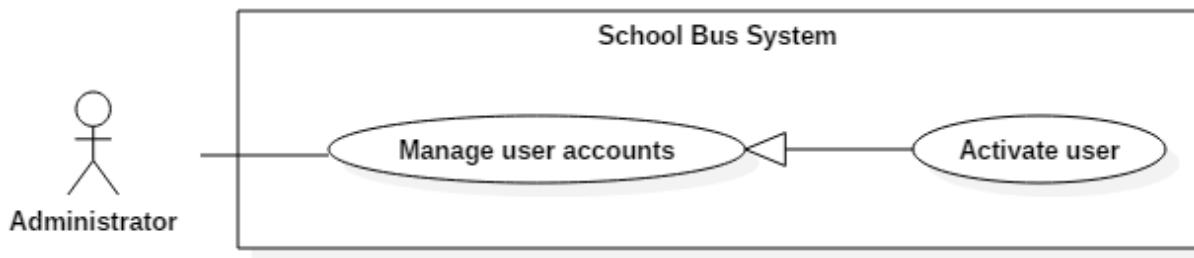
Relationships:

Business rules:

- Only some kind of user data that Administrator can change.

<Administrator> Update user Specification

2.3.5.3 <Administrator> Activate user



USE CASE - SCB_UC_53

Use Case No.	SCB_UC_53	Use Case Version	1.0
Use Case Name	Activate user		
Author	VinhNT		
Date	17/02/2019	Priority	Normal

Actor:

- Administrator

Summary:

- Change account status.

Goal:

- Control the user action.

Triggers:

- Administrator clicks “Activate” button in account list in “User account” page or in “Account detail” page.

Preconditions:

- Administrator must be logged in.

Post conditions:

- Success: Show user account table.
- Fail: Show error message.

Main Success Scenario:

Step	Actor Action	System Response
1	Administrator clicks “Activate” button in account list in “User account” page or in “Account detail” page	Show notification.

Alternative Scenario:

Exceptions:

No	Actor Action	System Response
1	Internet connection error.	Show error message with error code.

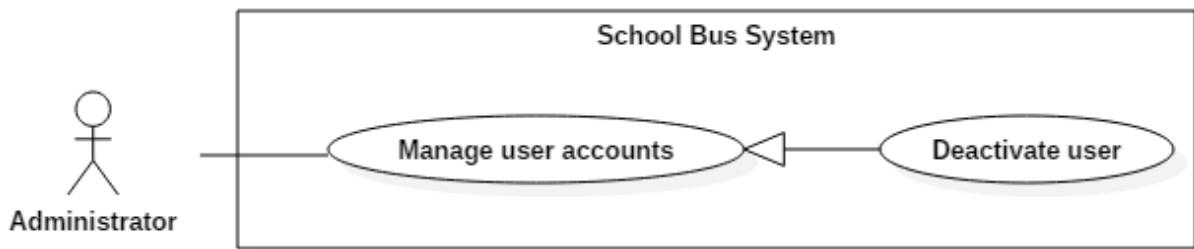
Relationships:

Business rules:

- For the new account:
 - New customer account is activated by default.
 - New driver account must be verified then being activated by administrator.
- Activate a deactivated account must log the reason.

<Administrator> Activate user Specification

2.3.5.4 <Administrator> Deactivate user



USE CASE - SCB_UC_54

Use Case No.	SCB_UC_54	Use Case Version	1.0
Use Case Name	Deactivate user		
Author	VinhNT		
Date	17/02/2019	Priority	Normal

Actor:

- Administrator

Summary:

- Change account status.

Goal:

- Control the user action.

Triggers:

- Administrator clicks “Deactivate” button in account list in “User account” page or in “Account detail” page.

Preconditions:

- Administrator must be logged in.

Post conditions:

- Success: Show user account table.
- Fail: Show error message.

Main Success Scenario:

Step	Actor Action	System Response
1	Administrator clicks “Deactivate” button in account list in “User account” page or in “Account detail” page	Show notification.

Alternative Scenario:

Exceptions:

No	Actor Action	System Response
1	Internet connection error.	Show error message with error code.

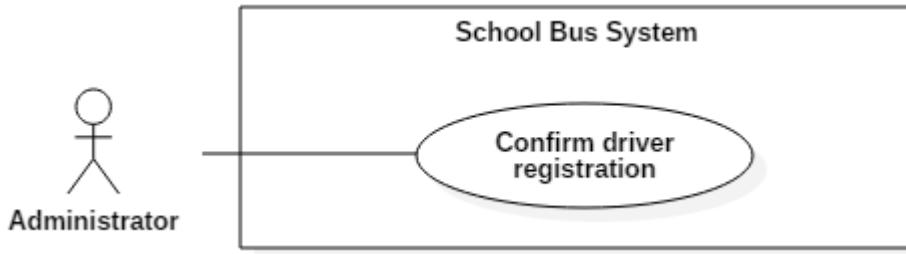
Relationships:

Business rules:

- Deactivate an account must provide at least one reason.
- Deactivate account cannot login to application.

<Administrator> Deactivate user Specification

2.3.5.5 <Administrator> Confirm driver registration



USE CASE - SCB_UC_55

Use Case No.	SCB_UC_55	Use Case Version	1.0
Use Case Name	Confirm driver registration		
Author	VinhNT		
Date	17/02/2019	Priority	Normal

Actor:

- Administrator

Summary:

- Check if driver information and car status.

Goal:

- Active driver account if he/she meet the requirements to become driver.

Triggers:

- Administrator clicks "Accept" / "Reject" button in account list in "Driver registration detail" page.

Preconditions:

- Administrator must be logged in.

Post conditions:

- Success: none.
- Fail: Show error message.

Main Success Scenario:

Step	Actor Action	System Response
1	Administrator clicks "Accept" / "Reject" button in account list in "Driver registration detail" page.	Show "Driver registration" page.

Alternative Scenario:

Exceptions:

No	Actor Action	System Response
1	Internet connection error.	Show error message with error code.

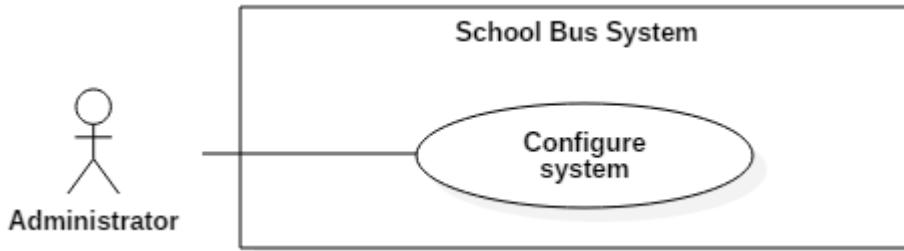
Relationships:

Business rules:

- Activate driver account when "Accept", deactivate when "Reject".
- Reject must input reason.
- Send notification to user when complete.

<Administrator> Confirm driver registration Specification

2.3.5.6 <Administrator> Configure system



USE CASE - SCB_UC_56					
Use Case No.	SCB_UC_56	Use Case Version	1.0		
Use Case Name	Configure System				
Author	VinhNT				
Date	17/02/2019	Priority	Normal		
Actor:	<ul style="list-style-type: none"> - Administrator 				
Summary:	<ul style="list-style-type: none"> - Change base price in server. 				
Goal:	<ul style="list-style-type: none"> - Change base price. 				
Triggers:	<ul style="list-style-type: none"> - Administrator clicks "Update" button beside each parameter to confirm the updating. 				
Preconditions:	<ul style="list-style-type: none"> - Administrator must be logged in. - There must be default value for parameters 				
Post conditions:	<ul style="list-style-type: none"> - Success: Parameters are updated successfully. - Fail: Show error message. 				
Main Success Scenario:					
Step	Actor Action	System Response			
1	Administrator goes to the parameter updating screen	List of system parameters like: <ul style="list-style-type: none"> - Matching time - Matching distance - Matching rate - Unit price - Starting time 			
2	Administrator clicks "Update" of Price section button in "System parameters" page.	Show notification page.			
Alternative Scenario:					
Exceptions:					
No	Actor Action	System Response			

1	Internet connection error.	Show error message with error code.
---	----------------------------	-------------------------------------

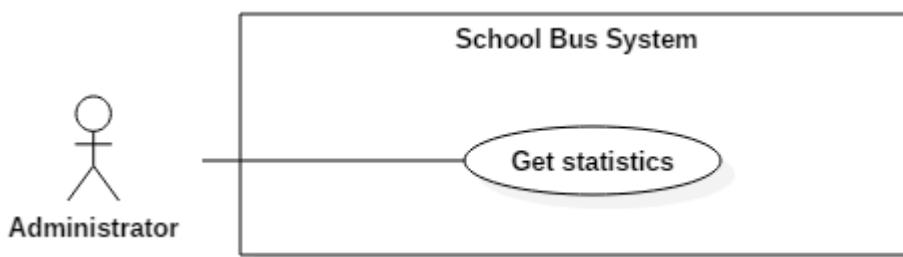
Relationships:

Business rules:

- After updating, all the functions depending on the system values must be stable without errors.

<Administrator> Configure system specification

2.3.5.7 <Administrator> Get statistics



USE CASE - SCB_UC_57

Use Case No.	SCB_UC_57	Use Case Version	1.0
Use Case Name	Get statistics		
Author	VinhNT		
Date	17/02/2019	Priority	Normal

Actor:

- Administrator

Summary:

- Overview about driver service and customer requirement.

Goal:

- Data statistic.

Triggers:

- Administrator clicks "Statistics" entry in left menu.

Preconditions:

- Administrator must be logged in.

Post conditions:

- Success: Update page.
- Fail: Show error message.

Main Success Scenario:

Step	Actor Action	System Response
1	Administrator clicks "Services Statistic" entry in left menu.	<p>Show "Services statistic" page with following statistics:</p> <ul style="list-style-type: none"> - Services and requirements statistics - Money statistics. - All trips statistics.

		- Trip cancellation statistics.
--	--	---------------------------------

Alternative Scenario:**Exceptions:**

No	Actor Action	System Response
1	Internet connection error.	Show error message with error code.

Relationships:**Business rules:**

- Analysis services density.
- Classify service quality.

<Administrator> Get statistics Specification

3. Software System Attribute

3.1 Usability

Texts used in mobile application must be configured to display in multi-language (Vietnamese and English):

- Font style: Roboto Regular, Roboto Medium.
- Font-size: 12-20dp.
- Color: theme color – green (#77b15a), gray, white, black, orange, red, light green...
- Background: Main background color is white, in home page, there are 3 image backgrounds based on different time of day.

The application is built for everyone. Therefore, it must be easy for every people to install on their own devices and use.

3.2 Reliability

The information is stored in database, which is backed up periodically.

Third party APIs are from trusted provider (Google and Expo).

3.3 Availability

The system must be available for 24/7 so that users can have access to system through mobile application at any time.

3.4 Security

Each user logging into system is provided and authorized by an access token.

User in each role has restricted functions within the scope of that role.

3.5 Maintainability

The system is divided into separated modules for the ease of maintenance.

Applying some design patterns (Immutability, Constant Data Manager, ...), which makes source code becomes clean and easy to update.

3.6 Portability

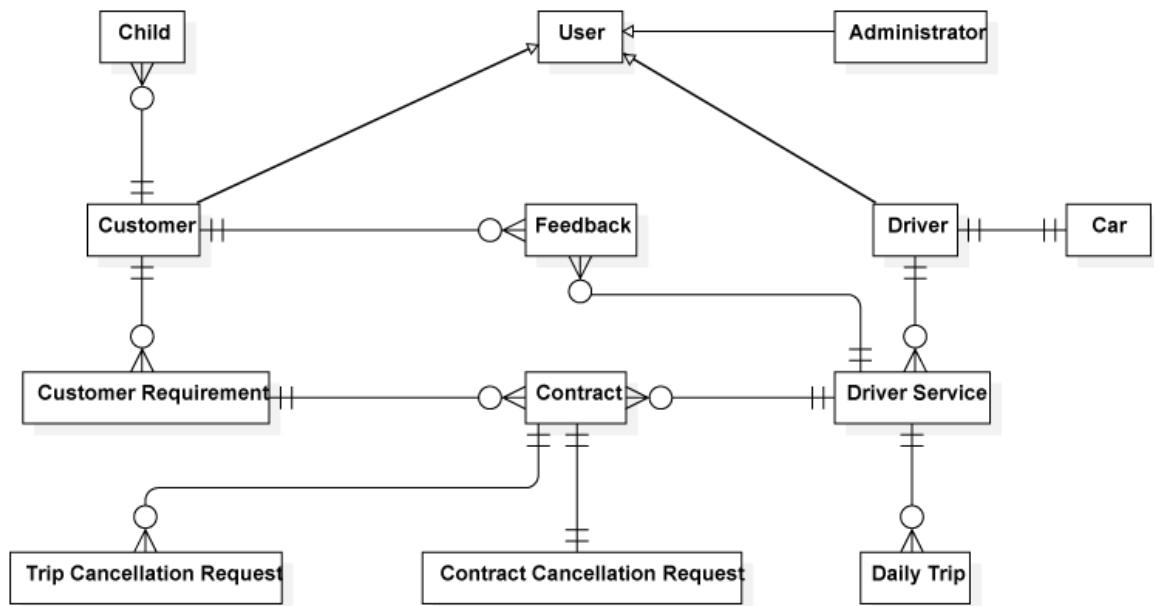
The application is available for all smartphone running on Android 7 or iOS 11 and above.

Web administration application is usable in many browsers.

3.7 Performance

The application has fast response when connected with server and low latency while using.

4. Conceptual Diagram



Data Dictionary

Entity Data dictionary: describe all content of all entities	
Entity Name	Description
User	Abstract entity describes a user in system
Customer	Contain the customer information
Driver	Contain the driver information
Administrator	Contain the administrator information
Child	Contain the child information
Car	Contain the car information
Customer Requirement	Contain the customer requirement information
Driver Service	Contain the driver service information
Contract	Refer the contract between driver service and customer requirement. Contain the contract information.
Feedback	Contain the feedback information of customer for a driver service.
Daily trip	Contain the daily trip information.
Trip cancellation request	Contain the trip cancellation request for a day-off.
Contract cancellation request	Contain the contract cancellation request.

D. Software Design Description

1. Design Overview

- This document describes the technical and user interface design SCB System using web and mobile device. It includes the architectural design, the detailed design of common functions and business functions and the design of database model.
- The architectural design describes the overall architecture of the system and the architecture of each main component and subsystem.
- The detailed design describes static and dynamic structure for each component and functions. It includes class diagrams, class explanations and sequence diagrams for each use cases.
- The database design describes the relationships between entities and details of each entity.
- Document overview:
 - o Section 2: gives an overall description of the system architecture design.
 - o Section 3: gives component diagrams that describe the connection and integration of the system.
 - o Section 4: gives the detail design description which includes class diagram, class explanation, activity diagram and sequence diagram to details the application functions.
 - o Section 5: describe screen design.
 - o Section 6: describe a fully attribute ERD.
 - o Section 7: describe algorithms.

2. System Architectural Design

Software Architecture Design

School Bus System

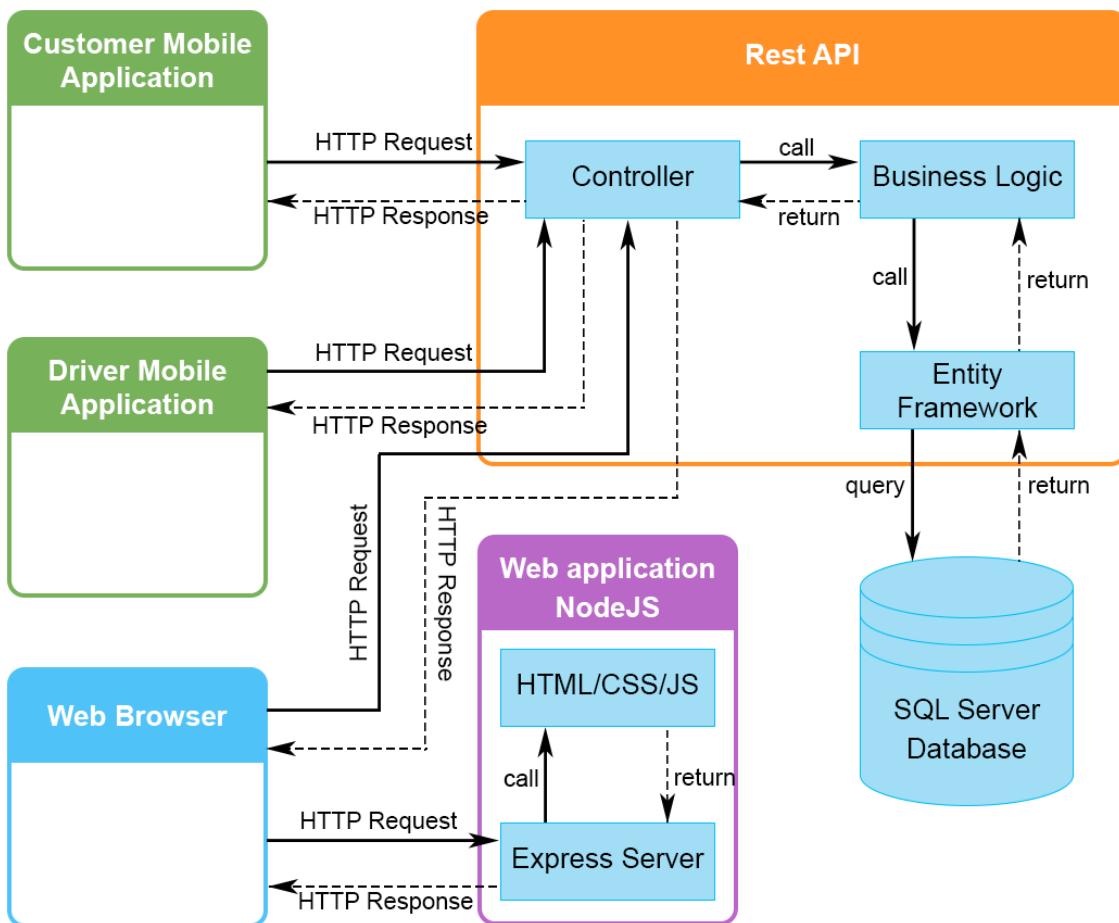


Figure - Software Architecture Design

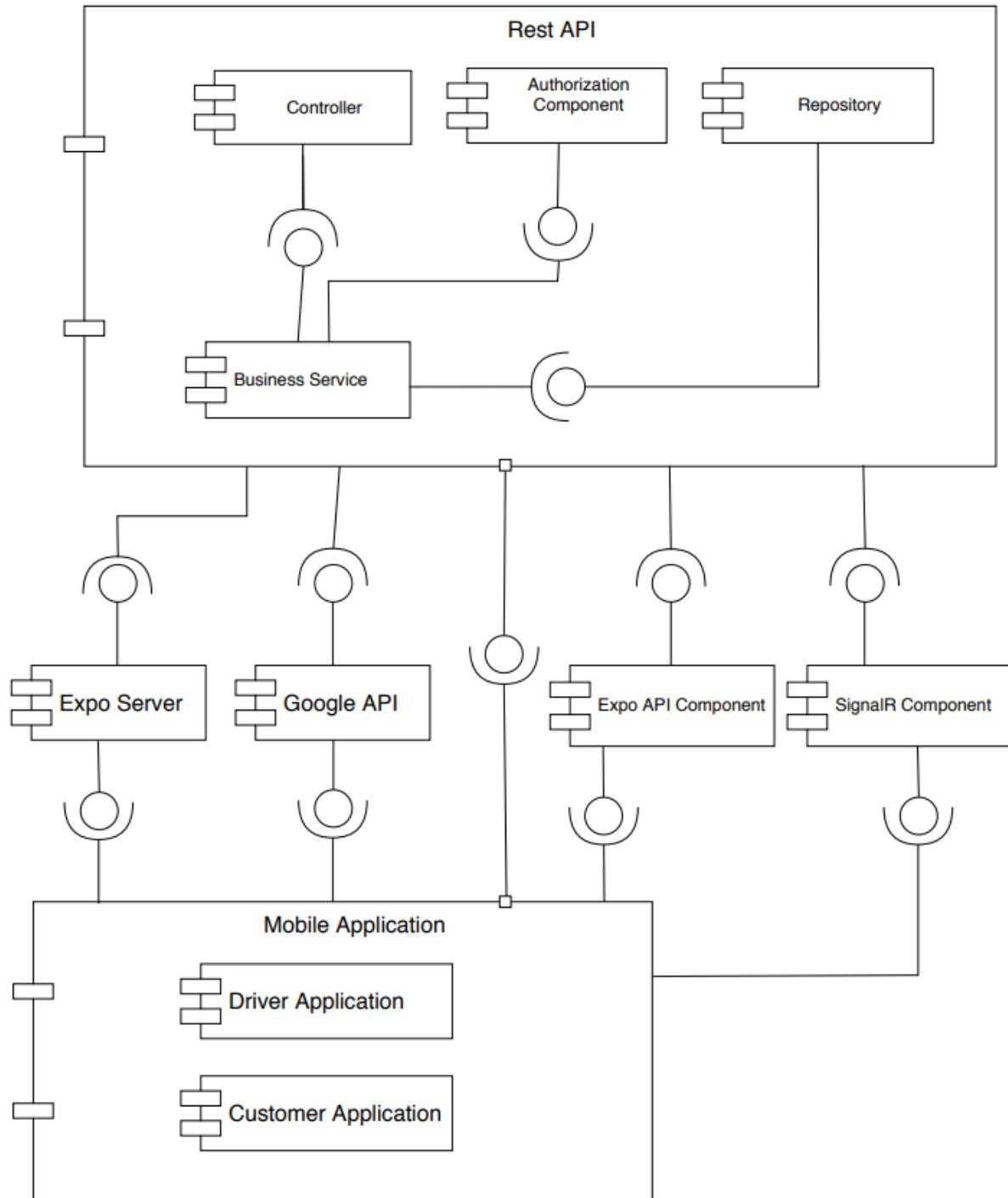
Description

Rest API: Because Mobile application and Web application are separated, web service is needed as a center to get and update data in database. Web service makes it easier to change business logic. Controller – Business Logic – Entity Framework is one of the most common structures used in modern systems.

Mobile application: Both driver and customer application are built with React-native and Expo framework. We choose React-native because it supports to build application in both Android and iOS. The component-based structure in React is the most common one used in user interface development, since it arranges everything in components, which are highly extendable, maintainable and reusable. Expo framework provides great SDK to develop mobile application, helps to build application easier and faster.

Web application: This is the web administration application developed with a stack combined by ReactJS, Webpack and some other related libraries. React gives many advantages for developing user interface as mentioned. Webpack is a JavaScript bundler, combining, minimizing, converting ES6 to pure JavaScript. Express Server is used to dispatch web application to user browsers.

3. Component Diagram

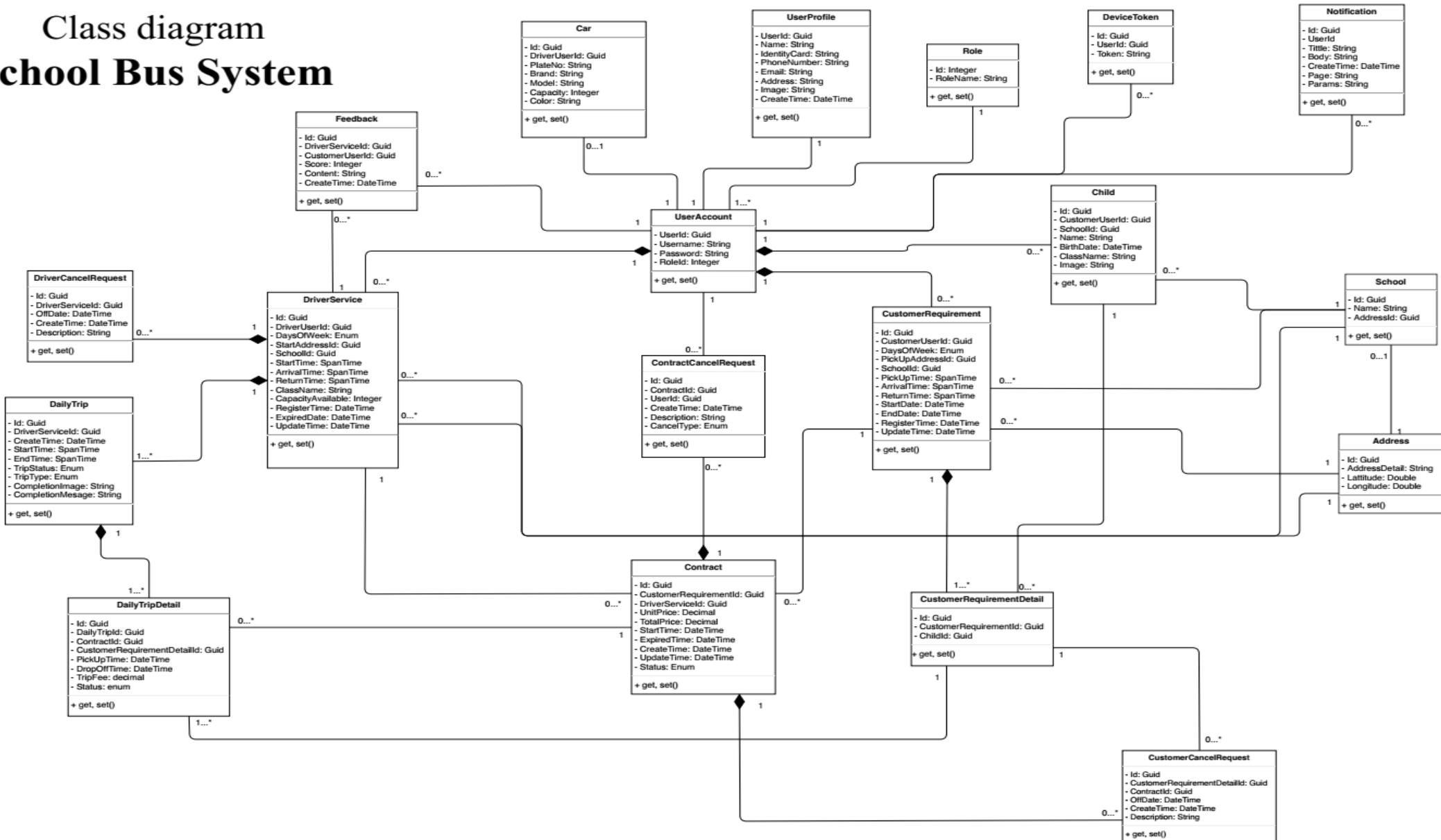


COMPONENT DICTIONARY: DESCRIBES COMPONENTS	
Rest API	
Controller	Handle requests and responses; accept input and convert it to commands for model and view.
Authorization Component	Check user role before API called.
Business Service	Handle business process, transform data.
Repository	Entity framework to connect to database.
COMPONENT DICTIONARY: DESCRIBES COMPONENTS	
Expo Server	Third party component that sends notification.
Google API	Handle map direction, place, geocoding and distance function.
Expo API component	Send notification to expo server.
SignalR component	Real time socket for location tracking function.
Mobile Application	
Driver Application	Android/IOS application
Customer Application	Android/IOS application

4. Detailed Description

4.1 Class Diagram

Class diagram School Bus System



CLASS DICTIONARY: DESCRIBE CLASS		
Class Name	Mapping column with Conceptual Diagram	Description
UserAccount	N/A	Not exist in conceptual diagram. Account information that contain login data.
UserProfile	User	Detail user information linked with UserAccount.
Role	N/A	Not exist in conceptual diagram. It's used to contain role information.
Car	Car	Information about Car linked to Driver account.
Child	Child	Children of customer who will directly use Driver Service.
DeviceToken	N/A	Not exist in conceptual diagram. Device identity linked to UserAccount. It's used to send notification.
Notification	N/A	Notification of user. User can read old notification out of OS notification.
Feedback	Feedback	Feedback of customer for driver services.
DriverService	DriverService	SchoolBus services that driver registered to our system.
DriverCancelRequest	Trip cancellation request	Contain driver request to off a trip.
CustomerRequirement	Customer requirement	Customer's requirements for a SchoolBus service. It's used to matching with DriverService.
CustomerRequirementDetail	N/A	Not exist in conceptual diagram. It contains children who linked to CustomerRequirement.
CustomerCancelRequest	Trip cancellation request	Customer's cancel request for child in a day.
Contract	Contract	It contains information of a contract between DriverService and CustomerRequirement.
ContractCancelRequest	Contract cancellation request	Requests of customer or driver to cancel current active contract.
DailyTrip	Daily trip	Trip information of a day in the contract.

DailyTripDetail	N/A	Not exist in conceptual diagram. Information for a child in a DailyTrip. It mostly contains status of child which changed after each stage of a trip.
School	N/A	Not exist in conceptual diagram. School information stored for reusing.
Address	N/A	Not exist in conceptual diagram. Contain address detail used for map, location function.

Table Class dictionary

Class Diagram Explanation

4.1.1 UserAccount

Attribute

Attribute	Type	Visibility	Description
UserId	Guid	Private	Unique identifier of user account
Username	String	Private	Account's username
Password	String	Private	Account's password
RoleId	Integer	Private	Identify of role

Method

Method	Return type	Visibility	Description
Getter	Attribute type	Public	Get attribute value
Setter	Void	Public	Set value of attribute

4.1.2 UserProfile

Attribute

Attribute	Type	Visibility	Description
UserId	Guid	Private	UserId of user account
Name	String	Private	Full name of user profile
IdentifyCard	String	Private	Identify card of user profile
PhoneNumber	String	Private	Phone number of user profile
Email	String	Private	Email of user profile
Address	String	Private	Address of user profile

Image	String	Private	Image reference link of user profile
CreateTime	DateTime	Private	Profile's time created

Method

Method	Return type	Visibility	Desctiption
Getter	Attribute type	Public	Get attribute value
Setter	Void	Public	Set value of attribute

4.1.3 Role**Attribute**

Attribute	Type	Visibility	Desctiption
Id	Integer	Private	Unique identifier of role
RoleName	String	Private	Role's name

Method

Method	Return type	Visibility	Desctiption
Getter	Attribute type	Public	Get attribute value
Setter	Void	Public	Set value of attribute

4.1.4 Car**Attribute**

Attribute	Type	Visibility	Desctiption
Id	Guid	Private	Unique identifier of car
DriverUserId	Guid	Private	User Id of driver
PlateNo	String	Private	Plate no of driver's car
Brand	String	Private	Brand of driver's car
Model	String	Private	Model of driver's car
Capacity	Integer	Private	Capacity of driver's car
Color	String	Private	Color of driver's car

Method

Method	Return type	Visibility	Desctiption

Getter	Attribute type	Public	Get attribute value
Setter	Void	Public	Set value of attribute

4.1.5 Child

Attribute

Attribute	Type	Visibility	Description
Id	Guid	Private	Unique identifier of child
CustomerUserId	Guid	Private	User Id of customer
SchoolId	Guid	Private	School identifier of child
Name	String	Private	Name of child
Birthdate	DateTime	Private	Birthdate of child
ClassName	String	Private	Class in school of child
Image	String	Private	Image reference link of child

Method

Method	Return type	Visibility	Description
Getter	Attribute type	Public	Get attribute value
Setter	Void	Public	Set value of attribute

4.1.6 DeviceToken

Attribute

Attribute	Type	Visibility	Description
Id	Guid	Private	Unique identifier of device token
UserId	Guid	Private	User Id of user account
Token	String	Private	Token of device token

Method

Method	Return type	Visibility	Description
Getter	Attribute type	Public	Get attribute value
Setter	Void	Public	Set value of attribute

4.1.7 Notification

Attribute

Attribute	Type	Visibility	Desctiption
Id	Guid	Private	Unique identifier of notification
UserId	Guid	Private	UserId of user account
Title	String	Private	Title of notification
Body	String	Private	Body of notification
CreateTime	DateTime	Private	Notification's time created
Page	String	Private	Page of notification
Params	String	Private	Params of notification

Method

Method	Return type	Visibility	Desctiption
Getter	Attribute type	Public	Get attribute value
Setter	Void	Public	Set value of attribute

4.1.8 Feedback**Attribute**

Attribute	Type	Visibility	Desctiption
Id	Guid	Private	Unique identifier of feedback
DriverServiceId	Guid	Private	Service identifier of driver
CustomerUserId	Guid	Private	UserId of customer
Score	Integer	Private	Score of feedback
Content	String	Private	Content of feedback
CreateTime	DateTime	Private	Feedback's time created

Method

Method	Return type	Visibility	Desctiption
Getter	Attribute type	Public	Get attribute value
Setter	Void	Public	Set value of attribute

4.1.9 DriverService**Attribute**

Attribute	Type	Visibility	Description
Id	Guid	Private	Unique identifier of service
DriverUserId	Guid	Private	User Id of driver
DaysOfWeek	Enum	Private	Day of week in service
StartAddressId	Guid	Private	Start address of service
SchoolId	Guid	Private	School identifier of service
StartTime	SpanTime	Private	Time of starting from driver's home of service
ArrivalTime	SpanTime	Private	Arrival time to school of service
ReturnTime	SpanTime	Private	Time of returning driver's home of service
ClassName	String	Private	Child's class of driver
CapacityAvailable	Integer	Private	Capacity available in car of service
RegisterTime	DateTime	Private	Service's time registered
ExpiredTime	DateTime	Private	Service's time expired
UpdateTime	DateTime	Private	Service's time updated

Method

Method	Return type	Visibility	Description
Getter	Attribute type	Public	Get attribute value
Setter	Void	Public	Set value of attribute

4.1.10 DriverCancelRequest**Attribute**

Attribute	Type	Visibility	Description
Id	Guid	Private	Unique identifier of driver cancel request
DriverServiceId	Guid	Private	Service identifier of driver
OffDate	DateTime	Private	Off date of driver service
CreateTime	DateTime	Private	Cancel request's time created
Description	String	Private	Description of request

Method

Method	Return type	Visibility	Desctiption
Getter	Attribute type	Public	Get attribute value
Setter	Void	Public	Set value of attribute

4.1.11 CustomerRequirement

Attribute

Attribute	Type	Visibility	Desctiption
Id	Guid	Private	Unique identifier of customer requirement
CustomerUserId	Guid	Private	User Id of customer
DaysOfWeek	Enum	Private	Day of week in requirement
PickUpAddressId	Guid	Private	Pick up address of requirement
SchoolId	Guid	Private	School identifier of requirement
PickUpTime	SpanTime	Private	Time of picking up from customer's home of requirement
ArrivalTime	SpanTime	Private	Arrival time to school of requirement
ReturnTime	SpanTime	Private	Time of returning customer's home of requirement
StartDate	DateTime	Private	Requested start date
EndDate	DateTime	Private	Requested end date
RegisterTime	DateTime	Private	Requirement's time registered
UpdateTime	DateTime	Private	Requirement's time updated

Method

Method	Return type	Visibility	Desctiption
Getter	Attribute type	Public	Get attribute value
Setter	Void	Public	Set value of attribute

4.1.12 CustomerRequirementDetail

Attribute

Attribute	Type	Visibility	Desctiption
Id	Guid	Private	Unique identifier of customer requirement detail

CustomerRequirementId	Guid	Private	Requirement identifier of customer
ChildId	Guid	Private	Child identifier of customer

Method

Method	Return type	Visibility	Description
Getter	Attribute type	Public	Get attribute value
Setter	Void	Public	Set value of attribute

4.1.13 CustomerCancelRequest**Attribute**

Attribute	Type	Visibility	Description
Id	Guid	Private	Unique identifier of customer cancel request
CustomerRequirementDetail Id	Guid	Private	Requirement detail identifier of requirement's customer
ContractId	Guid	Private	Contract identifier of customer and driver
OffDate	DateTine	Private	Off date of customer requirement
CreateTime	DateTine	Private	Cancel request's time created
Description	String	Private	Description of request

Method

Method	Return type	Visibility	Description
Getter	Attribute type	Public	Get attribute value
Setter	Void	Public	Set value of attribute

4.1.14 Contract**Attribute**

Attribute	Type	Visibility	Description
Id	Guid	Private	Unique identifier of contract
CustomerRequirementId	Guid	Private	Requirement identifier of customer
DriverServiceId	Guid	Private	Service identifier of driver

UnitPrice	Decimal	Private	Unit price per day in daily trip
TotalPrice	Decimal	Private	Total price of contract
StartTime	DateTime	Private	Start time of contract
ExpiredTime	DateTime	Private	Contract's time expired
CreateTime	DateTime	Private	Contract's time create
UpdateTime	DateTime	Private	Contract's time updated
Status	Enum	Private	Status of contract

Method

Method	Return type	Visibility	Desctiption
Getter	Attribute type	Public	Get attribute value
Setter	Void	Public	Set value of attribute

4.1.15 ContractCancelRequest

Attribute

Attribute	Type	Visibility	Desctiption
Id	Guid	Private	Unique identifier of contract cancel request
ContractId	Guid	Private	Contract identifier of customer and driver
UserId	Guid	Private	UserId of user account
CreateTime	DateTime	Private	Request's time create
Desctiption	String	Private	Description of request
CancelType	Enum	Private	Cancel type of request

Method

Method	Return type	Visibility	Desctiption
Getter	Attribute type	Public	Get attribute value
Setter	Void	Public	Set value of attribute

4.1.16 DailyTrip

Attribute

Attribute	Type	Visibility	Desctiption

Id	Guid	Private	Unique identifier of daily trip
DriverServiceId	Guid	Private	Service identifier of driver
CreateTime	DateTime	Private	Daily trip's time create
StartTime	SpanTime	Private	Time of starting daily trip
EndTime	SpanTime	Private	Time of finishing daily trip
TripStatus	Enum	Private	Trip status of daily trip
TripType	Enum	Private	Trip type of daily trip
CompletionImage	String	Private	Image reference link of daily trip when finish
CompletionMessage	String	Private	Driver's message sent to customer when finish

Method

Method	Return type	Visibility	Desctiption
Getter	Attribute type	Public	Get attribute value
Setter	Void	Public	Set value of attribute

4.1.17 DailyTripDetail**Attribute**

Attribute	Type	Visibility	Desctiption
Id	Guid	Private	Unique identifier of daily trip detail
DailyTripId	Guid	Private	Daily trip identifier of driver
ContractId	Guid	Private	Contract identifier of driver and customer
CustomerRequirementDetailId	Guid	Private	Requirement detail identifier of customer
PickUpTime	SpanTime	Private	Time of picking up child
DropOffTime	SpanTime	Private	Time of dropping off child
TripFee	Decimal	Private	Fee per child in daily trip
Status	Enum	Private	Status of daily trip detail

Method

Method	Return type	Visibility	Desctiption
Getter	Attribute type	Public	Get attribute value

Setter	Void	Public	Set value of attribute
--------	------	--------	------------------------

4.1.18 School

Attribute

Attribute	Type	Visibility	Description
Id	Guid	Private	Unique identifier of school
Name	String	Private	Name of school
AddressId	Guid	Private	Address identifier of school

Method

Method	Return type	Visibility	Description
Getter	Attribute type	Public	Get attribute value
Setter	Void	Public	Set value of attribute

4.1.18 Address

Attribute

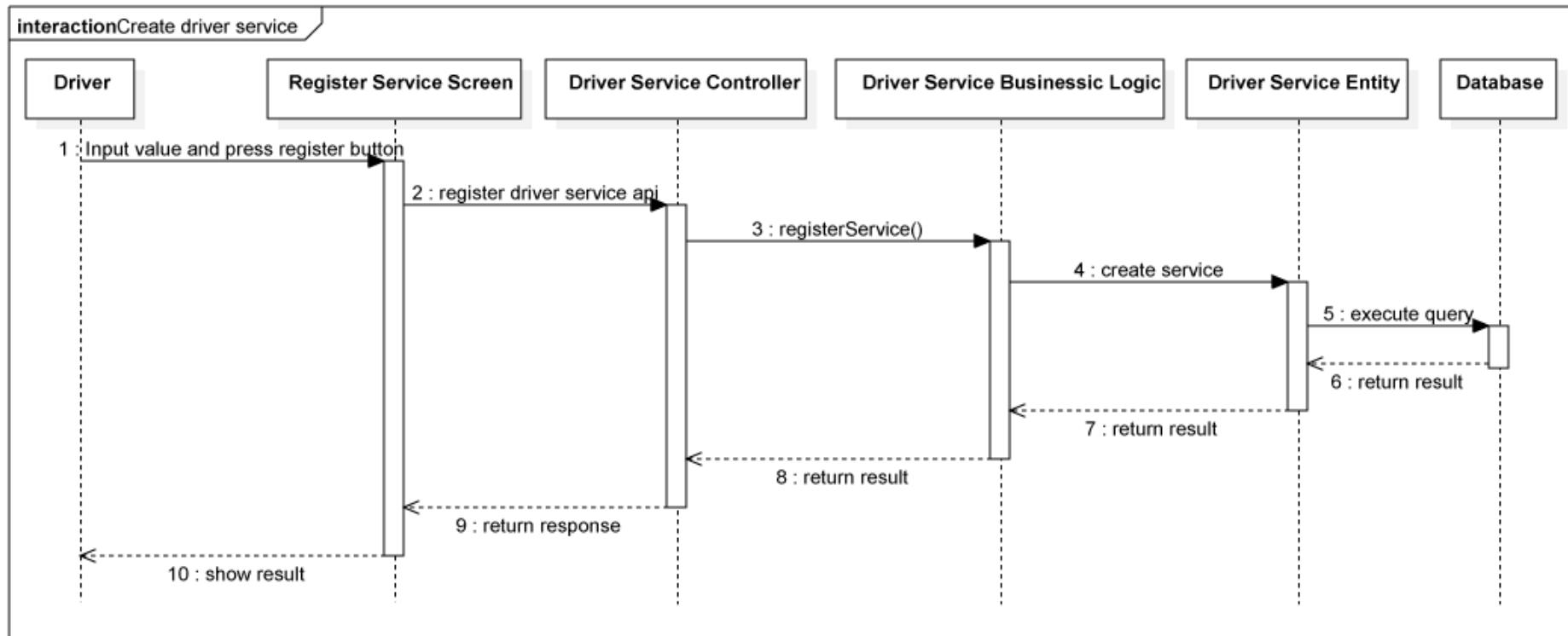
Attribute	Type	Visibility	Description
Id	Guid	Private	Unique identifier of address
AddressDetail	String	Private	Detail of address
Latitude	Double	Private	Latitude of address
Longitude	Double	Private	Longitude of address

Method

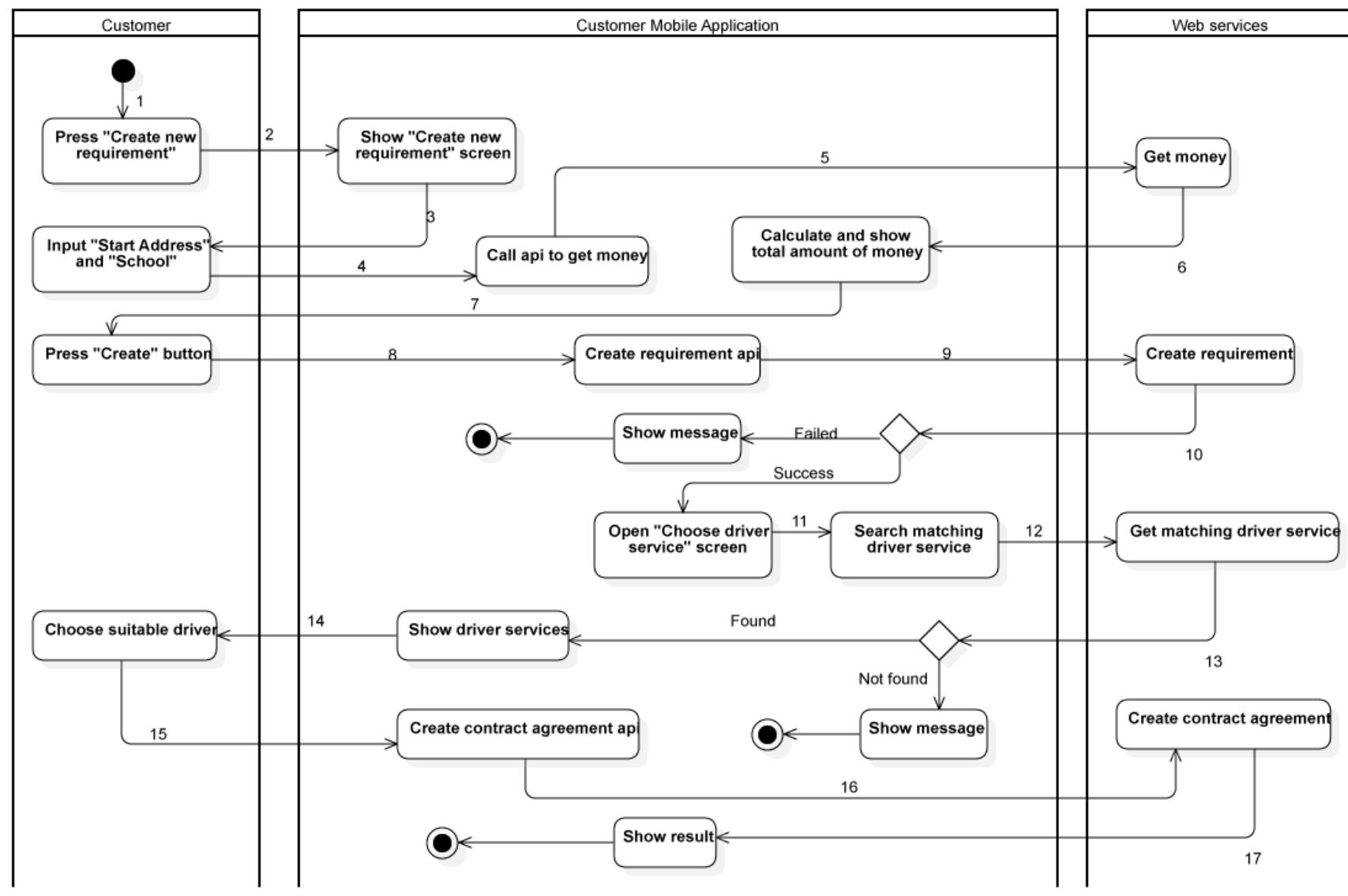
Method	Return type	Visibility	Description
Getter	Attribute type	Public	Get attribute value
Setter	Void	Public	Set value of attribute

4.2 Interaction Diagram

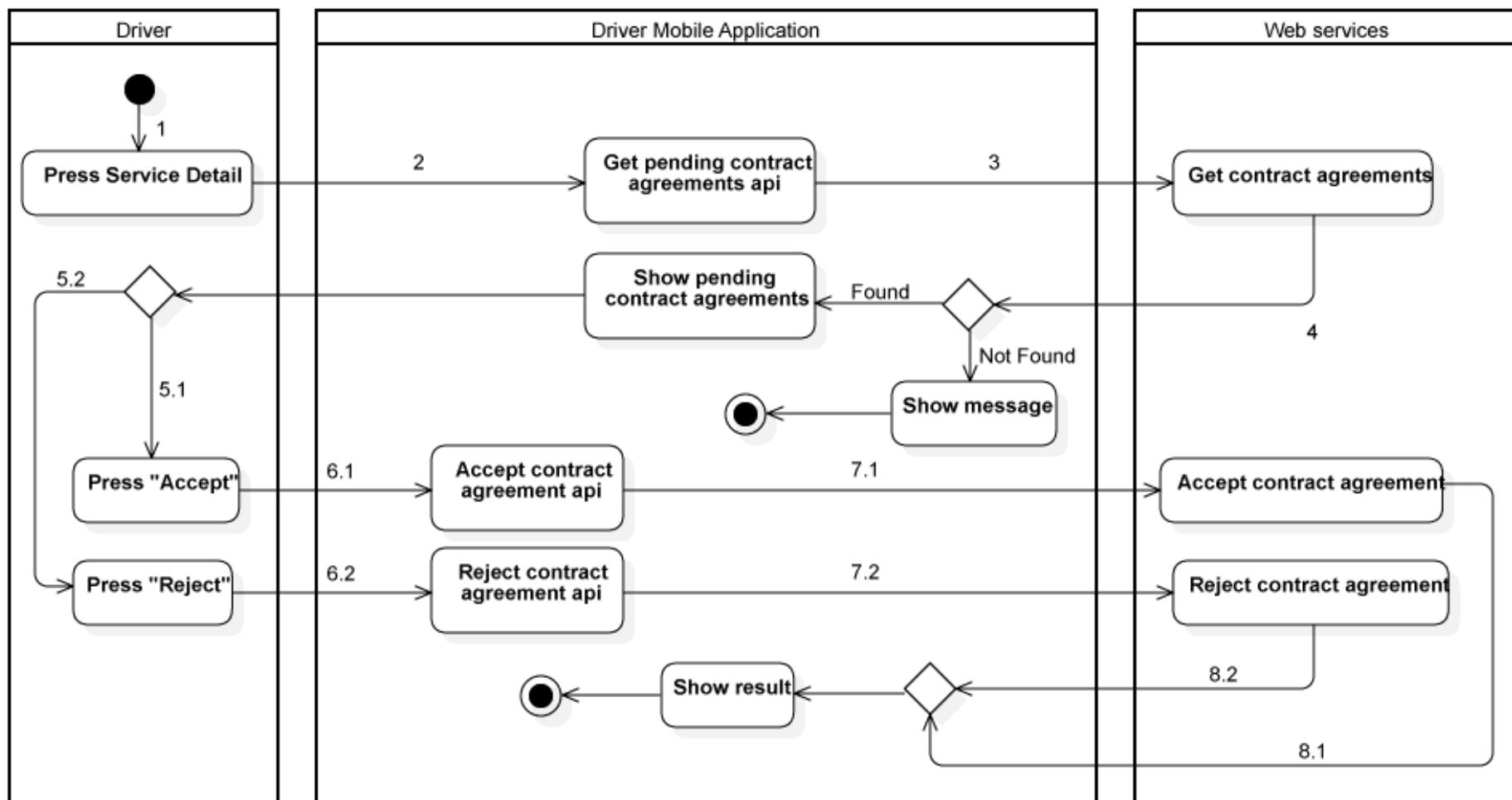
4.2.1 <Driver> Create driver service



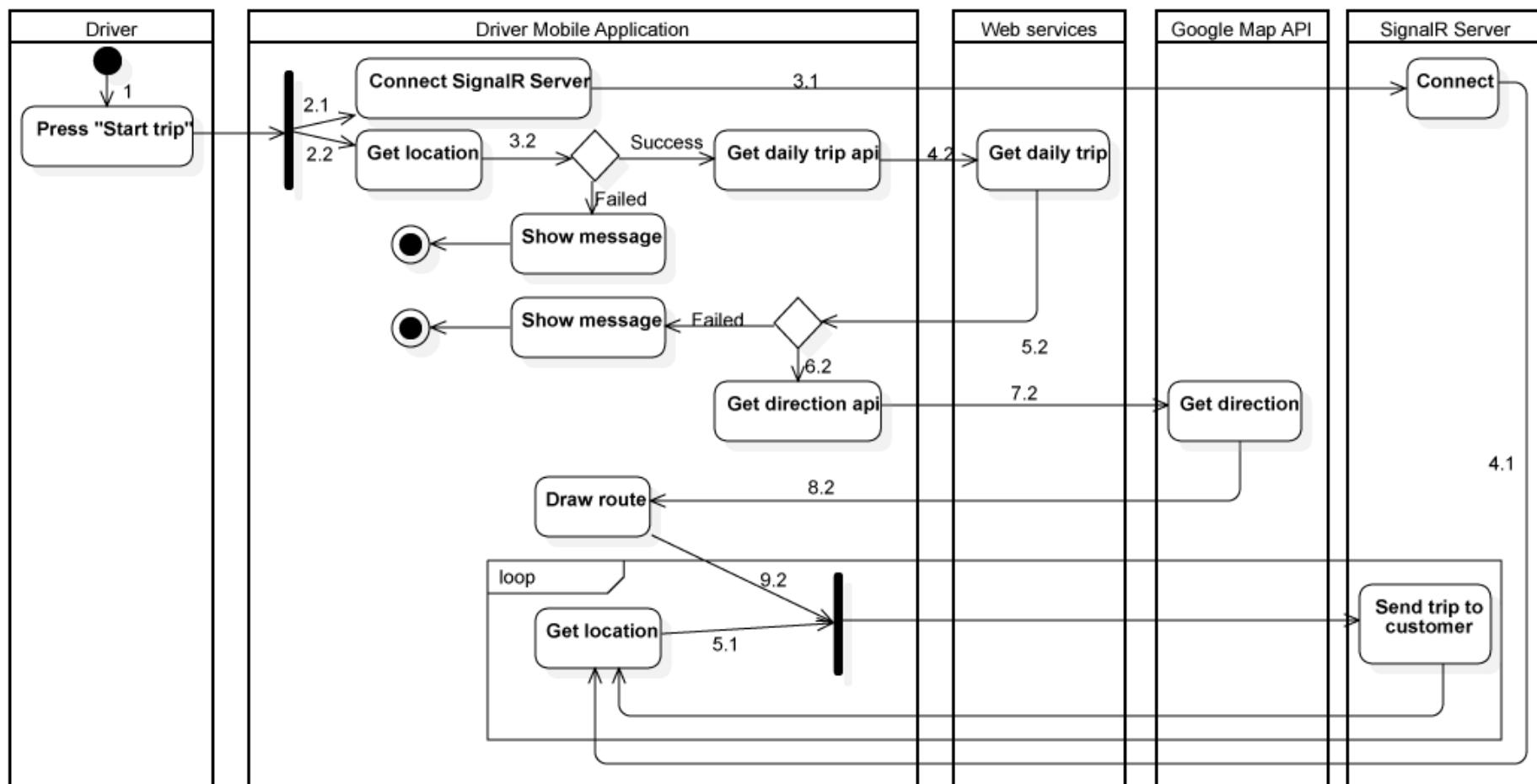
4.2.2 <Customer> Create customer requirement



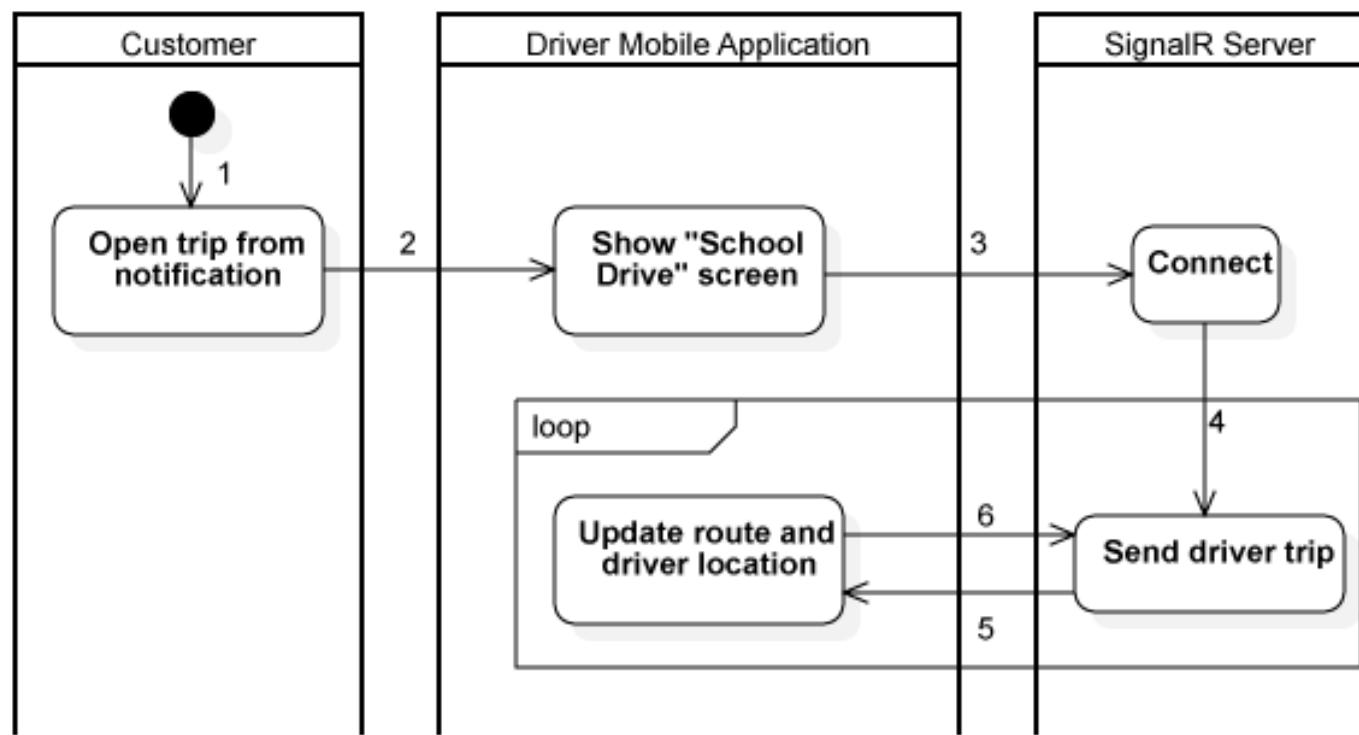
4.2.3 <Driver> Respond customer request



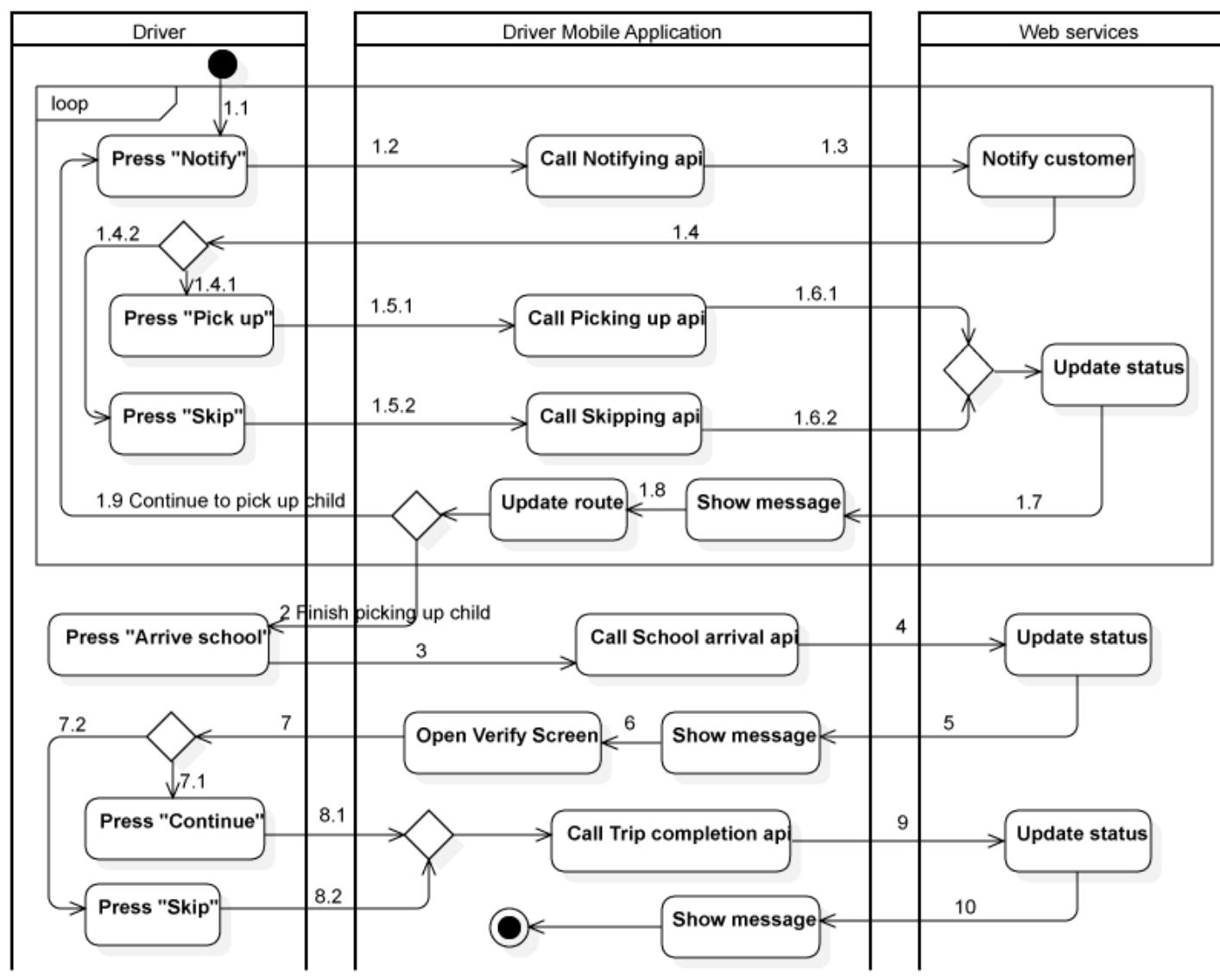
4.2.4 <Driver> Get daily trip



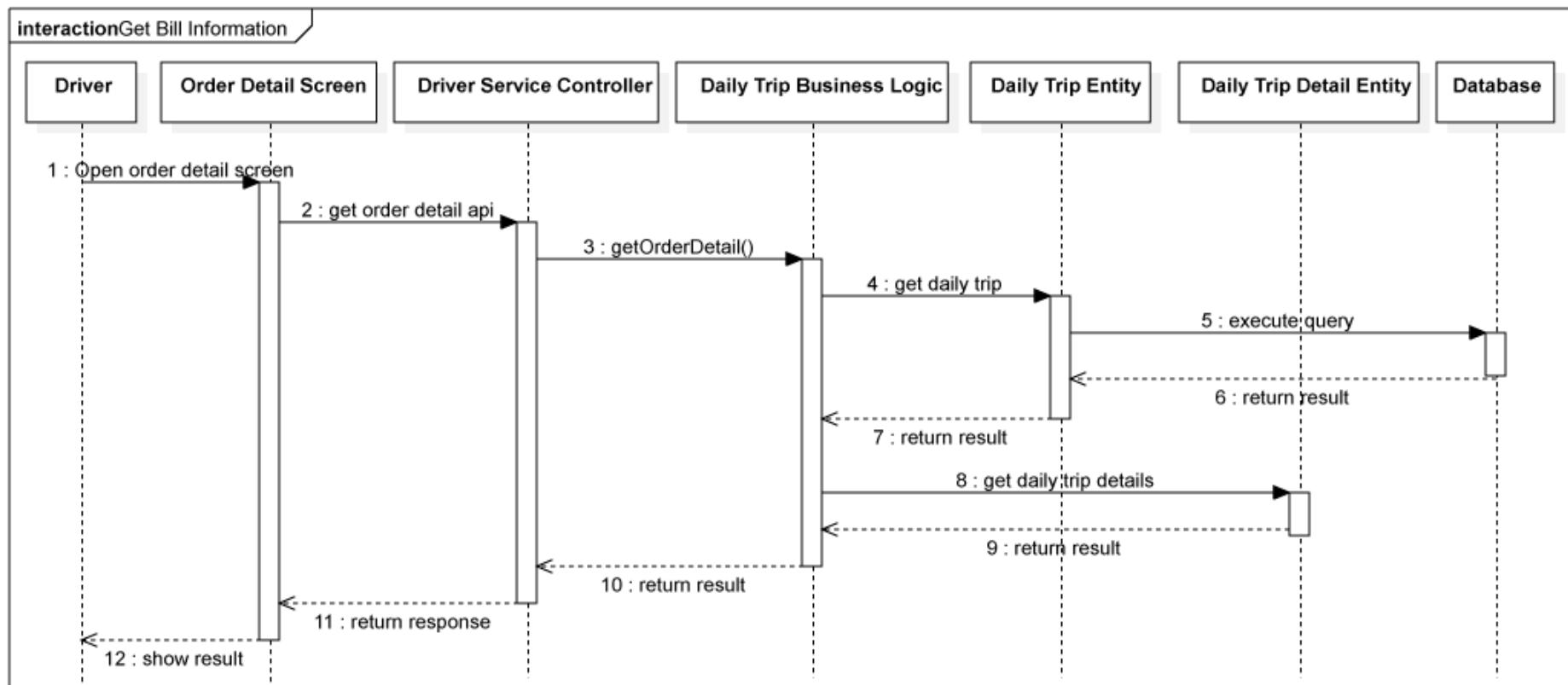
4.2.5 <Customer> Track driver



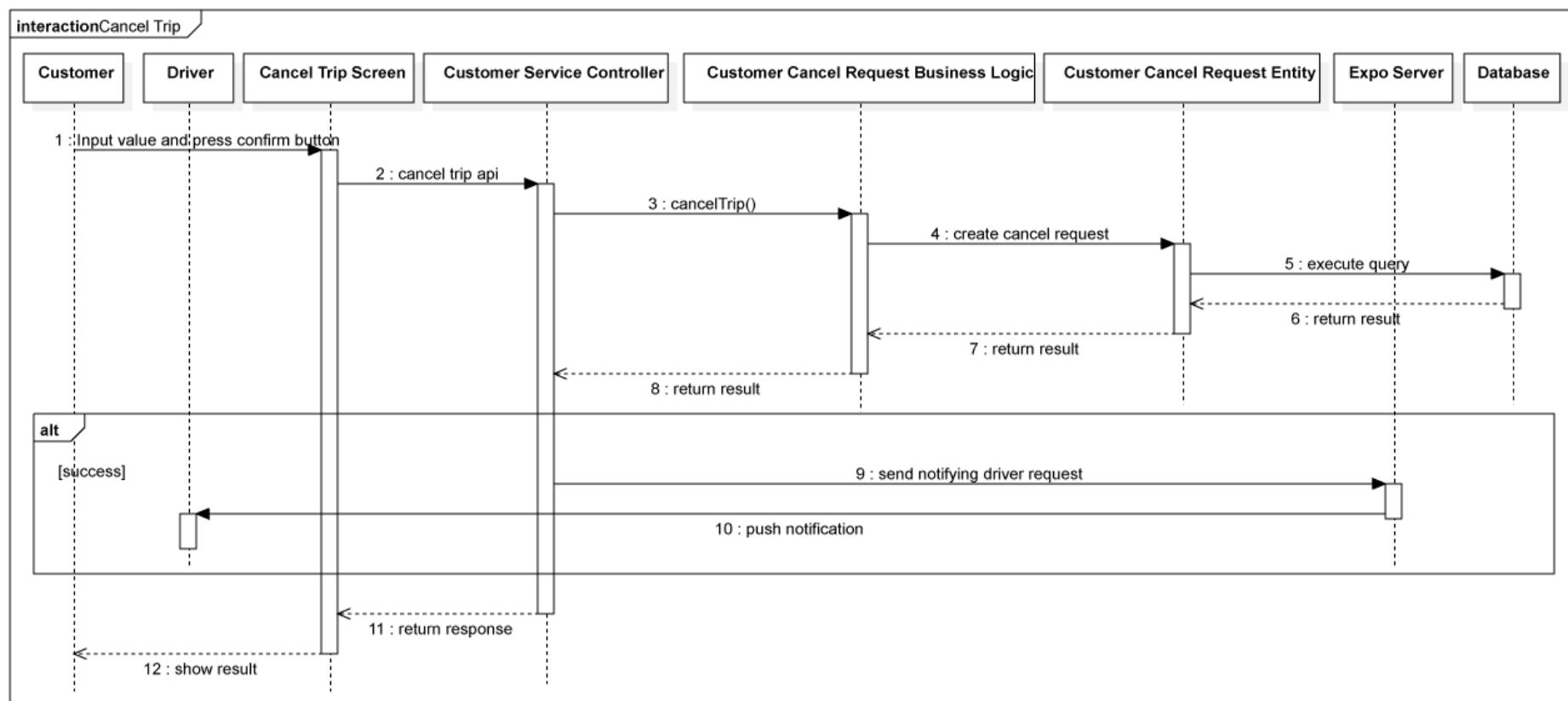
4.2.6 <Driver> Complete trip



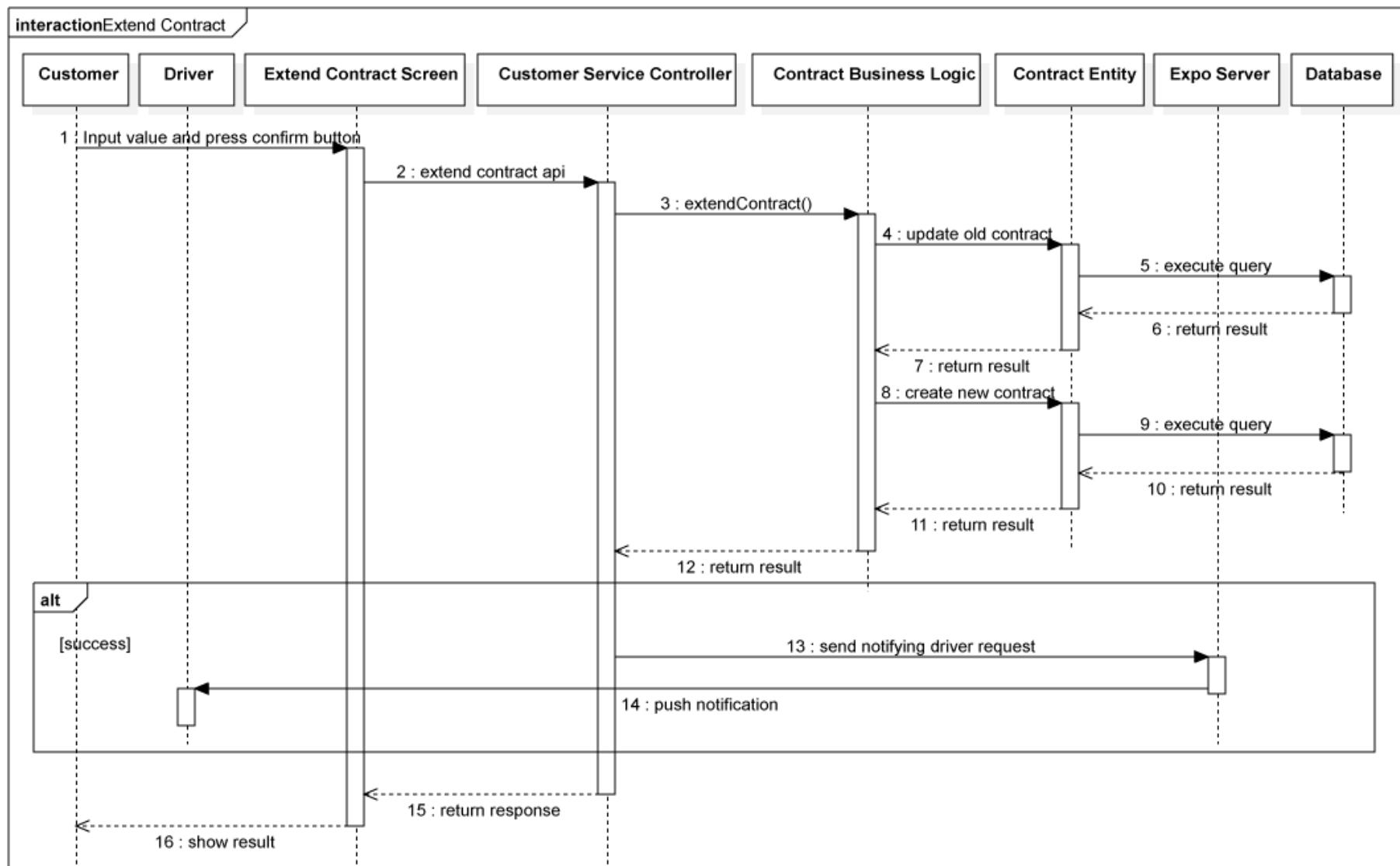
4.2.7 <Driver> Get Bill Information



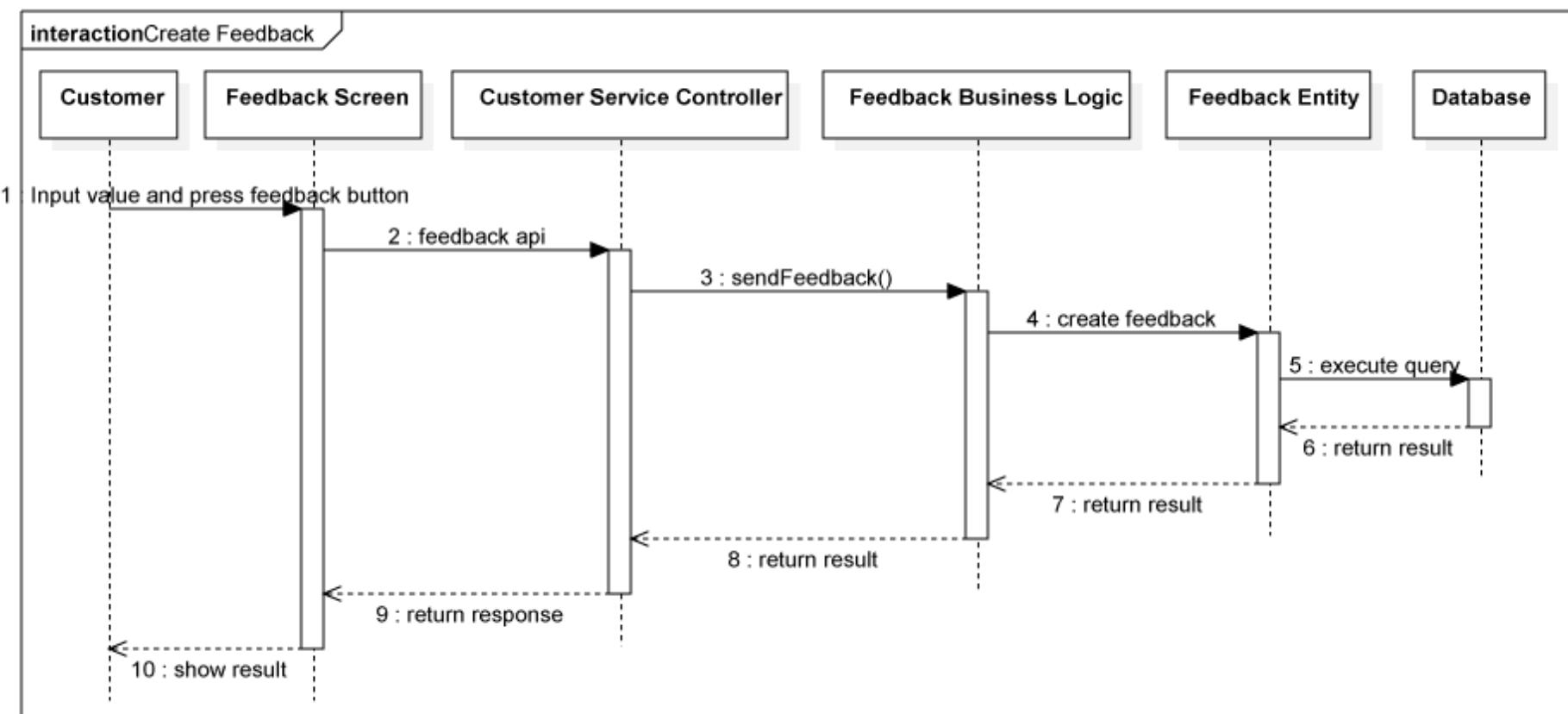
4.2.8 <Customer> Cancel trip



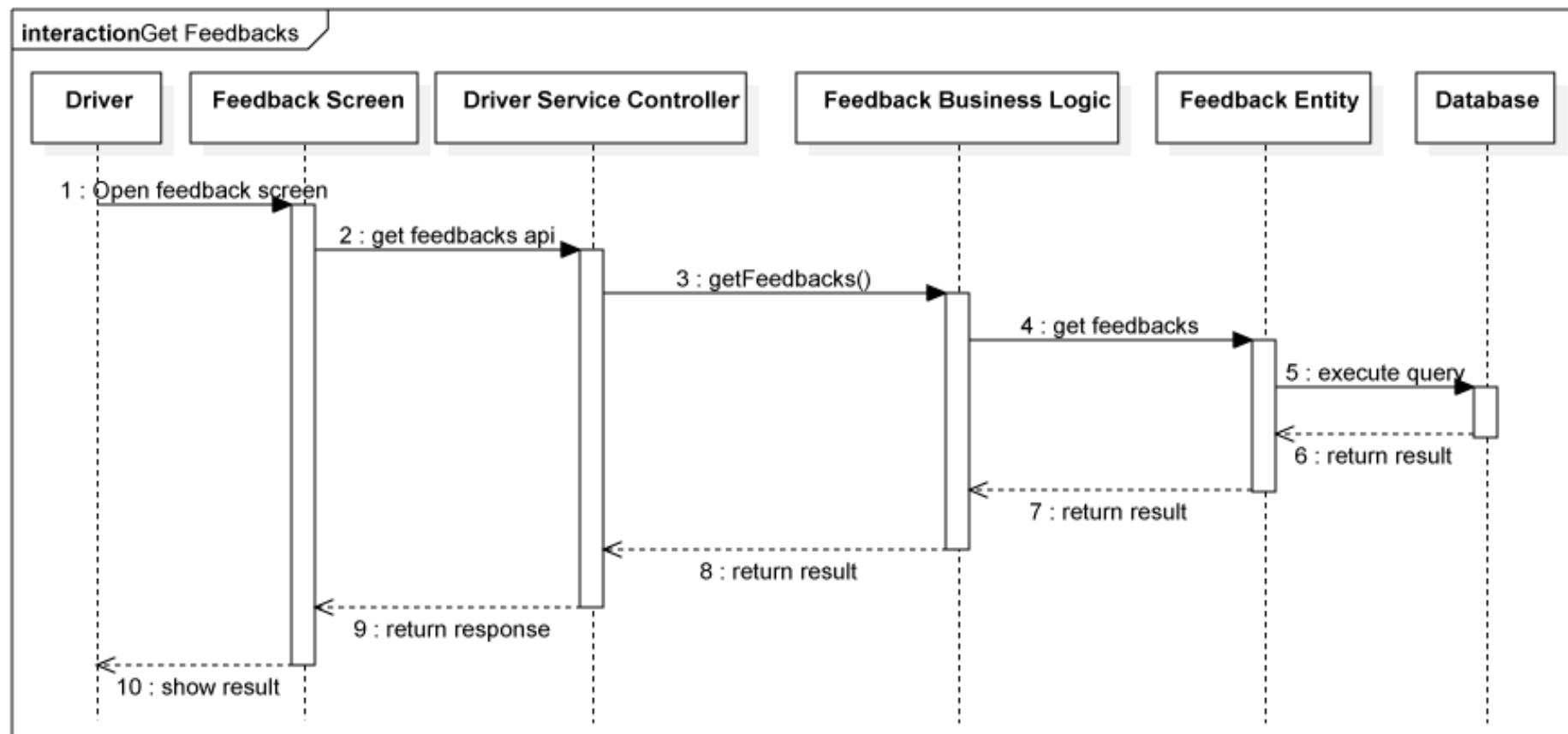
4.2.9 <Customer> Extend contract



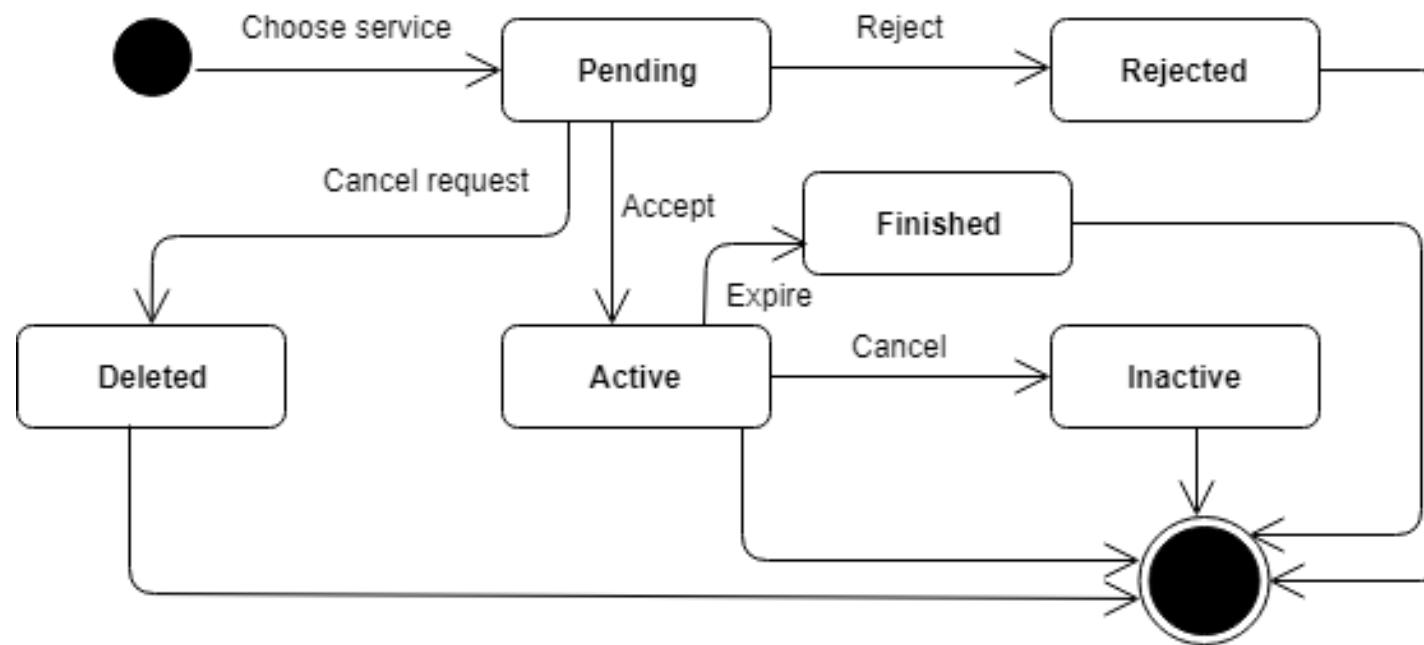
4.2.10 <Customer> Create feedback



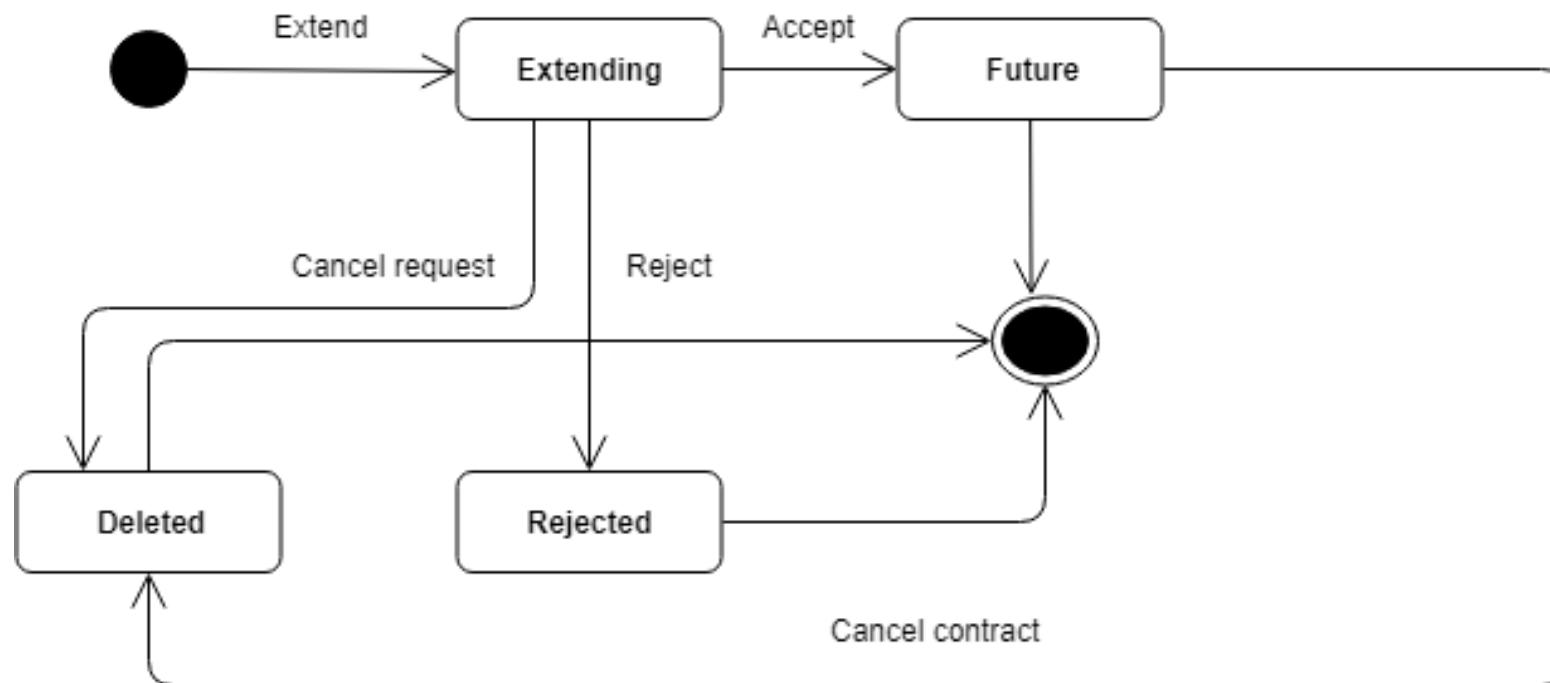
4.2.11 <Driver> Get feedbacks



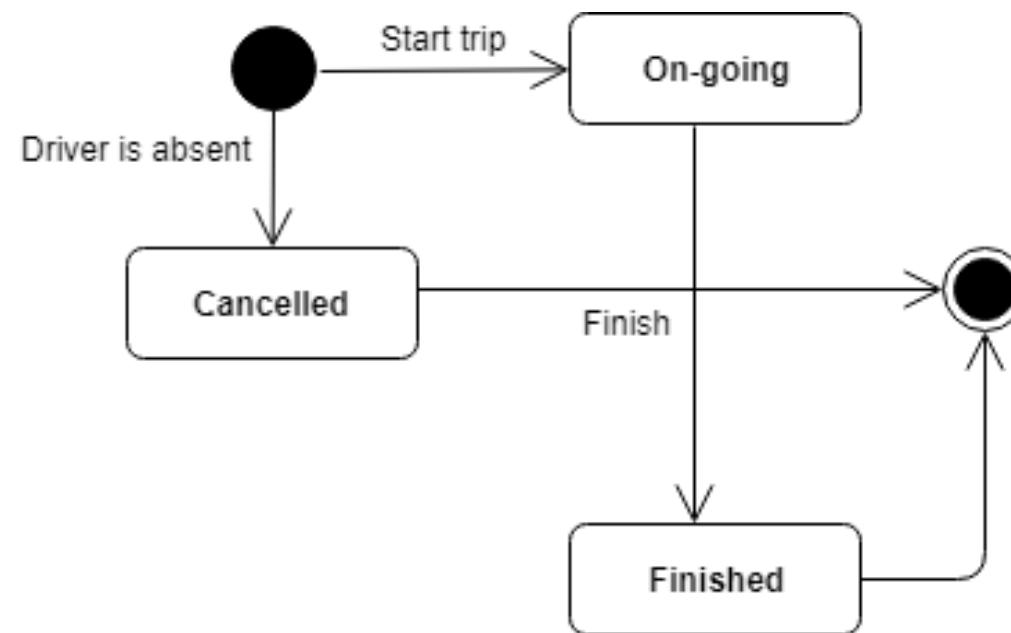
4.2.12 <Contract> State machine diagram



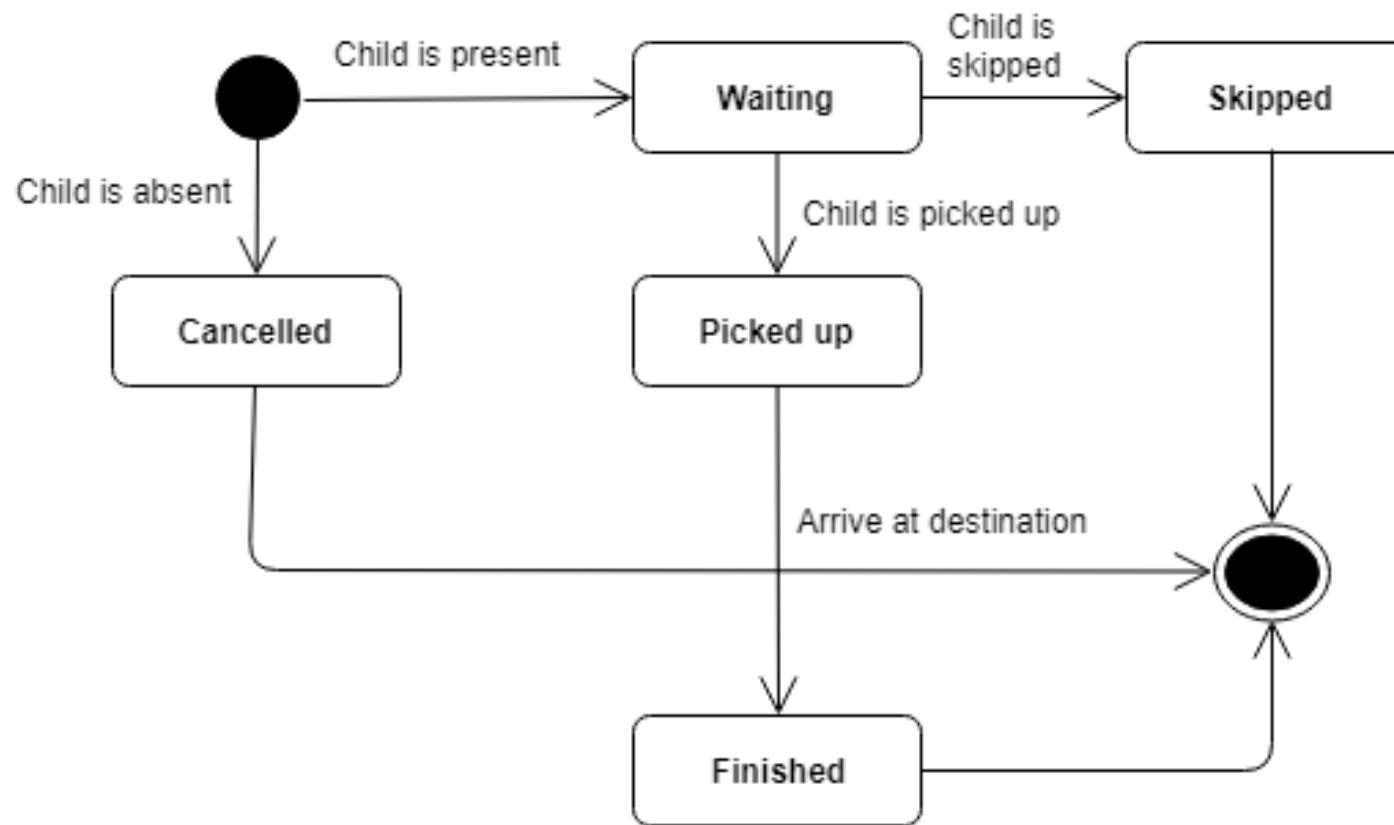
4.2.13 <Contract Extending> State machine diagram



4.2.14 <Daily Trip> State machine diagram



4.2.15 <Daily Trip - Child> State machine diagram



5. User Interface Design

5.1 Login Page

Fields

No	Field Name	Description	Read-only	Mandatory	Control Type	Data Type	Length
1	Username	User's mobile's phone	No	Yes	Textbox	Number	10 - 11
2	Password	User's password	No	Yes	Textbox	String	Greater than 0

Buttons/Hyperlinks

No	Function	Description	Validation	Outcome
3	Login	Login to the system	Username, password is not empty	Redirect to the home screen
4	Go to register screen	Switch to register screen	N/A	Redirect to the register screen

5.2 Register Page

Số điện thoại
0123456789

Mật khẩu
...

Xác nhận mật khẩu
...

ĐĂNG KÍ

Đăng nhập với tài khoản có sẵn

Fields

No	Field Name	Description	Read-only	Mandatory	Control Type	Data Type	Length
1	Username	User's mobile's phone	No	Yes	Textbox	Number	10 - 11
2	Password	User's password	No	Yes	Password	String	Greater than 0
3	Confirm Password	Confirm user's password	No	Yes	Password	String	Greater than 0

Buttons/Hyperlinks

No	Function	Description	Validation	Outcome
4	Register	Register as new account to the system	Password is confirmed	Redirect to the updating profile screen
5	Go to login screen	Switch to the login screen	N/A	Switch to the login screen

5.3 Update Profile Page

The screenshot shows a mobile application interface for updating a user's profile. The title bar indicates the device is at 74% battery and 16:59. The main screen has a header "Cập nhật thông tin cá nhân". It contains the following fields:

1. User's profile picture.
2. Name: Nguyễn Việt Hùng.
3. ID Number: 0123456789.
4. Address: 213 Lê Thánh Tôn, Bến Thành, Quận 1, Hồ Chí Minh, Việt Nam, accompanied by a location pin icon.
5. A green rounded rectangular button labeled "Kết thúc".

Fields

No	Field Name	Description	Read-only	Mandatory	Control Type	Data Type	Length
1	Image	User's image	No	Yes	ImagePicker	ImageURI	
2	Name	User's name	No	Yes	Textbox	String	Greater than 0
3	Identity Number	User's ID number	No	Yes	Textbox	Number	Greater than 0
4	Address	Address	No	Yes	Place Autocomplete	String	Greater than 0

Buttons/Hyperlinks

No	Function	Description	Validation	Outcome
5	Update Profile	Update the profile	All fields are mandatory	Redirect to the updating profile screen

5.4 Update Car Page

The screenshot shows a mobile application interface for updating car information. The title bar says "Xe của tôi". The form contains the following fields:

- Biển số xe: ABC-123456 (labeled 1)
- Hiệu: Toyota (labeled 2)
- Model: Innova (labeled 3)
- Số ghế: 6 (labeled 4)
- Màu: Đen (labeled 5)
- Cập nhật thông tin xe (labeled 6, a green button at the bottom)

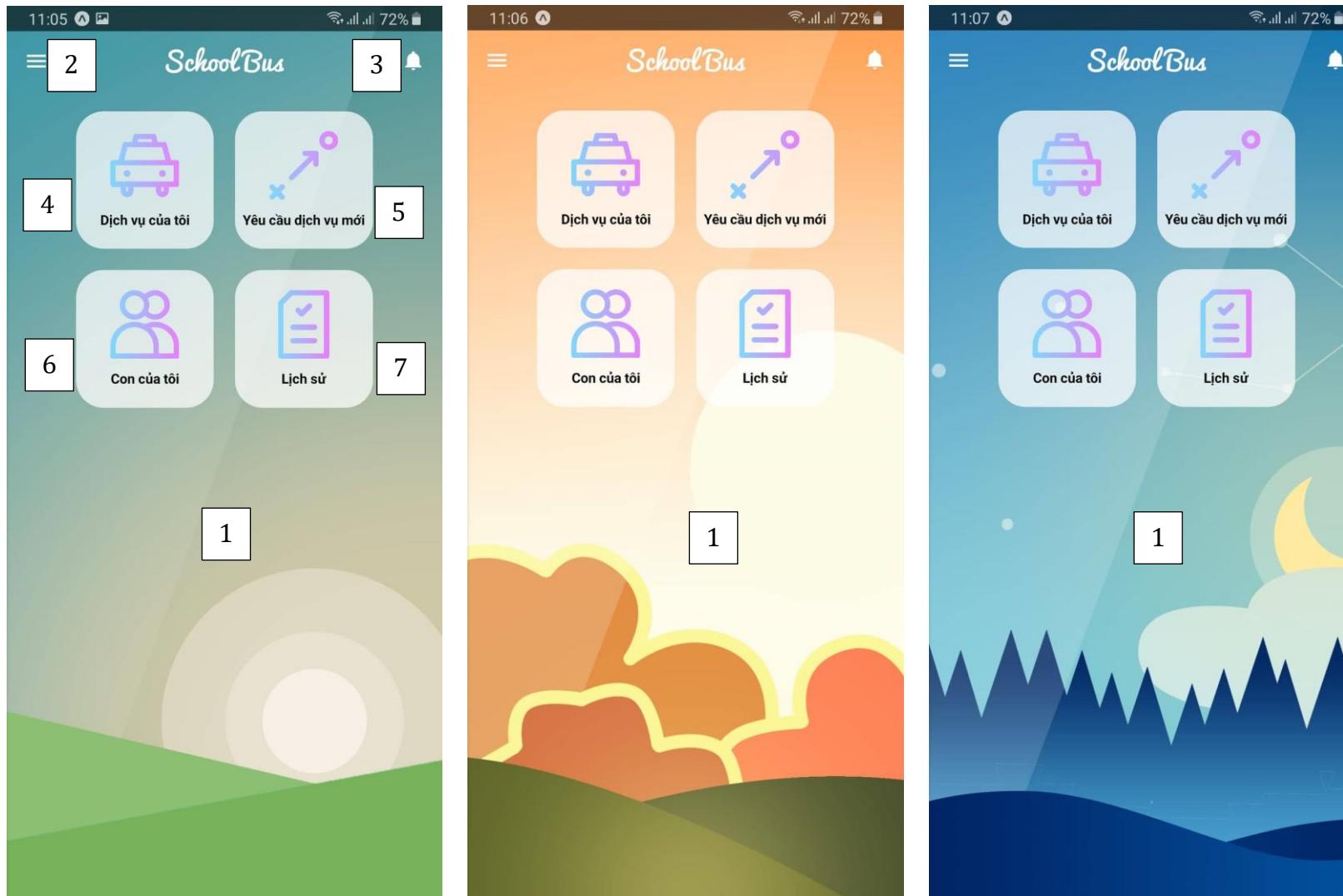
Fields

No	Field Name	Description	Read-only	Mandatory	Control Type	Data Type	Length
1	PlateNo	Car's plate number	Yes	Yes	Text	String	Greater than 0
2	Brand	Car's brand	Yes	Yes	Text	String	Greater than 0
3	Model	Car's model	Yes	Yes	Text	String	Greater than 0
4	Capacity	Car's capacity	Yes	Yes	Text	String	Greater than 0
5	Color	Car's color	Yes	Yes	Text	String	Greater than 0

Buttons/Hyperlinks

No	Function	Description	Validation	Outcome
6	Update Car	Update your car	N/A	Redirect to home screen

5.5 Home Screen - Customer



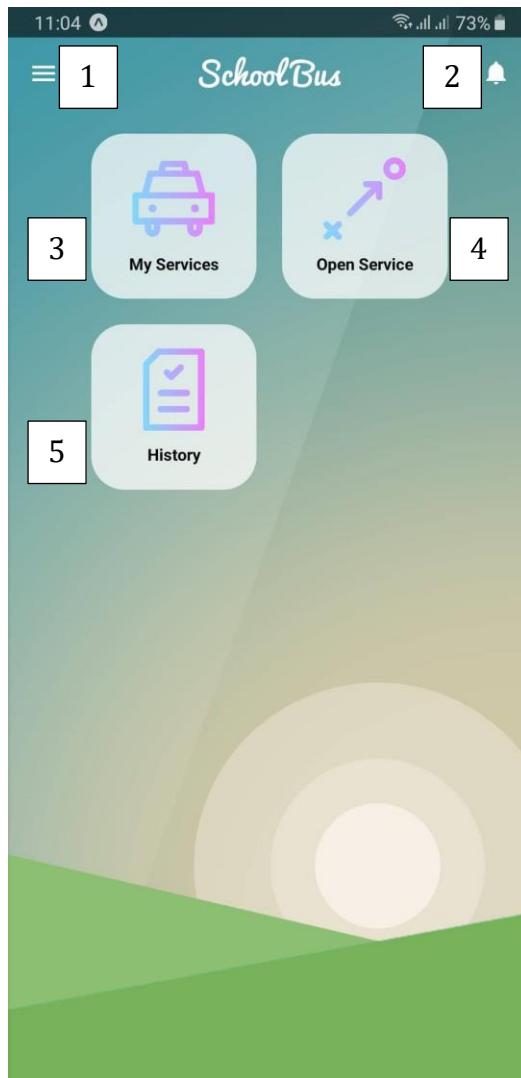
Fields

No	Field Name	Description	Read-only	Mandatory	Control Type	Data Type	Length
1	Background	App background for different time of day	Yes	Yes	ImageView	Image	

Buttons/Hyperlinks

No	Function	Description	Validation	Outcome
2	Open Side Bar	Toggle the side bar menu		
3	Notification	Open Notification page		
4	Requirements	Open Managing Requirements page		
5	Register requirement	Open Registering Requirement page		
6	Children	Open Managing Children page		
7	History	Open Trip History page		

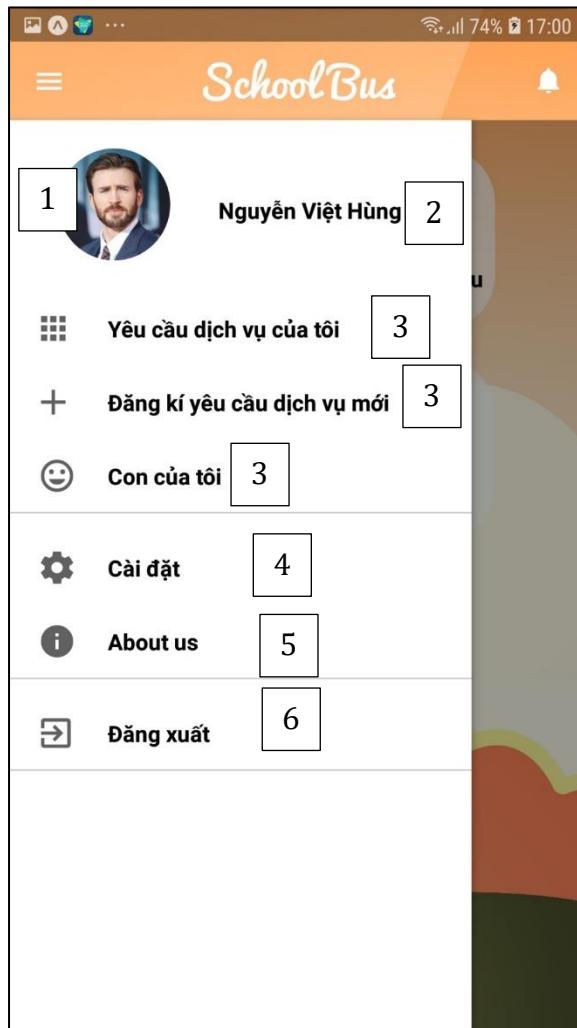
5.6 Home Screen - Driver



Buttons/Hyperlinks

No	Function	Description	Validation	Outcome
1	Open Side Bar	Toggle the side bar menu		
2	Notification	Open Notification page		
3	Services	Open Managing Services page		
4	Register service	Open Registering Service page		
5	History	Open Trip History page		

5.7 Home Screen - Side bar

**Fields**

No	Field Name	Description	Read-only	Mandatory	Control Type	Data Type	Length
1	Image	User's image	Yes	Yes	ImageButton	ImageURI	Greater than 0
2	Name	User's name	Yes	Yes	Textbox	String	Greater than 0

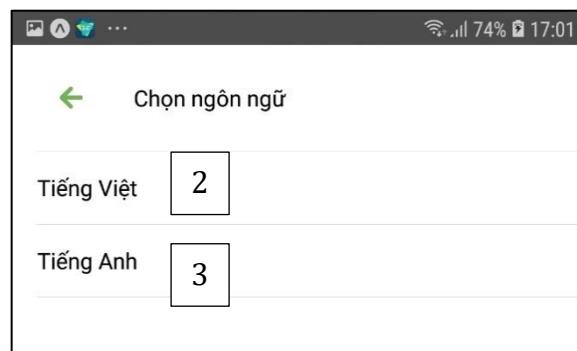
Buttons/Hyperlinks

No	Function	Description	Validation	Outcome
1	Go to Updating Profile	Go to Updating Profile screen	N/A	Redirect to the updating profile screen
3	Hyperlink	Open corresponding page		
4	Settings	Go to Application Settings		
5	About us	Go to Application information		
6	Log out	Log out account		

5.8 Settings Page

**Fields**

No	Field Name	Description	Read-only	Mandatory	Control Type	Data Type	Length
1	Current Language	Display current language	No	Yes		String	

**Buttons/Hyperlinks**

No	Function	Description	Validation	Outcome
2	Choose Vietnamese	Change language to Vietnamese	N/A	Redirect to the home screen
3	Choose English	Change language to English	N/A	Redirect to home screen

5.9 Logout Modal

**Buttons/Hyperlinks**

No	Function	Description	Validation	Outcome
1	Logout	Logout from the system	Agree to logout: - Yes - No	Redirect to the login screen

5.10 Customer - Add Child Page

The screenshot shows a mobile application interface for adding a child. The title bar says 'Thêm con'. The form fields are:

- Image of a child (labeled 1)
- Name: Trần Thị Ngọc Linh (labeled 2)
- Date of Birth: 13-04-2011 (labeled 3)
- Class: 9A (labeled 4)
- School: THPT Gia Định, Đường Võ Oanh, Phường 25, Bình Thạnh, Hồ Chí Minh, Việt Nam (labeled 5)
- Add Child button (labeled 6)

Fields

No	Field Name	Description	Read-only	Mandatory	Control Type	Data Type	Length
1	Image	Child's image	No	Yes	ImagePicker	ImageURI	Greater than 0
2	Name	Child's name	No	Yes	Textbox	String	Greater than 0
3	Birthday	Child's birthday	No	Yes	DatePicker	Date	N/A
4	Class	Child's class	No	Yes	Textbox	String	Greater than 0
5	School	Child's school	No	Yes	Place Autocomplete	String	Greater than 0

Buttons/Hyperlinks

No	Function	Description	Validation	Outcome
6	Add Child	Add new child	N/A	Redirect to the viewing all child's screen

5.11 Customer - Create Customer Requirement Page

The screenshot shows a mobile application interface for creating a customer requirement. Key elements include:

- Driver Selection:** A placeholder "Chọn con để đưa đón" (Select driver) with a placeholder image and name "Trần Thị Ngọc Linh".
- Time Selection:** A section "Chọn ngày đón con bạn trong tuần" (Select pick-up day of the week) with days T2-T7 highlighted.
- Date Selection:** Fields for "Thời gian đón" (Pick-up time), "Thời gian tới trường" (Arrival time at school), and "Thời gian quay về" (Return time from school), each with a time input and a calendar icon.
- Start Date:** "Ngày bắt đầu" (Start date) set to 14-04-2019.
- End Date:** "Ngày kết thúc" (End date) set to 14-04-2019.
- Location:** "Điểm đón" (Drop-off point) with address "213 Lê Thánh Tông, Thành, Quận 1, Hồ Chí Minh, Việt Nam" and a map pin icon.
- School:** "Trường" (School) listed as "THPT Gia Định, Đường Võ Oanh, Bình Thạnh, Hồ Chí Minh, Vietnam".
- Cost:** "Đơn giá (VND/người/ngày)" (Unit price) listed as "44.765 VND" and "Tổng giá" (Total price) listed as "0 VND".
- Action:** A green button "Đăng kí yêu cầu dịch vụ mới" (New service request registration).

Fields

No	Field Name	Description	Read-only	Mandatory	Control Type	Data Type	Length
1	Children	Children	No	Yes	ChildrenPicker	Array	Gt 0
2	Weekdays	Days in a week	No	Yes	WeekdaysPicker	Array	Gt 0
3	Pickup Time	Picking up time	No	Yes	TimePicker	Date	Gt 0
4	Arrival Time	Arrival time at school	No	Yes	TimePicker	Date	Gt 0
5	Return Time	Return time from school	No	Yes	TimePicker	Date	Gt 0
6	Start Date	Start day	No	Yes	DatePicker	Date	Gt 0
7	End Date	End day	No	Yes	DatePicker	Date	Gt 0
8	Pickup Address	Picking up address	No	Yes	Place Autocomplete	String	Gt 0
9	School	School	No	Yes	Place Autocomplete	String	Gt 0
10	Money	Money required	Yes	Yes	Text	String	

Buttons/Hyperlinks

No	Function	Description	Validation	Outcome
11	Add Requirement	Add new requirement	Return Time is greater than or equal to Arrival Time. Arrival Time is greater than or equal to picking up time. End date is greater than or equal to start date.	Redirect to the viewing driver's services screen

5.12 Customer - Create Contract Agreement Page

The screenshot shows a mobile application interface for selecting a driver. At the top, it displays the price per person per day (Đơn giá: 44.765 VND/người/ngày) and the total price (Tổng giá: 850.535 VND). Below this, there is a circular profile picture of a driver (labeled 1), his name (Nguyễn Văn Hậu) (labeled 2), and his contact number (0087654321) (labeled 3). Further down, it shows the car model (Toyota Innova Đen 6 chỗ) (labeled 4) and license plate number (Biển số xe ABC-123456) (labeled 5). A rating of five stars (labeled 6) is also displayed. At the bottom, there is a button labeled 'Chọn tài xế' (labeled 9). The entire screen is labeled 7 at the top right corner.

Fields

No	Field Name	Description	Read-only	Mandatory	Control Type	Data Type	Length
1	Driver Avatar	Driver's avatar	Yes	Yes	Image		
2	Driver Name	Driver's name	Yes	Yes	Text	String	Greater than 0
3	Phone Number	Driver's phone number	Yes	Yes	Text	String	Greater than 0
4	Car Info	Driver's car's information	Yes	Yes	Text	String	Greater than 0
5	Plate Number	Driver's car's plate number	Yes	Yes	Text	String	Greater than 0
6	Avg Score	Driver's average feedback score	Yes	Yes	Text	Double	Greater than 0
7	Unit Price	Price per seat	Yes	Yes	Text	Double	Greater than 0
8	Total Price	Total price	Yes	Yes	Text	Double	Greater than 0
9	Message	Message not found driver	Yes	Yes	Text	String	

Buttons/Hyperlinks

No	Function	Description	Validation	Outcome
10	Choose Service	Choose the driver's service for transporting	N/A	Redirect to the viewing requirement's detail screen

5.13 Customer - Requirement Detail Page (No contract)

The screenshot shows a mobile application interface for a school bus requirement. At the top, there's a header bar with icons for signal strength, battery level (80%), and time (17:58). Below the header, the title "Chi tiết yêu cầu" is displayed, with a back arrow icon, a number "10" in a box, and a more options icon.

The main content area contains the following elements:

- A message: "Chưa có hợp đồng" (No contract yet) with a yellow "Tim ngay" (Search now) button labeled "11".
- A user profile section with a circular photo of a woman, the name "Trần Thị Ngọc Linh", and a number "1".
- An address: "213 Lê Thánh Tôn, Bến Thành, Quận 1, Hồ Chí Minh, Việt Nam" with a location pin icon labeled "2".
- School information: "THPT Gia Định" with a building icon labeled "3".
- Shift details: "Chuyến đi: 18:45 - 19:15" with a bus icon labeled "4" and "Chuyển về: 21:00" with a bus icon labeled "5".
- Weekdays: "CN T2 T3 T4 T5 T6 T7" with a calendar icon labeled "6".
- Dates: "13-04-2019 -> 09-05-2019" with a calendar icon labeled "7".
- Icons along the left side: people (1), house (2), bus (3), bus (4), bus (5), calendar (6), and calendar (7).

Fields

No	Field Name	Description	Read-only	Mandatory	Control Type	Data Type	Length
1	Children	Children	Yes	Yes	ChildrenList	Array	Gt than 0
2	PickupAddress	Picking up address	Yes	Yes	Text	String	Gt than 0
3	School	School	Yes	Yes	Text	String	Gt than 0
4	PickupTime	Picking up time	Yes	Yes	Text	String	Gt than 0
5	ArrivalTime	Arrival time to school	Yes	Yes	Text	String	Gt than 0
6	ReturnTime	Return time from school	Yes	Yes	Text	String	Gt than 0
7	Weekdays	Days in week	Yes	Yes	Weekdays	String	Gt than 0
8	StartDate	Start day	Yes	Yes	Text	String	Gt than 0
9	EndDate	End day	Yes	Yes	Text	String	Gt than 0

Buttons/Hyperlinks

No	Function	Description	Validation	Outcome
10	Open Options	Options for current requirement	N/A	Show the options
11	Find driver	Go to find matching driver service screen	No contract available	Redirect to updating screen

5.14 Customer – Requirement Detail Page (Has contract)

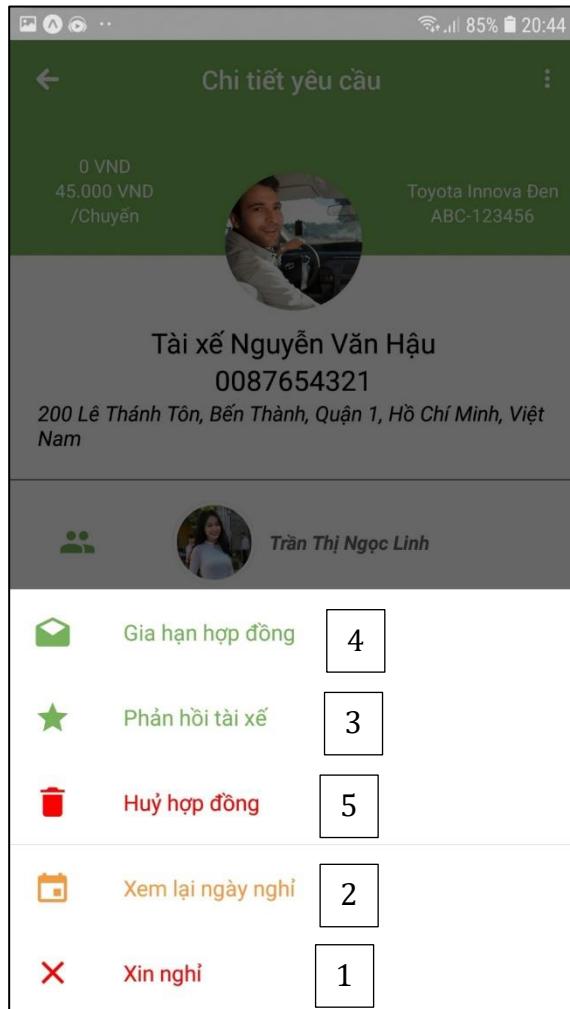


Fields

No	Field Name	Description	Read-only	Mandatory	Control Type	Data Type	Length
1	Driver Avatar	Driver's avatar	Yes	Yes	Image		
2	Driver Name	Driver's name	Yes	Yes	Text	String	Greater than 0
3	Phone Number	Driver's phone number	Yes	Yes	Text	String	Greater than 0
4	Car Info	Driver's car's information	Yes	Yes	Text	String	Greater than 0
5	Plate No	Driver's car's plate number	Yes	Yes	Text	String	Greater than 0
6	Address	Driver's address	Yes	Yes	Text	Double	Greater than 0
7	Unit Price	Price per seat	Yes	Yes	Text	Double	Greater than 0
8	Total Price	Total price	Yes	Yes	Text	Double	Greater than 0
9	Contract status	Different color for different status of contract	Yes	Yes			



5.15 Customer – Requirement Detail Page (Options)



Buttons/Hyperlinks

No	Function	Description	Validation	Outcome
1	Go to Cancelling Trip	Go to Cancelling Trip screen	There is an active contract	Redirect to the cancelling trip screen
2	Go to Reviewing Cancelled Trip	Go to reviewing cancelled trip screen	There is an active contract	Redirect to the reviewing cancelled trip screen
3	Go to Feedback	Go to sending feedback screen	There is an active contract	Redirect to the sending feedback screen
4	Go to Extending Contract	Go to extending contract screen	There is an active contract	Redirect to the extending contract screen
5	Go to Cancelling Contract	Go to cancelling contract screen	There is an active contract	Redirect to the cancelling contract screen

5.16 Customer - Extend Contract Page

The screenshot shows a mobile application interface. At the top, there is a header with a back arrow and the text 'Gia hạn dịch vụ'. Below this, a green button labeled 'Chọn ngày gia hạn hợp đồng' is displayed. To its right is a date input field showing '10-05-2019' with a calendar icon. Next to it is a text input field containing the number '1'. At the bottom, there is a green button labeled 'Gia hạn' and a white button labeled '2'.

Fields

No	Field Name	Description	Read-only	Mandatory	Control Type	Data Type	Length
1	ExtendingDate	New end day for contract	No	Yes	DatePicker	Date	Greater than 0

Buttons/Hyperlinks

No	Function	Description	Validation	Outcome
2	Extend Contract	Extend the expired time of the transporting contract	New end day must be greater than the old one.	Redirect to detail screen

5.17 Customer - Cancel Contract Page

The screenshot shows a mobile application interface. At the top, there is a header with a back arrow and the text 'Huỷ hợp đồng'. Below this, there are two dropdown menus. The first dropdown has the label 'Lí do huỷ' and contains the value '1'. The second dropdown has the label 'Lí do khác' and contains the value '2'. At the bottom, there is a green button labeled 'Huỷ hợp đồng' and a white button labeled '3'.

Fields

No	Field Name	Description	Read-only	Mandatory	Control Type	Data Type	Length
1	Description	Reason for cancelling contract	No	Yes	Picker	String	Greater than 0
2		(If there is another reason)	No	Yes	Textarea	String	Greater than 0

Buttons/Hyperlinks

No	Function	Description	Validation	Outcome
3	Cancel Contract	Cancel the current contract	N/A	Redirect to the viewing requirement's detail screen

5.18 Customer - Cancel Trip Page

Huỷ chuyến đi

Chọn con bạn muốn cho nghỉ

Trần Thị Ngọc Linh

Chọn ngày bạn muốn cho bé nghỉ

T2, 15-04-2019

Lí do Lý do khác

Xin nghỉ

Fields

No	Field Name	Description	Read-only	Mandatory	Control Type	Data Type	Length
1	Children	Your children that you want to request day-off	No	Yes	ChildrenPicker	Array	Greater than 0
2	OffDate	Children's off-day	No	Yes	DatePicker	Date	Greater than 0
3	Description	Reason for requesting off-day	No	Yes	Picker	String	Greater than 0
4		(If there is another reason)	No	No	Textarea	String	Greater than 0

Buttons/Hyperlinks

No	Function	Description	Validation	Outcome
5	Cancel Trip	Cancel your children's trip on the off-day	The off-day must be from the requirement's days in week	Redirect to the viewing requirement's detail screen

5.19 Customer - Get Cancellation Requests Page

Xem lại ngày nghỉ

Cửa Bạn

Trần Thị Ngọc Linh

Nghỉ vào 15-04-2019

Lí do: Con tôi đi trám răng!

OK

1 2 3 4

Fields

No	Field Name	Description	Read-only	Mandatory	Control Type	Data Type	Length
1	Children	Children which are off (customer's request)	Yes	Yes	ChildrenList	Array	Greater than 0
2	OffDate	Children's off-day / Driver's off-day	Yes	Yes	Text	Date	Greater than 0
3	Description	Reason for off (children/driver)	Yes	Yes	Text	String	Greater than 0

Buttons/Hyperlinks

Xem lại ngày nghỉ

Cửa Tài xế

Nghỉ vào 16-04-2019

Lí do: Về quê.

OK

2 3

No	Function	Description	Validation	Outcome
4	OK	Go back	N/A	Redirect to the viewing requirement's detail screen
5	Change to driver's off-days	View driver's off-days	N/A	Switch screen to viewing driver's off-days
6	Change to your off-days	View your off-days	N/A	Switch screen to viewing your off-days

5.20 Customer - Feedback Page

Phản hồi

Tài xế của con bạn

Nguyễn Văn Hậu
0087654321

1

Chạy cẩn thận. 2

Gửi phản hồi 3

Fields

No	Field Name	Description	Read-only	Mandatory	Control Type	Data Type	Length
1	Score	Feedback's score	No	Yes	StarRating	int	Greater than 0
2	Content	Feedback's content	No	No	Textarea	String	Greater than 0

Buttons/Hyperlinks

No	Function	Description	Validation	Outcome
3	Send Feedback	Send feedback to the drivers	N/A	Redirect to the viewing requirement's detail screen

5.21 Driver – Create Driver Service Page

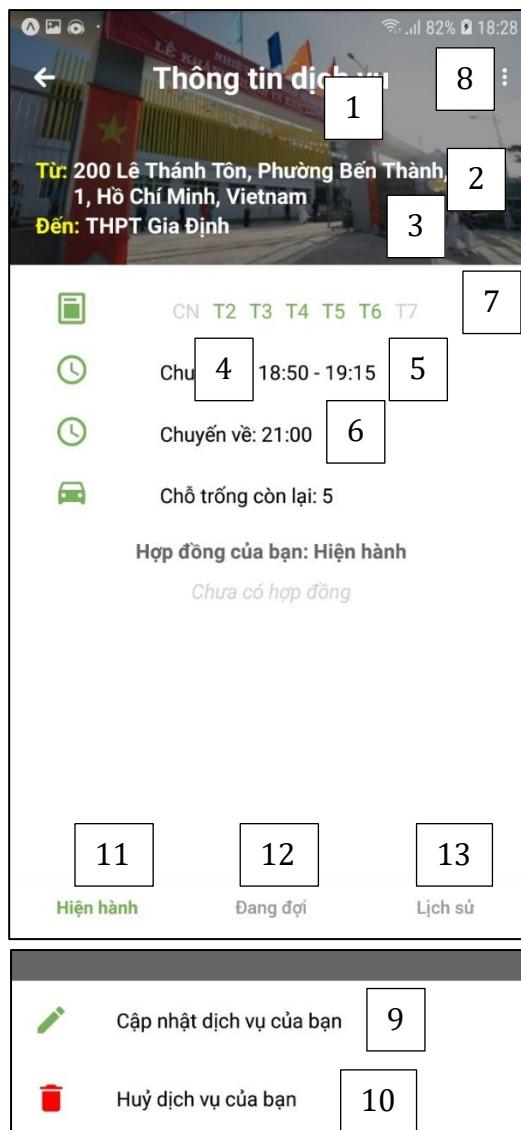
Fields

No	Field Name	Description	Read-only	Mandatory	Control Type	Data Type	Length
1	Weekdays	Days in a week	No	Yes	WeekdaysPicker	Array	Gt 0
2	Start Time	Start time	No	Yes	TimePicker	Date	Gt 0
3	Arrival Time	Arrival time at school	No	Yes	TimePicker	Date	Gt 0
4	Return Time	Return time from school	No	Yes	TimePicker	Date	Gt 0
5	Start Address	Start address	No	Yes	Place Autocomplete	String	Gt 0
6	School	School	No	Yes	Place Autocomplete	String	Gt 0
7	Class	Child class	No	Yes	Textbox	String	Gt 0
8	Capacity Available	Capacity in car	No	Yes	Textbox	Number	Gt 0, no gt car's capacity

Buttons/Hyperlinks

No	Function	Description	Validation	Outcome
9	Add Service	Add new service	Return Time is greater than or equal to Arrival Time. Arrival Time is greater than or equal to start time	Redirect to the driver's service detail screen

5.22 Driver – Service Detail Page (No Contract)



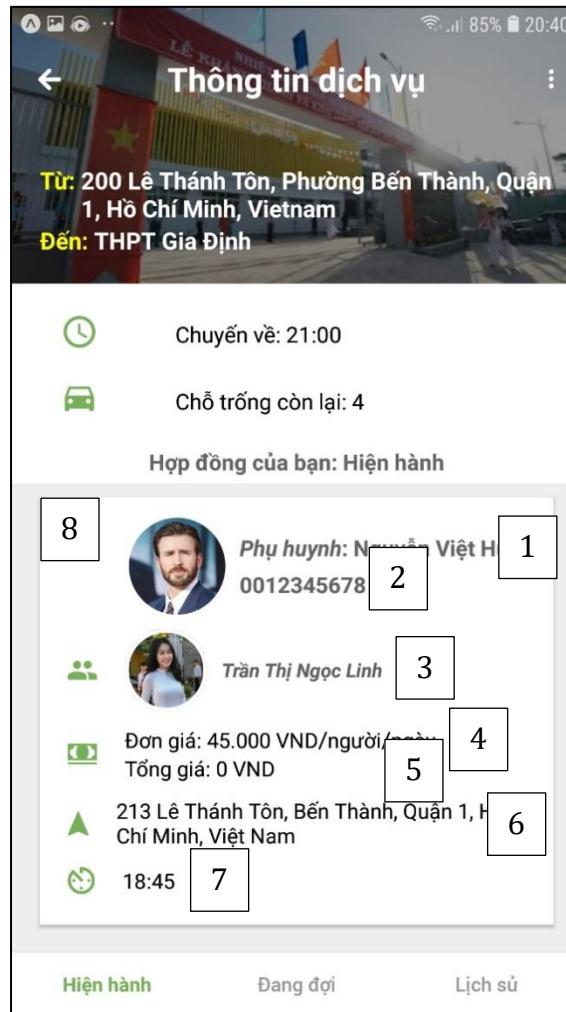
Fields

No	Field Name	Description	Read-only	Mandatory	Control Type	Data Type	Length
1	SchoolImage	School Image	Yes	Yes	ImageView		
2	StartAddress	Service's Start Address	Yes	Yes	Text	String	Greater than 0
3	School	Service's School	Yes	Yes	Text	String	Greater than 0
4	StartTime	Service's Start Time	Yes	Yes	Text	String	Greater than 0
5	ArrivalTime	Service's Arrival Time	Yes	Yes	Text	String	Greater than 0
6	ReturnTime	Service's Return Time	Yes	Yes	Text	String	Greater than 0
7	Weekdays	Service's days in week	Yes	Yes	Weekdays	String	Greater than 0

Buttons/Hyperlinks

No	Function	Description	Validation	Outcome
8	Open Options	Display options for current service	N/A	Show the options
9	Update Service	Go to Updating Service screen	No contract available	Redirect to the updating service screen
10	Delete Service	Delete the current service	No contract available	Redirect to the viewing services screen
11	Current Contract	Open current contracts tab		
12	Pending Contract	Open pending contracts tab		
13	History Contract	Open history contracts tab		

5.23 Driver – Service Detail Page (Has Contract)

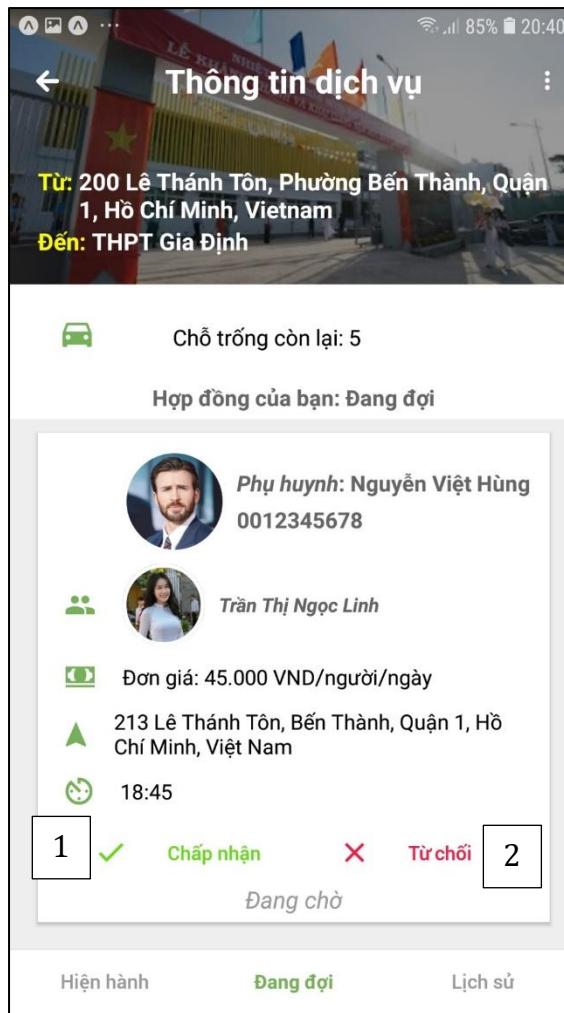
**Fields**

No	Field Name	Description	Read-only	Mandatory	Control Type	Data Type	Length
1	CustomerName	Customer's name	Yes	Yes	Text	String	Greater than 0
2	CustomerPhone	Customer's phone number	Yes	Yes	Yes	String	Greater than 0
3	Children	Customer's children	Yes	Yes	ChildrenList	Array	Greater than 0
4	UnitPrice	Price per seat	Yes	Yes	Yes	String	Greater than 0
5	TotalPrice	Total price	Yes	Yes	Yes	String	Greater than 0
6	Pickup Address	Pick Up Address	Yes	Yes	Yes	String	Greater than 0
7	Pickup Time	Pick Up Time	Yes	Yes	Yes	String	Greater than 0

Buttons/Hyperlinks

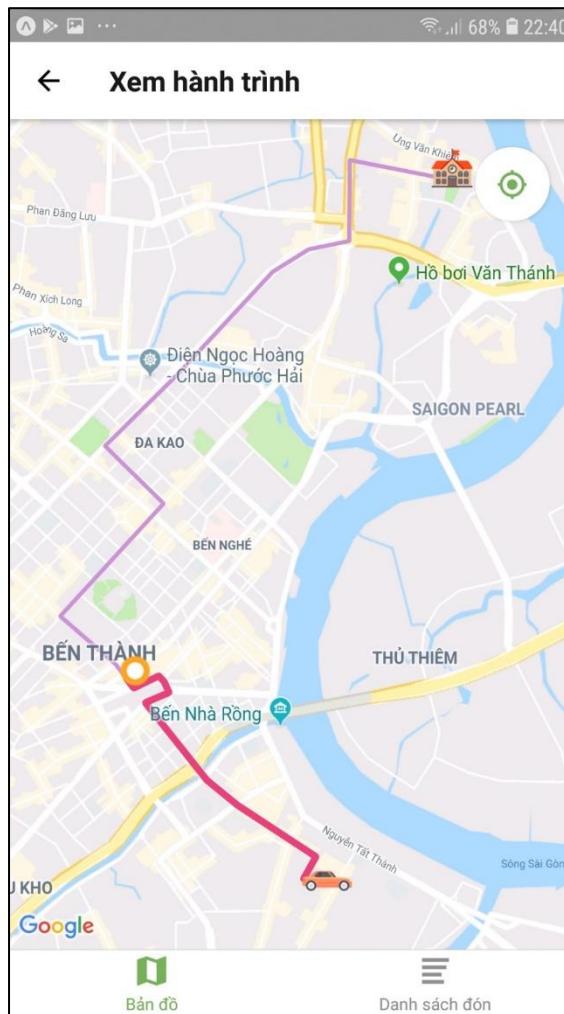
No	Function	Description	Validation	Outcome
8	Go to Contract Cancellation Screen	Go to Contract Cancellation screen	There are active contracts. Long press.	N/A

5.24 Driver – Service Detail Page (Has Pending Contract)

**Buttons/Hyperlinks**

No	Function	Description	Validation	Outcome
1	Accept	Accept pending contract		
2	Reject	Reject pending contract		

5.25 Driver – Daily Trip Page (Map View)

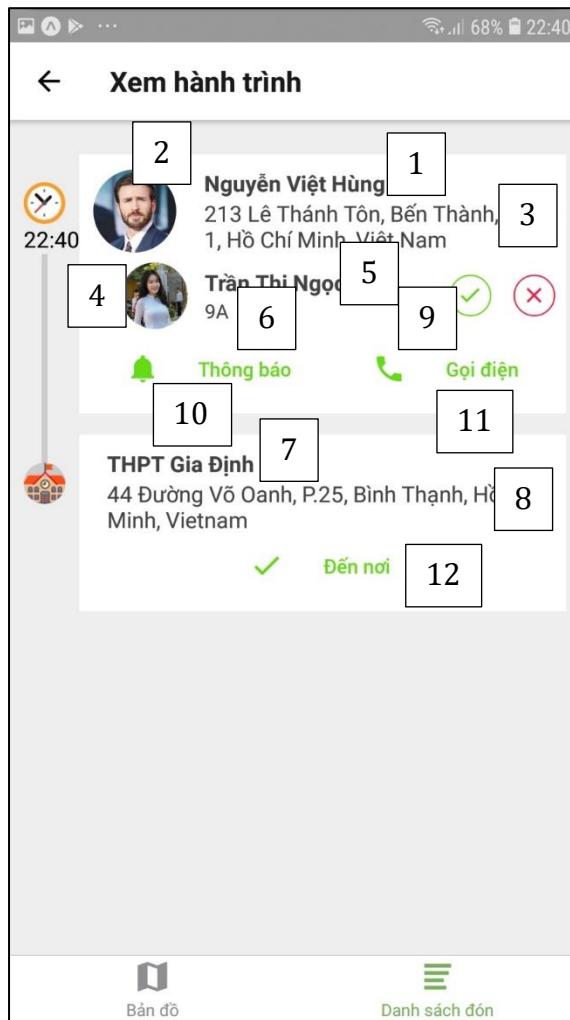
**Fields**

No	Field Name	Description	Read-only	Mandatory	Control Type	Data Type	Length
----	------------	-------------	-----------	-----------	--------------	-----------	--------

Buttons/Hyperlinks

No	Function	Description	Validation	Outcome
----	----------	-------------	------------	---------

5.26 Driver – Daily Trip Page (List View)

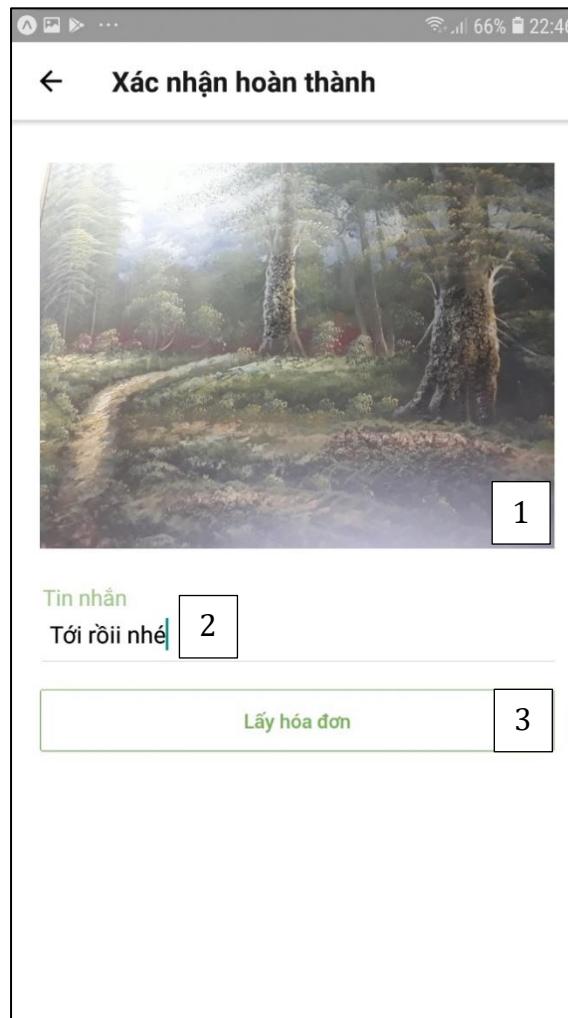
**Fields**

No	Field Name	Description	Read-only	Mandatory	Control Type	Data Type	Length
1	CustomerName	Customer's name	Yes	Yes	Text	String	Gt 0
2	CustomerImage	Customer's image	Yes	Yes	Image	ImageURI	Gt 0
3	PickupAddress	Picking up address	Yes	Yes	Text	String	Gt 0
4	ChildImage	Child's image	Yes	Yes	Image	ImageURI	Gt 0
5	ChildName	Child's name	Yes	Yes	Text	String	Gt 0
6	ChildClass	Child's class	Yes	Yes	Text	String	Gt 0
7	SchoolName	School's name	Yes	Yes	Text	String	Gt 0
8	SchoolAddress	School's address	Yes	Yes	Text	String	Gt 0

Buttons/Hyperlinks

No	Function	Description	Validation	Outcome
9	Confirm Attendance	Confirm child's attendance for trip	Trip is on-going	N/A
10	Notify Customer	Notify the customer that driver arrived	Trip is on-going	N/A
11	Call Customer	Call customer by phone	Trip is on-going	N/A
12	Confirm Arrival	Confirm arriving at school	Trip is on-going	N/A
13	Finish	Finish the trip	Trip is on-going	N/A

5.27 Driver – Verify Trip Page

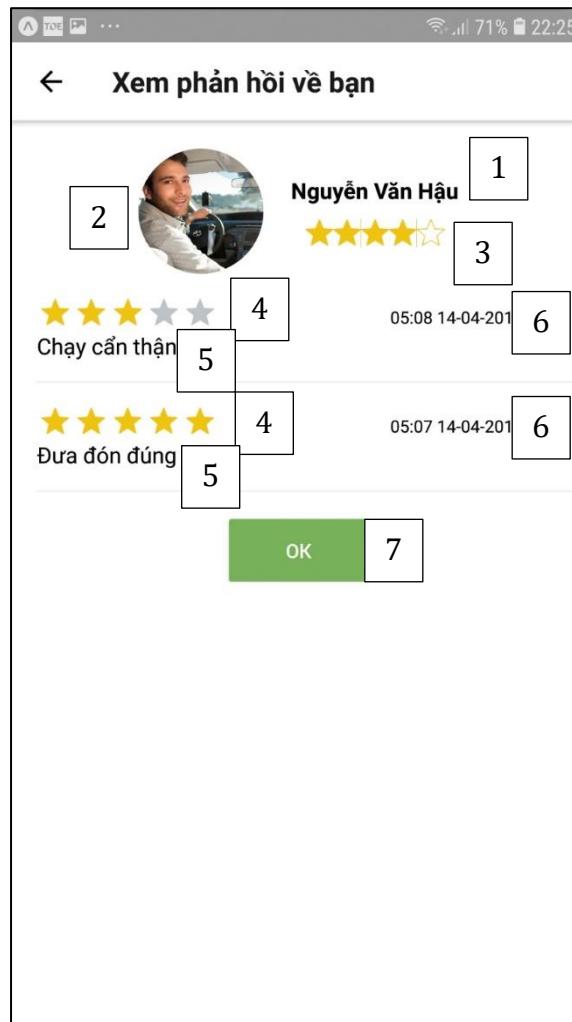
**Fields**

No	Field Name	Description	Read-only	Mandatory	Control Type	Data Type	Length
1	Image	Trip's evidence image	No	No	ImagePicker	ImageURI	
2	Description	Trip's evidence description	No	No	Textarea	String	

Buttons/Hyperlinks

No	Function	Description	Validation	Outcome
3	Verify	Confirm verification evidence for school trip	N/A	Redirect to the bill information screen

5.28 Driver – Get Feedbacks Page

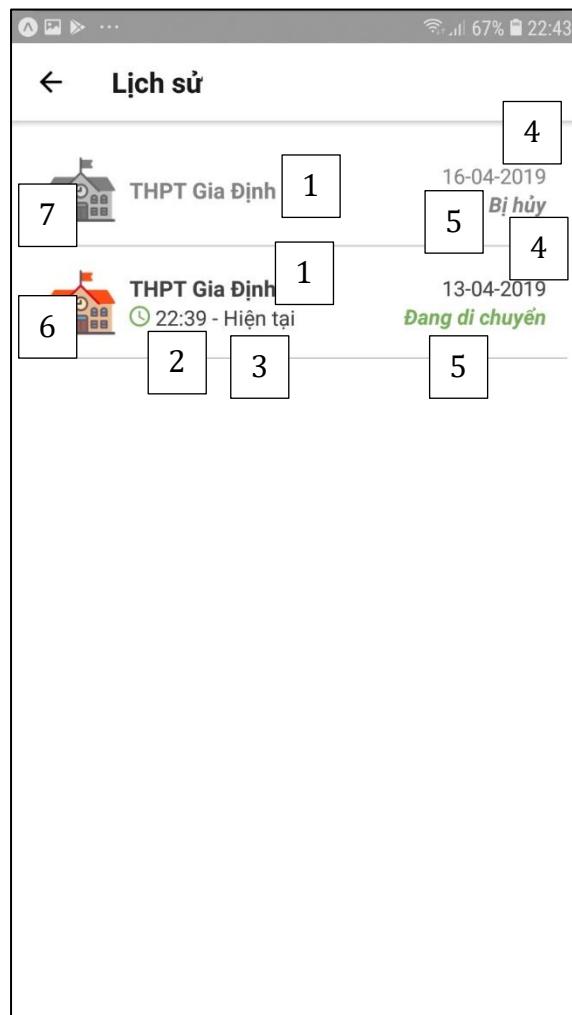
**Fields**

No	Field Name	Description	Read-only	Mandatory	Control Type	Data Type	Length
1	Name	Driver's name	Yes	Yes	Text	String	Greater than 0
2	Avatar	Driver's avatar	Yes	Yes	Text	String	Greater than 0
3	AvgScore	Driver's average feedback score	Yes	Yes	Text	String	Greater than 0
4	Score	Feedback's score	Yes	Yes	Text	String	Greater than 0
5	Content	Feedback's content					
6	Time	Feedback's created time					

Buttons/Hyperlinks

No	Function	Description	Validation	Outcome
7	Go back	Go back to previous screen	N/A	Redirect to previous screen.

5.29 Trip History Page



Fields

No	Field Name	Description	Read-only	Mandatory	Control Type	Data Type	Length
1	SchoolName	School's name	Yes	Yes	Text	String	Greater than 0
2	StartTime	Trip's start time	Yes	Yes	Text	String	Greater than 0
3	EndTime	Trip's end time	Yes	Yes	Text	String	Greater than 0
4	TripDate	Trip's day	Yes	Yes	Text	String	Greater than 0
5	Status	Trip's status	Yes	Yes	Text	String	Greater than 0

Buttons/Hyperlinks

No	Function	Description	Validation	Outcome
6	View Trip's Detail	Go to tracking driver screen	Trip is on-going	Redirect to the tracking driver screen
7	View bill's information	Go to viewing bill screen	Trip is finished	Redirect to the bill information screen

5.30 Bill Information Page

The screenshot shows a mobile application interface for a bill. At the top, it says "Hóa Đơn" (Bill). The header includes fields 1 (Trip Date: 14/03/2019), 2 (Arrival Time: 00:09), and 3 (Mã Chuyến: caoec730...). Below this is a photo of the driver (Nguyễn Văn) with ID 0087654321 and license ABC-123456, labeled 4. The driver's name is Nguyễn Văn. The trip details section includes "Chi tiết chuyến" (Trip details) with icons for start (8) and end (9), and "Bắt đầu" (Start) at 00:08. The destination is THPT Gia Định, 44 Đường Võ Oanh, P.25, Bình Thạnh, Hồ Chí Minh, Vietnam. The total amount is 10 000 VND, labeled 10. A button "Xem xác nhận của tài xế" (View driver's verification) is labeled 11. At the bottom, a "Tiếp tục" (Continue) button is labeled 12.

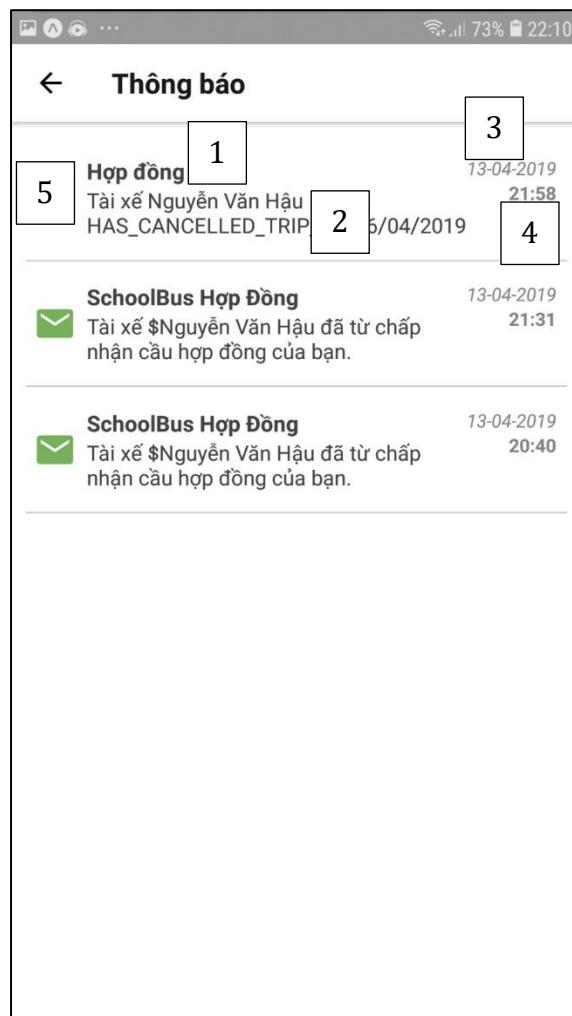
Fields

No	Field Name	Description	Read-only	Mandatory	Control Type	Data Type	Length
1	TripDate	Trip's day	Yes	Yes	Text	String	Gt 0
2	ArrivalTime	Trip's arrival time	Yes	Yes	Text	String	Gt 0
3	TripID	Trip's id	Yes	Yes	Text	String	Gt 0
4	DriverName	Driver's name	Yes	Yes	Text	String	Gt 0
5	DriveAvatar	Driver's avatar	Yes	Yes	Text	String	Gt 0
6	DriverPhone	Driver's phone number	Yes	Yes	Text	String	Gt 0
7	PlateNo	Driver's car's plate No	Yes	Yes	Text	String	Gt 0
8	Pickup Time	Trip's Pickup Time	Yes	Yes	Text	String	Gt 0
9	School	Children's school	Yes	Yes	Text	String	Gt 0
10	TotalPrice	Total price	Yes	Yes	Text	String	Gt 0

Buttons/Hyperlinks

No	Function	Description	Validation	Outcome
11	View Driver's Verification	Go to viewing driver's verification for current trip	N/A	Redirect to viewing driver's verification screen
12	Continue	Go back	N/A	Redirect to viewing trip history screen

5.31 Notifications Page

**Fields**

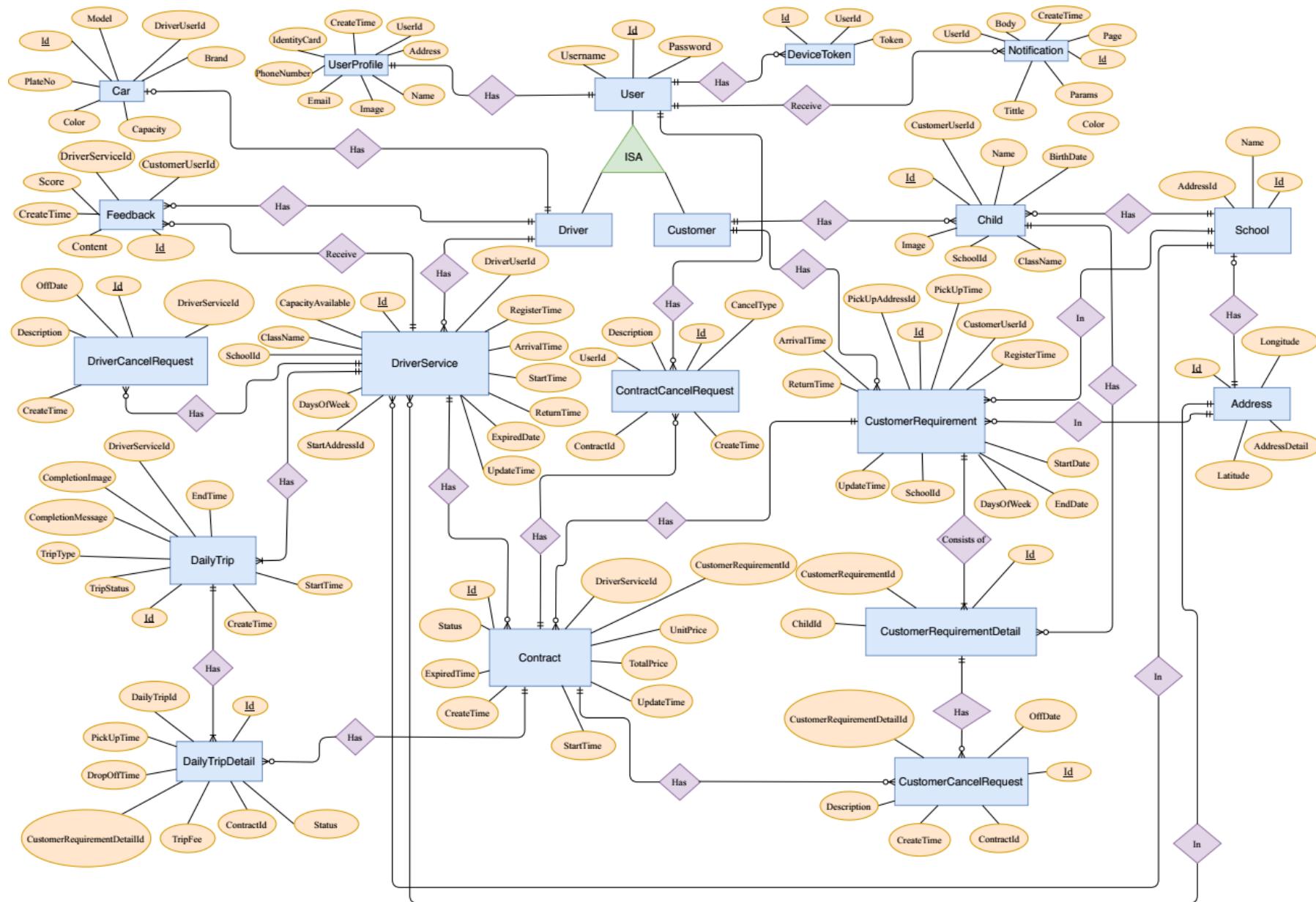
No	Field Name	Description	Read-only	Mandatory	Control Type	Data Type	Length
1	Title	Notification's title	Yes	Yes	Text	String	Greater than 0
2	Content	Notification's content	Yes	Yes	Text	String	Greater than 0
3	Date	Notification's day	Yes	Yes	Text	String	Greater than 0
4	Time	Notification's time	Yes	Yes	Text	String	Greater than 0

Buttons/Hyperlinks

No	Function	Description	Validation	Outcome
5	Go to Screen	Each notification would redirect to appropriate screens	N/A	Redirect to the appropriate screens

6. Database Design

6.1 Entity Relational Diagram



Entity Data Dictionary: describe content of all entities	
<u>Entity name</u>	<u>Description</u>
User	Contains the login's information.
User profile	Contain user's information.
DeviceToken	Contain user's devices token.
Car	Contain car's information of driver.
Feedback	Contain customer feedback for driver's service.
Notification	Contain notification of user.
Child	Contain child's information of customer.
DriverService	Contain Driver Service's information.
Contract	Contain Contract's information. Contract links CustomerRequirement with DriverService.
DriverCancelRequest	Contain cancel request of driver.
DailyTrip	Contain information for a trip.
DailyTripDetail	Contain daily trip information for a child.
ContractCancelrequest	Contain contract cancel request for driver and customer.
CustomerRequirement	Contain customer requirement for matching Driver Service.
CustomerRequirementDetail	Contain inforamtions for each child in a CustomerRequirement.
CustomerCancelRequest	Contain cancel request for a trip of customer.
School	Contain school information.
Address	Contain detail information for School and other Address.

Entity Name	Attributes	Description	Domain	Null
CustomerRequirement	<u>CustomerRequirementID</u>	ID of requirement	Guid	No
	UserID	ID of owner	Guid	No
	DaysOfWeek	String contain all day of requirement	Nvarchar(max)	No
	PickUpAddressID	ID of address	Guid	No
	SchoolID	ID of school	Guid	No
	PickUpTime	Time to pick children	DateTime	No
	ArrivalTime	Time required to go to school	DateTime	No
	ReturnTime	Time to pick up child after school	DateTime	No
	StartDate	The day customer wants to be served	DateTime	No
	EndDate	Day when customer want to stop the service	DateTime	No
	RegisterTime	Time when the Requirement is registered	DateTime	No
	UpdateTime	Latest time when the Requirement is updated	DateTime	No
	Status	Status of requirement	Bit	No
Contract	<u>ContractID</u>	Id of contract	Guid	No
	CustomerRequirementID	ID of customer requirement	Guid	No
	DriverServiceID	ID of driver service	Guid	No
	UnitPrice	Fee for a child	Decimal(18,2)	Yes
	TotalPrice	Total fee of the contract	Decimal(18,2)	Yes
	StartTime	Time when the contract is activated	DateTime	Yes
	ExpiredTime	Time when the contract is expired	DateTime	Yes
	CreateTime	Time when the contract is created	DateTime	No
	UpdateTime	Lasted time when the contract is updated	DateTime	No
	Status	Status of the contract	nvarchar(max)	No
DailyTrip	<u>DailyTripID</u>	ID of the trip	Guid	no
	DriverServiceID	ID of the service to be served	Guid	No
	StartTime	Time when the trip is started	DateTime	No
	EndTime	Time when the trip is finished	DateTime	Yes
	CreateTime	Time when the trip is created	DateTime	No
	Status	Status of the trip	Nvarchar(max)	No
	Type	Trip' type	Nvarchar(max)	No
	CompletionImage	Link to confirm image	Nvarchar(max)	Yes
	CompletionMessage	Completion message	Nvarchar(max)	Yes
UserProfile	<u>UserID</u>	Id of profile	Guid	No

	IdentityCard	Identity card number	Nvarchar(max)	No
	Name	User's name	Nvarchar(max)	No
	PhoneNumber	User's phone number	Nvarchar(max)	No
	Email	User's email	Nvarchar(max)	yes
	Address	User's address	Nvarchar(max)	Yes
	Image	Link to user's avatar	Nvarchar(max)	Yes
	CreateDate	Time when user account was created	DateTime	No
	IsActive	This account is activated or not	Bit	No
DailyTripDetail	<u>DailyTripDetailID</u>	Id of the trip detail	Guid	No
	DailyTripID	ID of the original trip	Guid	No
	ContractID	Id of the original contract	Guid	No
	CustomerRequirementDetailID	Id of the requirement detail	Guid	No
	PickUpTime	Time when driver picked up the child in the trip	DateTime	Yes
	DropOffTime	Time when driver dropped the child	DateTime	Yes
	TripFee	Fee of the trip	Decimal(18,2)	No
	status	Status of the trip	Nvarchar(max)	No
Children	<u>ChildID</u>	Id of the child	Guid	No
	UserID	Parent ID	Guid	No
	Name	Name of the child	Nvarchar(max)	No
	BirthDate	Date of birth of the child	DateTime	No
	SchoolID	Id of child's school	Guid	No
	ClassName	Nvarchar(max)	Class of the child	No
	Image	Link to child image	Nvarchar(max)	No
	IsActive	Is this child inactive or not	bit	No
Car	<u>CarID</u>	Id of the car	Guid	No
	PlateNo	Plate number of the car	Nvarchar(max)	No
	UserID	ID of the owner	Guid	No
	Brand	Car's brand	Nvarchar(max)	No
	Model	Car's model	Nvarchar(max)	No
	Capacity	Car's empty slot	int	No
	Color	Car's color	Nvarchar(max)	No
	IsActive	Is this car inactive or not	Bit	No
Notification	<u>NotificationID</u>	ID of the notification	Guid	No
	UserID	Id of the owner	Guid	No
	Title	Notification's title	Nvarchar(max)	No
	Body	Notification's body	Nvarchar(max)	No
	Time	Time when notification was pushed	DateTime	No
	Page	Todo action	Nvarchar(max)	Yes
	Params	Params for the action	Nvarchar(max)	Yes
CustomerCancelRequest	<u>CustomerCancelRequestID</u>	Id of the request	Guid	No

	<u>CustomerRequirementDetailID</u>	ID of the detail that need to be cancelled	Guid	No
	<u>ContractID</u>	Id of the contract	Guid	No
	<u>OffDate</u>	Day to be cancelled	DateTime	No
	<u>CreateTime</u>	Time when request was created	DateTime	No
	<u>Description</u>	Addition information for the request	Nvarchar(max)	No
<u>ContractCancelRequest</u>	<u>ContractCancelRequestID</u>	ID of the request	Guid	No
	<u>ContractID</u>	Id of the request to be cancelled	Guid	No
	<u>UserID</u>	Id of the request's owner	Guid	No
	<u>CreateTime</u>	Time when the request was created	DateTime	No
	<u>Description</u>	Cancel message	Nvarchar(max)	No
	<u>CancelType</u>	Type of the cancel	Nvarchar(max)	No
<u>Feedback</u>	<u>FeedbackID</u>	Id of the feedback	Guid	No
	<u>DriverServiceID</u>	Id of the service that the feedback is belong to	Guid	No
	<u>UserID</u>	Feedback owner	Guid	No
	<u>Score</u>	Score	Int	No
	<u>Content</u>	Feedback in message	Nvarchar(max)	Yes
	<u>CreateTime</u>	Time when the feedback was created	DateTime	No
<u>DriverCancelRequest</u>	<u>DriverCancelRequestID</u>	Id of the request	Guid	No
	<u>DriverServiceID</u>	Id of the service	Guid	No
	<u>OffDay</u>	Day to be cancelled	DateTime	No
	<u>CreateTime</u>	Time when the request was created	DateTime	No
	<u>Description</u>	Cancel message	Nvarchar(max)	No
<u>Address</u>	<u>AddressID</u>	Id of the address	Guid	No
	<u>AddressDetail</u>	Name of the address	Nvarchar(max)	No
	<u>Latitude</u>	Latitude	Float	No
	<u>Longitude</u>	Longitude	float	No
<u>UserAccount</u>	<u>Username</u>	User nick name	Nvarchar(128)	No
	<u>UserID</u>	Id of the account	Guid	No
	<u>Password</u>	User password	Nvarchar(max)	No
	<u>RoleID</u>	User role	int	No
<u>CustomerRequirementDetail</u>	<u>CustomerRequirementDetailID</u>	Id of the detail	Guid	No
	<u>CustomerRequirementID</u>	ID of the original requirement	Guid	No
	<u>ChildID</u>	Id of the child linked to this detail	Guid	No
<u>DeviceToken</u>	<u>DeviceTokenID</u>	ID of the token	Guid	No
	<u>UserID</u>	Id of the owner	Guid	No
	<u>Token</u>	Token string	Nvarchar(max)	No
<u>School</u>	<u>SchoolID</u>	Id of the school	Guid	No
	<u>Name</u>	School name	Nvarchar(max)	No

	AddressID	ID of the detail linked to the school	Guid	No
Role	RoleID	Id of the role	int	No
	RoleName	Role name	Nvarchar(max)	no

7. Algorithms

7.1 Find matching Driver Service and Customer Requirement

To find the appropriate driver's services for customers, we suggest the following matching algorithm which goes through 4 following steps:

- Firstly, the school from driver's service and customer's requirement must be matched; otherwise, there's no matching between driver's service and customer's requirement.
- Secondly, the distance from customer's requirement for pick-up address must be within configured radius of driver service's starting address; or else, there's no matching between driver's service and customer's requirement.
- Thirdly, the pick-up time of customer's requirement must be before or after a configured moment from driver's service's start time; or else, there's no matching between driver's service and customer's requirement.
- Finally, there's a matching rate between the days in week from customer's requirement and the days in week from driver's service. If the registered days are matched, a matching rate is calculated by the available days of the driver divided by the total days in the requirement of customer, considering that the available day is checked by comparing the available seats with the number of children required by customer in that day. If this rate is greater than or equal to a configured rate, the customer requirement would match the driver's service; or else, there's no matching between driver's service and customer's requirement.

Complexity: $(2n)^2$

Pseudocode:

Input the requirement

Call services

Initialize counter to zero

Initialize 'matching_school' services

While counter is less than length of services

If requirement's school is service's school

 Add service to 'matching_school' services

 Add one to counter

Set counter to zero

Initialize 'matching_distance' services

Initialize configured distance

While counter is less than length of 'matching_school' services

If requirement's picking up address is not far from service's start address over configured distance

 Add service to 'matching_distance' services

Add one to counter

Set counter to zero

Initialize 'matching_time' services

Initialize configured time

While counter is less than length of 'matching_distance' services

If requirement's picking up time is before or after service's start time for within a configured time

 Add service to 'matching_time' services

Add one to counter

Set counter to zero

Initialize 'matching_days_of_week_and_capacity' services

Initialize configured rate

Initialize service-percentage to zero

Initialize adding-percentage to 100 percent over numbers of requirement's days in week

While counter is less than length of 'matching_time' services

If requirement's day in week is the same as service's day in week

 If service's available capacity is greater than or equal to requirement's number of children

 Add adding-percentage to service-percentage

Add one to counter

Set counter to zero

Return 'matching_days_of_week_and_capacity' services

7.2 Find shortest routes between multiple stops (based on Traveling Salesman)

Situation: In our system, the driver has to go to multiple places to pick up kids before going to school. Therefore, the system has a responsibility to provide suitable routes for driver to help him/her to travel easier.

Input: The starting point, the end point and all of the stop points during the trip, each point must have latitude and longitude.

Output: A sequence of points, from starting point, to every of stop points and to end point.

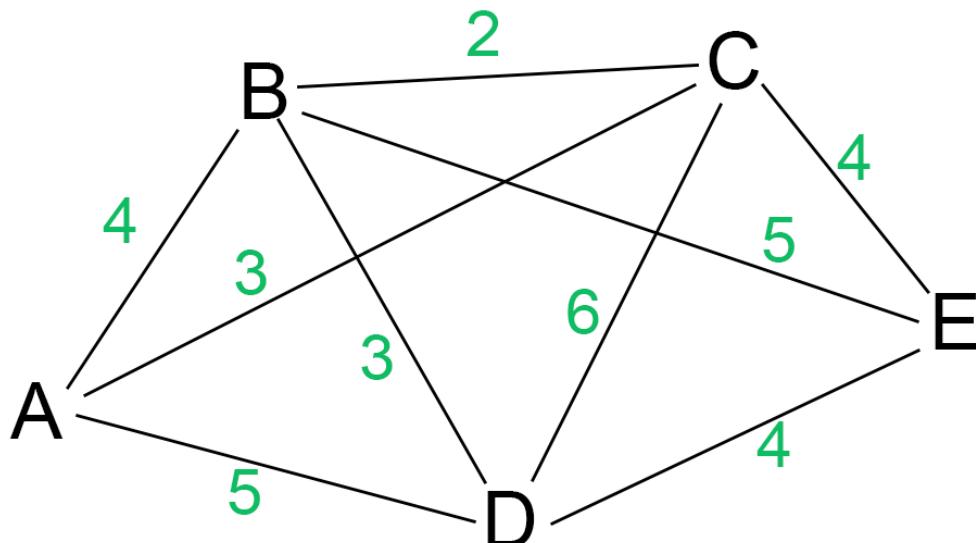
Implementation:

- **Customized Dijkstra:**

We use Dijkstra algorithm to find the shortest way from starting point to end point. However, one limitation of this algorithm is that it only supports the shortest one without visiting every required point. We enhance this simply by adding a step to check if the current route has visited all of stop points yet.

To get started, the distances between all pairs of points must be calculated (except the distance between starting point and end point directly). These distances can be easy to get since the latitude and longitude are provided.

For example, we have a route from point A to point E, which has to visit point B, C, D. The below figure shows the distances among them.



We make a queue to contain some possible route. At first, the queue only has one element:

[A]

Then, take the first element of the queue, and find the next stop. For example, in this case, from A, we have 3 possible routes that is AB – 4, AC – 3 and AD – 5. Push all of them to queue after sorting:

[AC – 3, AB – 4, AD – 5]

Repeat previous step until done. However, in each route, we do not need to travel to visited points and we do not need to travel to stop point as well if all of stop points are not visited.

To be specific, in this case, from route AC – 3, we can generate ACB – 5 and ACD – 9, the queue now is:

[AB – 4, AD – 5, ACB – 5, ACD – 9]

Similar to that, the next queue is:

[AD – 5, ACB – 5, ABC – 6, ABD – 7, ACD – 9]

Then:

[ACB – 5, ABC – 6, ABD – 7, ADB – 8, ACD – 9, ADC – 11]

[ABC – 6, ABD – 7, ADB – 8, ACBD – 8, ACD – 9, ADC – 11]

[ABD – 7, ADB – 8, ACBD – 8, ACD – 9, ADC – 11, ABCD – 12]

[ADB – 8, ACBD – 8, ACD – 9, ADC – 11, ABCD – 12, ABDC – 13]

[ACBD – 8, ACD – 9, ADBC – 10, ADC – 11, ABCD – 12, ABDC – 13]

From here, we can find one of the shortest routes from ACBD – 8, that is ACBDE – 12. However, we need to checkout if any route is shorter than 12.

[ADBC – 10, ADC – 11, ACDB – 12, ABCD – 12, ABDC – 13]

We found another route, that is ADBCE – 14 from ADBC – 10. However, it is greater than ACBDE – 12.

[ACDB – 12, ABCD – 12, ABDC – 13, ADCB – 13]

Now we realize that all of elements in queue is not less than 12. So, the shortest route is ACBDE – 12.

With this algorithm, we definitely find the shortest route to solve the problem. However, this is the complete graph, the complexity for this solution is really costly. In the worst case, all of the available routes must be checked out to find the best one. Though, if we have n stops, we must travel $n!$ possible routes, then, this algorithm shouldn't be used in this case.

- ***Greedy algorithm:***

This algorithm can be stated simply as: Finding the nearest point from previous point and push to the list. Of course, the end point shouldn't be chosen if all of stop points are not visited.

This algorithm is fast enough, though it would not be the best route in many cases. However, the output is still usable.

In conclusion, we find two possible algorithms to apply in our problem. The first one has the best result, but it is slow in performance. On the contrary, the second one is fast but not effective. In real practice, we will apply the first one for the case there are no greater than 7 stops, and the second one for the rest. We choose 7 because $7!$ is 5040, so in the worst case, 5000 is acceptable. Moreover, our system expects drivers as people owning private cars, and the maximum slots for these cars are 7.

E. System Implementation & Test

1. Introduction

1.1 Overview

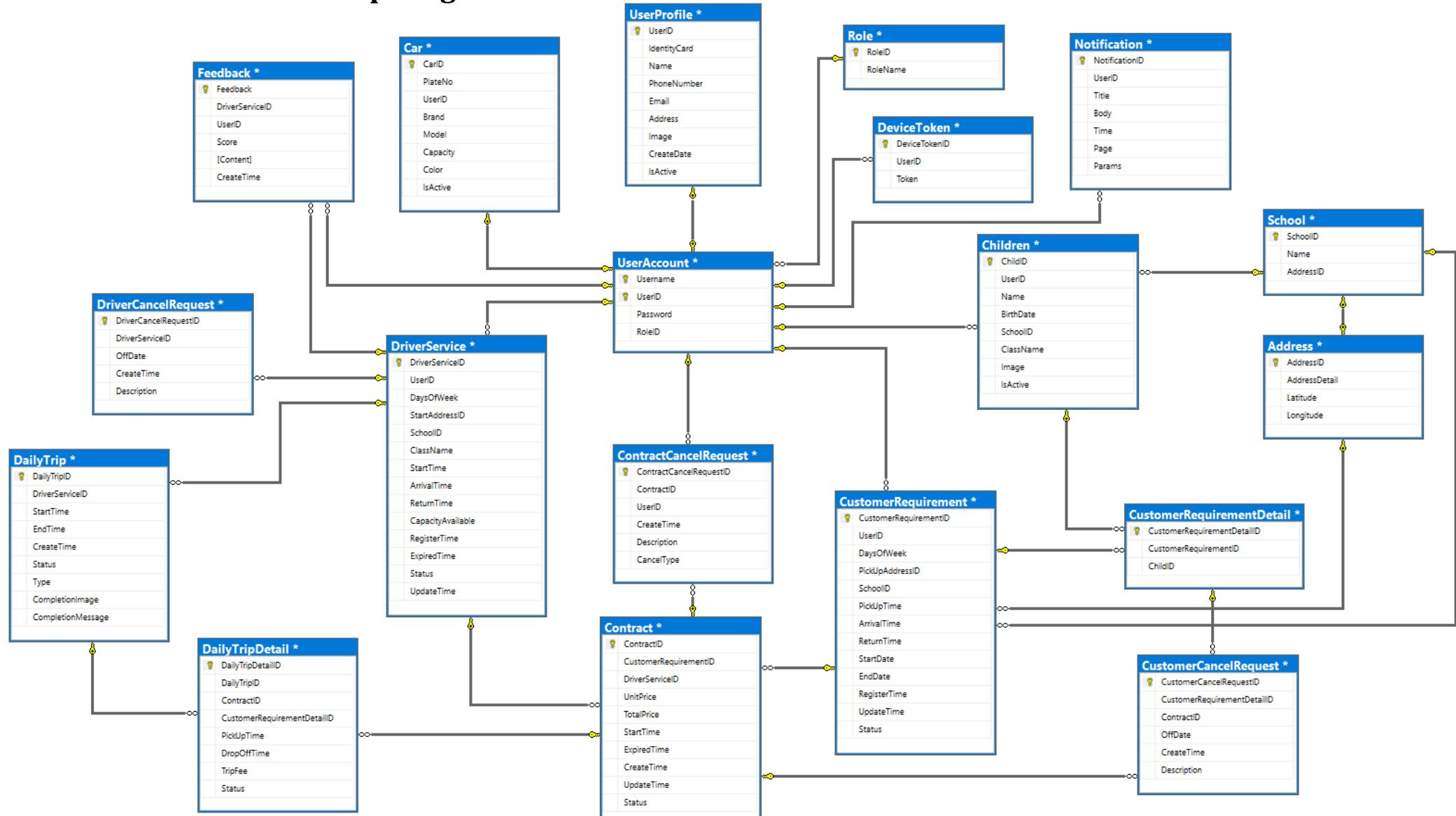
This section describes approach and methodologies used by our team to plan, organize and manage testing process of SCB.

It provides all necessary information about implementation and testing procedure of the system including test plans, test cases, test results and pass/fail criteria as well as testing flow to cover all possible cases.

1.2 Test Approach

- **Goal:** Test main features in the whole WCS based on core flow
- **Method:** Black-box testing

2. Data Relationship Diagram



Entity Data Dictionary: Describe content of all tables		
No.	Table Name	Description
1	UserAccount	Contain account information.
2	UserProfile	Contain user profile.
3	Car	Contain car information of driver.
4	Role	Role for users.
5	DeviceToken	Device unique token for pushing notifications.
6	Notification	Contain user's notifications.
7	Children	Contain children information.
8	School	Contain school information.
9	Address	Contain address information.
10	Feedback	Contain user's feedbacks for driver.
11	CustomerRequirement	Contain customer requirement for a service.
12	CustomerRequirementDetail	Contain customer requirement detail.
12	DriverService	Contain driver service information.
13	Contract	Contain contract made by driver service and customer requirement.
14	DailyTrip	Contain daily trip of driver service.
15	DailyTripDetail	Contain daily trip detail information.
17	ContractCancelRequest	Contain requests for canceling a contract.
18	CustomerCancelRequest	Contain customer requests for canceling a daily trip in contract.
19	DriverCancelRequest	Contain driver requests for canceling a daily trip in contract.

3. Test plan

The overall purpose of testing is to ensure SCB meets its entire technical, functional and business requirement. The purpose of this document is to describe the

test plan and strategy for testing the SCB. The following part will describe which features to be tested and which will not.

3.1 Features to be tested

Driver:

- Register account.
- Login.
- Update information.
- Manage services.
- Accept/Reject customer's pending contract agreement.
- Start trip.
- Notify customer.
- Pick up/Skip child.
- Finish trip.
- Get bill.
- Cancel a trip.
- Cancel contract.
- Get feedbacks.

Customer:

- Register account.
- Login.
- Update information.
- Manage requirements.
- Manage children.
- Find matching driver services.
- Create contract agreement.
- Track driver trip.
- Get bill.
- Cancel contract.
- Cancel pending contract agreement.
- Cancel a trip.

- Extend contract.
- Feedback driver.

3.2 Features not to be tested

Administrator:

- Manage users.
- Configure systems.
- Get statistics.

4. System testing Test case

4.1 Driver Test Case

ID	Test Case Description	Test Case Procedure	Expected Output	Inter Dependence Test Case	Result	Test Date	Note
SCB_TC_1	Driver creates an empty account	1. Driver selects “Tạo tài khoản mới”. 2. Driver leaves any field blank. 3. Driver touches “Đăng ký”.	Application Alert about missing information.	N/A	pass	2019/19/04	
SCB_TC_2	Driver creates a duplicated account	1. Driver selects “Tạo tài khoản mới”. 2. Driver inputs all fields. 3. Driver touches “Đăng ký”.	Application show notification “Tài khoản đã tồn tại”.	N/A	pass	2019/19/04	
SCB_TC_4	Driver creates a valid account	1. Driver selects “Tạo tài khoản mới”. 2. Driver inputs all fields. 3. Driver touches “Đăng ký”.	Application show notifies “Đăng ký tài khoản thành công” and shows “Update account” screen.	N/A	pass	2019/19/04	
SCB_TC_5	Driver Updates account without image	1. Driver inputs all fields but selecting image. 2. Driver touches “Kết thúc”.	Application alert “Tài xế bắt buộc phải có ảnh”.	N/A	pass	2019/19/04	
SCB_TC_6	Driver updates his/her car's information	1. Driver inputs all fields. 2. Driver touches “Cập nhật thông tin xe”	Application notifies “Đã cập nhật thông tin xe” and shows “Home” screen.	N/A	pass	2019/19/04	

SCB_TC_7	Driver creates new service.	<ol style="list-style-type: none"> 1. Driver touches “Mở dịch vụ” in “Home” screen. 2. Driver inputs all required fields. 3. Driver touches “Mở dịch vụ”. 	Application notifies “Mở dịch vụ thành công” and show “Services” screen.	N/A	pass	2019/19/04	
SCB_TC_8	Driver Accepts a requirement	<ol style="list-style-type: none"> 1. Driver opens “Requirement detail” screen. 2. Driver touches “Chấp nhận”. 	Application shows “Contract detail” screen. If any Pending requirement remains then rejecting all of them automatically.	N/A	pass	2019/19/04	
SCB_TC_9	Driver Rejects a requirement	<ol style="list-style-type: none"> 1. Driver opens “Requirement detail” screen. 2. Driver touches “Tù chối”. 	Back to the previous screen.	N/A	pass	2019/19/04	
SCB_TC_10	Driver starts a trip which is not on a valid time.	<ol style="list-style-type: none"> 1. Driver opens “Contracts” screen. 2. Driver touches “Bắt đầu chuyến đi” on a trip which is not on a valid time. 	Application notifies “Chuyến đi chưa đến giờ bắt đầu”.	N/A	pass	2019/19/04	
SCB_TC_11	Driver starts a valid trip.	<ol style="list-style-type: none"> 1. Driver opens “Contracts” screen. 2. Driver touches “Bắt đầu chuyến đi” on a trip which is on valid time 	<ul style="list-style-type: none"> - Application opens “Driving screen” and sends notification to customer. - All cancelled children is not included in the pickup list. - Recommend trip is drawn in the map. 	N/A	pass	2019/19/04	
SCB_TC_12	Driver notifies customer on arrival	1. Driver touches “Thông báo”.	Respective customer's phone receives notification.	N/A	pass	2019/19/04	

SCB_TC_13	Driver picks a child.	1. Driver touches tick-shaped button.	<ul style="list-style-type: none"> - Daily trip linked with the child is set to PICKED_UP status. - Send notification to parent's phone. 	N/A	pass	2019/19/04	
SCB_TC_14	Driver skips a child	1. Driver touches x-shaped button.	<ul style="list-style-type: none"> - Daily trip linked with the child is set to SKIPPED status. - Send notification to parent's phone. 	N/A	pass	2019/19/04	
SCB_TC_15	Driver finishes a trip.	1. Driver touches “Đến nơi”.	<ul style="list-style-type: none"> - All Daily detail trip linked with the trip which have status other than SKIPPED or CANCELLED is set to FINISHED status. - Send notification to customer's phone. 	N/A	pass	2019/19/04	
SCB_TC_16	Driver view the bill.	1. Driver touches “Lấy hóa đơn”	<ul style="list-style-type: none"> - Show bill for driver. - All skipped or cancelled child is not included in the total fee. 	N/A	pass	2019/19/04	
SCB_TC_17	Driver create new Service which has working time overlap other existing services.	1. Driver touches “Mở dịch vụ”. 2. Driver inputs all required fields but time is overlapped.	<ul style="list-style-type: none"> - Alert “Dịch vụ bị trùng thời gian”. 	N/A	Pass	2019/19/04	

SCB_TC_18	Driver cancels a trip on a valid time	<ol style="list-style-type: none"> 1. Driver selects a contract in "Contracts" screen. 2. Driver touches "Xin nghỉ". 3. Driver selects desired day and input reason. 4. Driver touches "Xin nghỉ". 	<ul style="list-style-type: none"> - Send notification to customer. - Find temporary driver for that trip. 	N/A	pass	2019/19/04	
SCB_TC_19	Driver cancels a trip which not on a valid time	<ol style="list-style-type: none"> 1. Driver selects a contract in "Contracts" screen. 2. Driver touches "Xin nghỉ". 3. Driver selects desired day and input reason. 4. Driver touches "Xin nghỉ". 	<ul style="list-style-type: none"> - Alert "Chỉ có thể xin nghỉ trước giờ bắt đầu 6 giờ" 	N/A	pass	2019/19/04	
SCB_TC_20	Driver updates his/her service which is has PENDING or ACTIVE contract	<ol style="list-style-type: none"> 1. Driver selects a contract from "Contracts" screen. 2. Driver touches "Cập nhật". 	<ul style="list-style-type: none"> - Alert "Không thể cập nhật dịch vụ đang có hợp đồng hoạt động". 	N/A	pass	2019/19/04	
SCB_TC_21	Driver updates his/her service but newly inputted time is overlapped other existing service	<ol style="list-style-type: none"> 1. Driver selects a service form "Services" screen. 2. Driver touches "Cập nhật" 3. Driver inputs new information. 4. Driver touches "Cập nhật". 	<ul style="list-style-type: none"> - Alert "Bị trùng thời gian với dịch vụ khác" 	N/A	pass	2019/19/04	
SCB_TC_22	Driver starts a trip but GPS signal is missing	<ol style="list-style-type: none"> 1. Driver selects a contract in "Contracts" screen. 2. Driver touches "Bắt đầu chuyến đi". 	<ul style="list-style-type: none"> - Alert "Không thể lấy thông tin vị trí. Không thể bắt đầu." 	N/A	pass	2019/19/04	

SCB_TC_23	Driver views his/her feedbacks but he has no feedback	1. Driver selects “Feedbacks” entry in Draw menu.	- Feedback list is empty. - Average score is set to 0.	N/A	pass	2019/19/04	
SCB_TC_24	Driver logs in for the first time	1. Driver inputs account in the “Login” screen. 2. Driver touches “Đăng nhập”.	- Log driver in. - Open “Update account” screen.	N/A	pass	2019/20/04	
SCB_TC_25	Driver logs in but he/she has no car	1. Driver inputs account in the “Login” screen. 2. Driver touches “Đăng nhập”.	- Log driver in - Open “Update Car” screen.	N/A	pass	2019/20/04	
SCB_TC_26	Driver cancels contract	1. Driver selects a service from “Services” screen. 2. Driver holds a Requirement for secs. 3. Driver selects “Hủy hợp đồng” on appearing menu. 4. Driver inputs all required fields. 5. Driver touches “Hủy hợp đồng”.	- Create a cancel request. - Send notification to customer. - Recommend new driver for customer.	N/A	pass	2019/20/04	

4.2 Customer Test Case

ID	Test Case Description	Test Case Procedure	Expected Output	Inter Dependence Test Case	Result	Test Date	Note
SCB_TC_27	Customer logs in for the first time	1. Customer inputs account information in “Login” screen. 2. Customer touches “Đăng nhập”.	- Open “Update account” screen.	N/A	pass	2019/20/04	
SCB_TC_28	Customer creates a requirement with no child	1. Customer touches “Đăng ký yêu cầu dịch vụ mới”. 2. Customer inputs all required fields.	- Alert “Cần ít nhất một trẻ cho mỗi yêu cầu”.	N/A	pass	2019/20/04	

SCB_TC_29	Customer creates a valid requirement	<ol style="list-style-type: none"> 1. Customer touches “Đăng ký yêu cầu dịch vụ mới”. 2. Customer inputs all fields. 3. Customer selects a child from children list. 4. Customer touches “Đăng ký” 	<ul style="list-style-type: none"> - Notify “Đã tạo yêu cầu dịch vụ”. - Show “Requirements” screen. 	N/A	pass	2019/20/04	
SCB_TC_30	Customer removes a child	<ol style="list-style-type: none"> 1. Customer opens “Con của tôi” entry in the left menu. 2. Customer hold for secs on a child. 3. Customer touches “Xóa khỏi danh sách” in the appearing menu. 	<ul style="list-style-type: none"> - Notify “Đã xóa bé khỏi danh sách”. 	N/A	pass	2019/20/04	
SCB_TC_31	Customer removes a child who is in a ACTIVE contract	<ol style="list-style-type: none"> 1. Customer opens “Con của tôi” entry in the left menu. 2. Customer hold for secs on a child. 3. Customer touches “Xóa khỏi danh sách” in the appearing menu. 	<ul style="list-style-type: none"> - Notify “Không thể loại bỏ trẻ trong hợp đồng”. 	N/A	pass	2019/20/04	
SCB_TC_32	Customer finds driver and a service is match it.	<ol style="list-style-type: none"> 1. Customer opens “Requirements” screen. 2. Customer selects a requirement which has no contract. 3. Customer touches “Tìm tài xế”. 	<ul style="list-style-type: none"> - Show the services that match the requirement. 	N/A	pass	2019/20/04	
SCB_TC_33	Customer opens “Tracking” screen when touching “Arrival” notification	<ol style="list-style-type: none"> 1. Customer touches the os notification. 	<ul style="list-style-type: none"> - Show “Tracking” screen. - Current location of driver is continuously drawn in the map. 	N/A	pass	2019/20/04	
SCB_TC_34	Customer views bill after trip.	<ol style="list-style-type: none"> 1. Customer touches the os notification. 	<ul style="list-style-type: none"> - Show “Bill” screen for his/her. - Skipped or cancelled child is not included in fee. 	N/A	pass	2019/20/04	

SCB_TC_35	Customer views detailed information of a new requirement	<ol style="list-style-type: none"> 1. Customer opens “Requirements” screen. 2. Customer selects a requirement to view detail. 	<ul style="list-style-type: none"> - Show no contract. - Option menu changes to new mode that allow customer to update requirement. 	N/A	pass	2019/20/04	
SCB_TC_36	Customer views detailed information of an requirement with PENDING contract.	<ol style="list-style-type: none"> 1. Customer opens “Requirements” screen. 2. Customer selects a requirement to view detail. 	<ul style="list-style-type: none"> - Show Pending contract. - Option menu changes to PENDING mode that allows customer to cancel requirement. 	N/A	pass	2019/20/04	
SCB_TC_37	Customer views detailed information of an requirement with PENDING contract.	<ol style="list-style-type: none"> 1. Customer opens “Requirements” screen. 2. Customer selects a requirement to view detail. 	<ul style="list-style-type: none"> - Show Active contract. - Option menu changes to ACTIVE mode that allows customer to cancel trip, feedback,... 	N/A	pass	2019/20/04	
SCB_TC_38	Customer feedback driver with no content in a valid time	<ol style="list-style-type: none"> 1. Customer views requirement detail. 2. Customer opens option menu, 3. Customer selects “Phản hồi tài xế”. 4. Customer touches score bar to select score. 5. Customer touches “Gửi phản hồi”. 	<ul style="list-style-type: none"> - Send feedback to driver. 	N/A	pass	2019/20/04	
SCB_TC_39	Customer feedback driver second times within 24h	<ol style="list-style-type: none"> 1. Customer views requirement detail. 2. Customer opens option menu. 3. Customer selects “Phản hồi tài xế”. 4. Customer touches score bar to select score. 5. Customer inputs content. 6. Customer touches “Gửi phản hồi”. 	<ul style="list-style-type: none"> - Alert “Thời gian gửi phản hồi tiếp theo là [hh:mm:ss]”. 	N/A	pass	2019/20/04	

SCB_TC_40	Customer cancel trip for a child in requirement that has two children	<ol style="list-style-type: none"> 1. Customer views requirement detail. 2. Customer opens option menu. 3. Customer selects "Xin nghỉ". 4. Customer selectes children from list. 5. Customer selects day. 6. Customer inputs reason. 7. Customer selects "Xin nghỉ" 	<ul style="list-style-type: none"> - New cancel request is created. 	N/A	pass	2019/20/04	
SCB_TC_41	Customer extend current contract	<ol style="list-style-type: none"> 1. Customer views requirement detail. 2. Customer opens option menu. 3. Customer selects "Gia hạn". 4. Customer selects new time. 5. Customer selects "Yêu cầu gia hạn" 	<ul style="list-style-type: none"> - Send notification to Driver. - Create new contract. 	N/A	pass	2019/20/04	
SCB_TC_42	Customer updates child who is on an ACTIVE contract	<ol style="list-style-type: none"> 1. Customer opens "Children" screen on left menu. 2. Customer holds on a child. 3. Customer selects "Cập nhật thông tin" 	<ul style="list-style-type: none"> - Alert "Bé đang có hợp đồng có hiệu lực" 	N/A	pass	2019/20/04	
SCB_TC_43	Customer cancel contract	<ol style="list-style-type: none"> 1. Customer views requirement detail. 2. Customer opens option menu. 3. Customer selects "Hủy hợp đồng". 4. Customer input all required fileds 	<ul style="list-style-type: none"> - Create new cancel request. - Send notification to driver. 	N/A	pass	2019/20/04	

4.3 Logic Test Case

ID	Test Case Description	Test Case Procedure	Expected Output	Inter Dependence Test Case	Result	Test Date	Note

SCB_TC_44	Customer logins on a device that another customer currently uses.	<ol style="list-style-type: none"> 1. Driver opens SchoolBus Customer application. 2. Customer inputs his account in "Login" screen. 3. Driver touches "Đăng nhập". 	<ul style="list-style-type: none"> - Customer is logged in. - The token for that device is changed to be owned by new Customer. 	N/A	pass	2019/20/04	
SCB_TC_45	Driver logins on a device that another driver currently uses.	<ol style="list-style-type: none"> 1. Driver opens SchoolBus Driver application. 2. Driver inputs his account in "Login" screen. 3. Customer touches "Đăng nhập". 	<ul style="list-style-type: none"> - Driver is logged in. - The token for that device is changed to be owned by new Driver. 	N/A	pass	2019/20/04	
SCB_TC_46	Compute daily fee and total fee when customer inputs school location	<ol style="list-style-type: none"> 1. Customer opens "Register requirement" screen. 2. Customer selects school. 	<ul style="list-style-type: none"> - New distance is gotten. - New fee is updated. 	N/A	pass	2019/20/04	
SCB_TC_47	Driver rejects application's request to use GPS function	<ol style="list-style-type: none"> 1. Driver opens SchoolBus Driver application. 2. Driver selects Deny on GPS request. 	<ul style="list-style-type: none"> - Alert "Ứng dụng cần quyền truy cập vị trí để có thể hoạt động" - Close application. 	N/A	pass	2019/20/04	
SCB_TC_48	Can't get recommendation trip when driver starts the trip	<ol style="list-style-type: none"> 1. Driver opens "Services" screen. 2. Driver touches "Bắt đầu chuyến đi" on a trip which can be started. 	<ul style="list-style-type: none"> - Return the trip in default order. 	N/A	pass	2019/20/04	
SCB_TC_49	Deactivated driver login to application	<ol style="list-style-type: none"> 1. Driver opens application. 2. Driver inputs his/her account. 3. Driver touches "Đăng nhập". 	<ul style="list-style-type: none"> - Alert "Tài khoản của bạn đã bị khóa". 	N/A	pass	2019/20/04	

F. Software's User Manual

1. Installation Guide

1.1 Setting up environment at server side

1.1.1 Hardware requirements

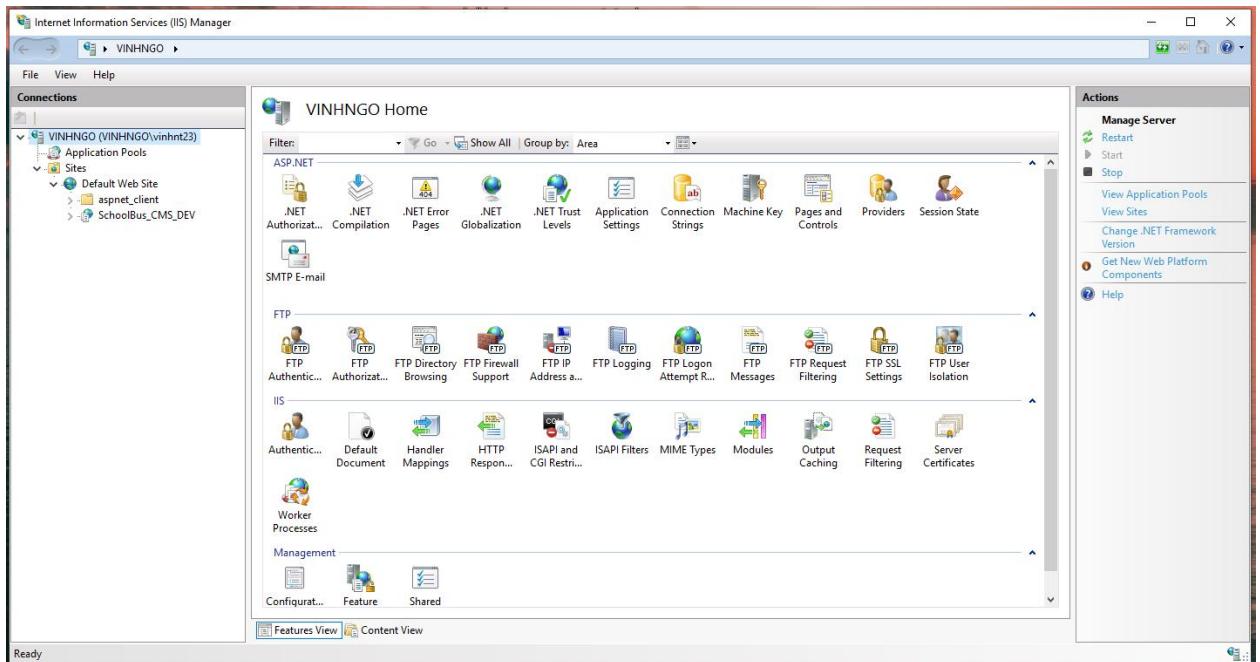
Hardware	Minimum Requirement	Recommended
Internet Connection	Cable, Wi-Fi (8 Mbps)	Cable, Wi-Fi (50 Mbps or more)
Operation System	XP, Vista, 7, 10, Window Server 2008	10, Window server 2008
Computer Processor	Intel® Xeon ® 1.4GHz	Intel® Xeon ® Quad Core (12M Cache, 2.50 GHz)
Computer memory	4GB RAM	32 GB RAM or more
Storage space	1GB	5GB or more

1.1.2 Software requirements

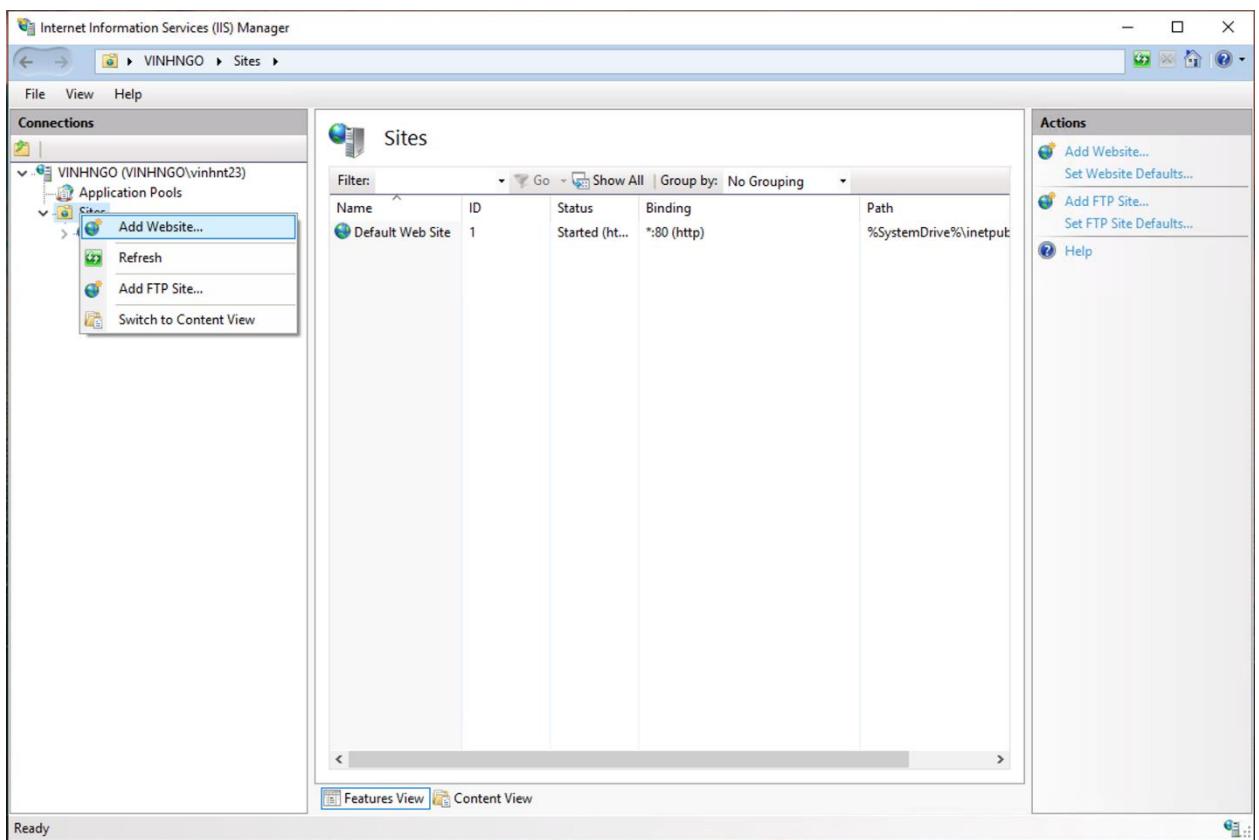
Software	Name / Version	Description
Operation System	Windows 10 build 1809, Windows Server 2016	Operating system and platform for development
Environment	C# 7, .NET framework 4.7.2	Specification for developing web services
	Node v10, npm v6	Specification for developing mobile application and web app
DBMS	Microsoft SQL Server	Used to create & manage the database for system
Source control	Git-SCM SourceTree	Used for source control
CI/CD	Jenkins	Used for creating new deployment
Web browser	Chrome 72 or above	Debugging browser

1.2 Deployment at server side

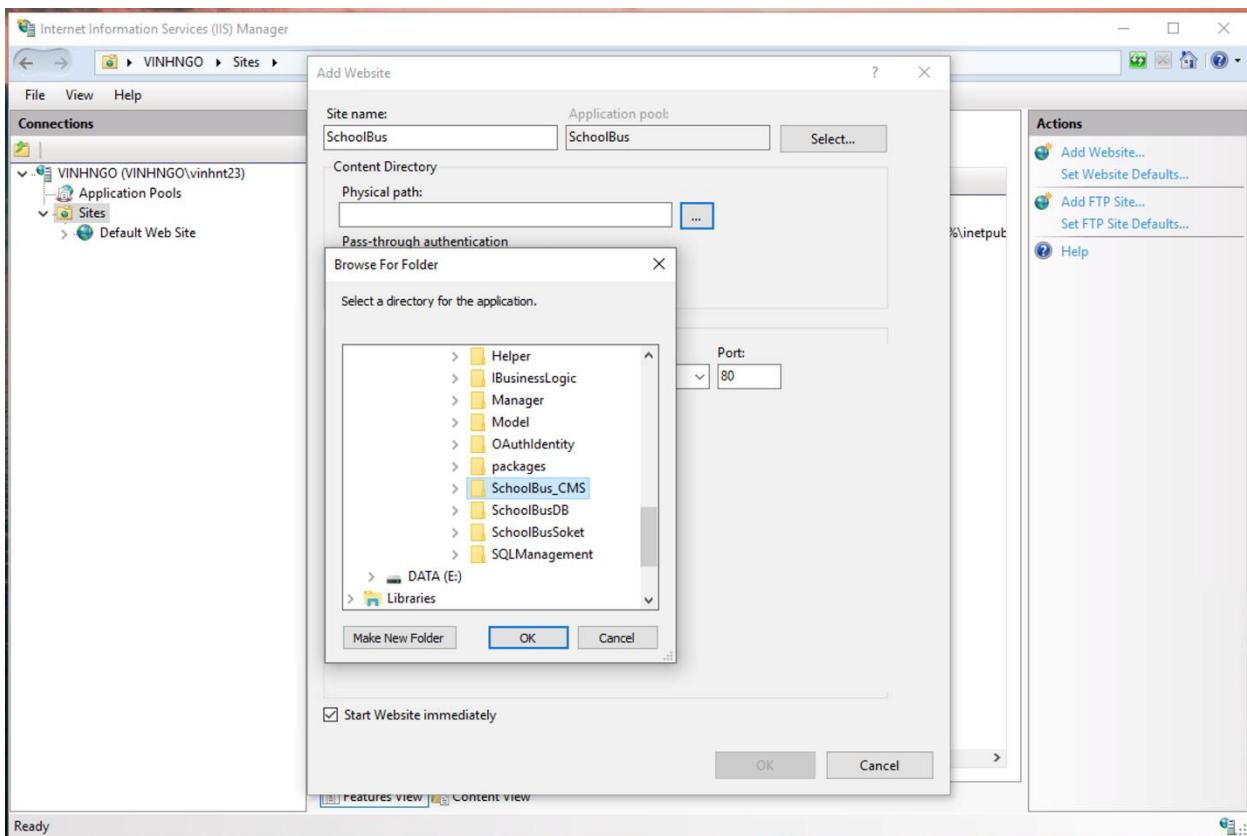
Step 1: Open IIS Manager in Windows



Step 2: Right click to “Sites” and choose “Add Website...”

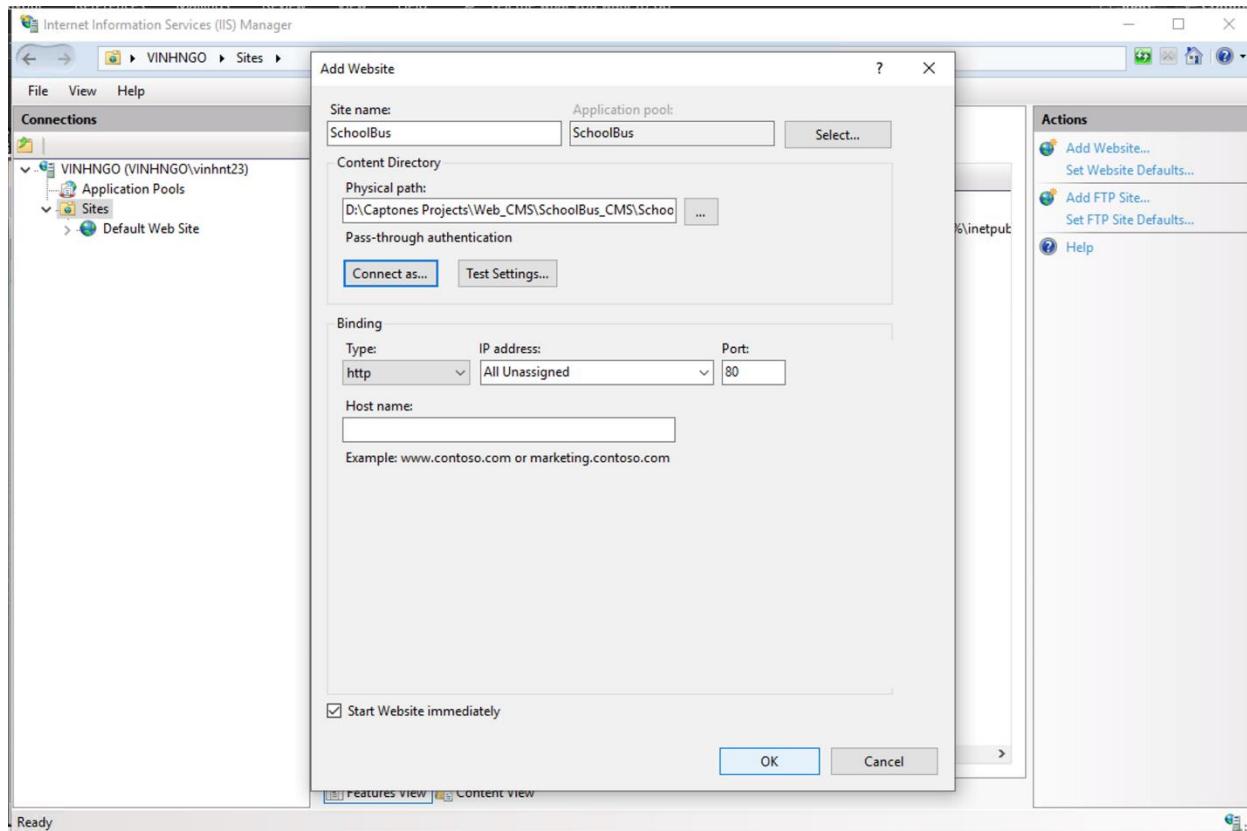


Step 3: Click Physical path and choose the build destination folder configured in Jenkins



Jenkins

Step 4: Click “OK” to add web application -> OK



1.3 Setting up environment at client side

1.3.1 Setting up for computer (web-admin)

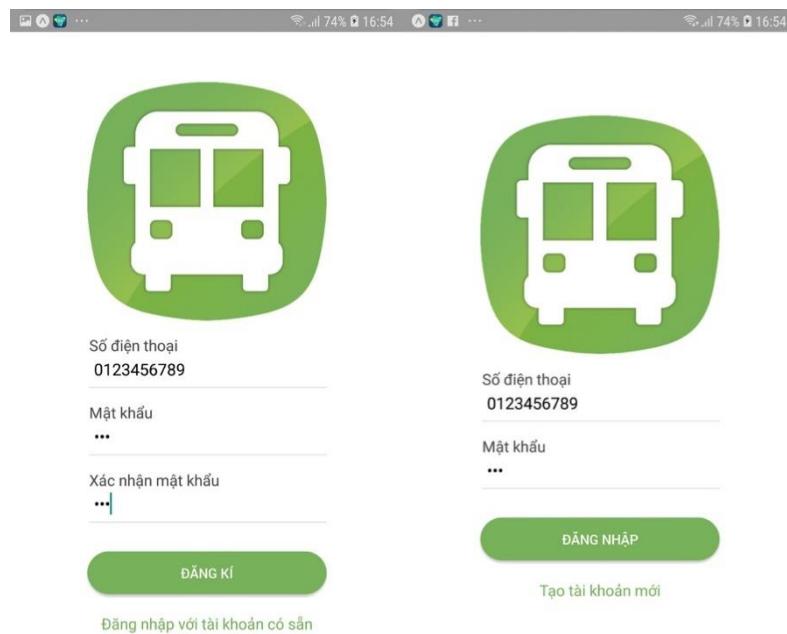
Software	Minimum Requirement	Recommended
Operation System	Windows XP, Vista, 7	Windows 8, 8.1, 10
Computer Processor	Intel® Core 2 Duo E7500® 2.93GHz	Intel® Core ® i3 2100 3.10Ghz
Computer memory	2GB RAM	4 GB RAM or more
Internet Connection	Cable, Wi-Fi (8 Mbps)	Cable, Wi-Fi (50 Mbps or more)
Web browser	Chrome, Firefox, Safari, ...	<ul style="list-style-type: none"> - Chrome from 72 - Firefox Quantum - ...

1.3.2 Setting up for mobile

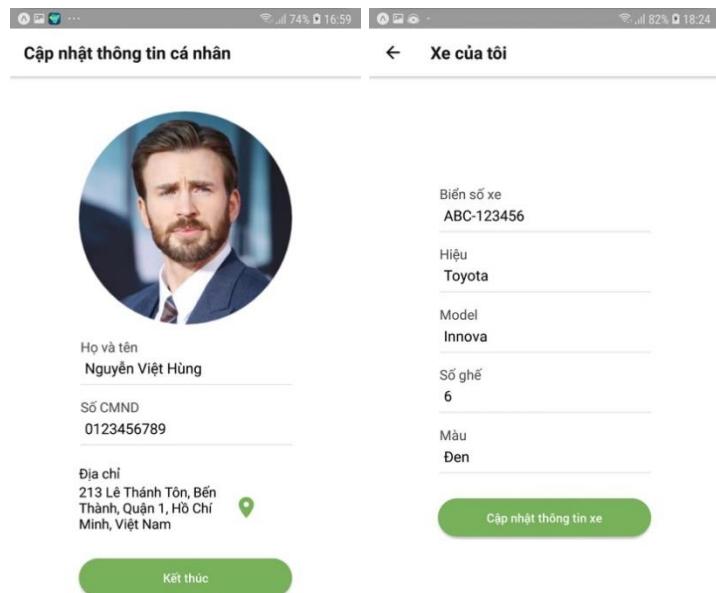
PC	Minimum Requirements	Recommended
Internet Connection	Wi-Fi (4 Mbps) / 2G / 3G	Wi-Fi (8 Mbps) / 4G / 5G
Operating System	Android 7 iOS 9	Android 9 iOS 12
Storage Space	100MB	200MB
Memory	1GB RAM	3GB RAM or more

2. User Guide

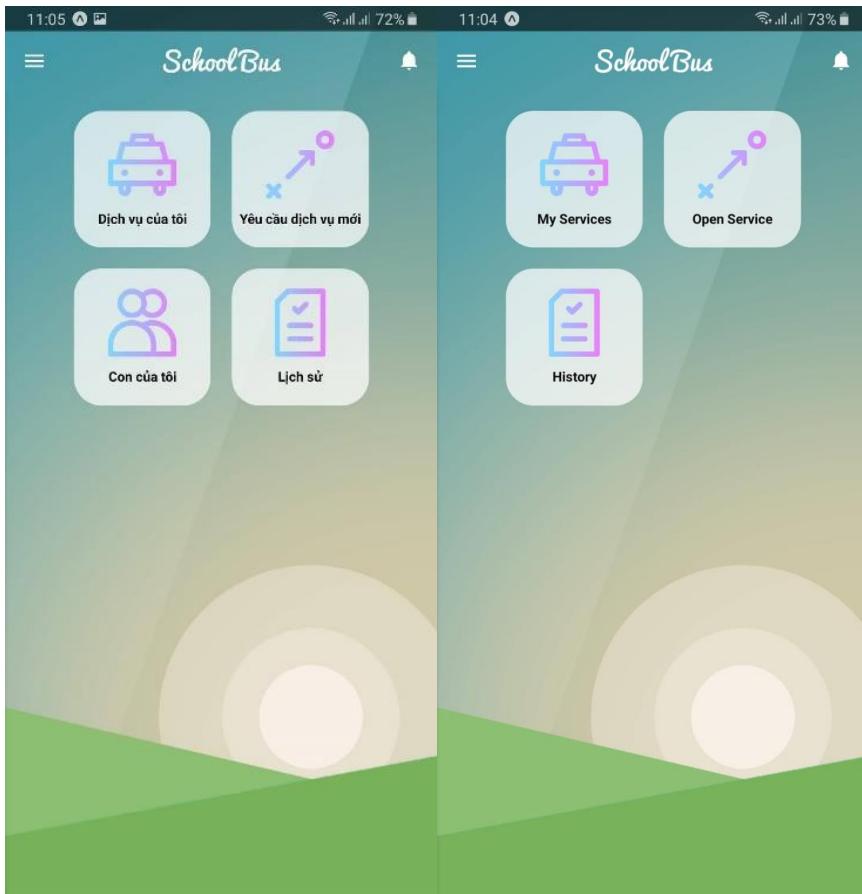
2.1 For Guest



After registering new account, customer/driver uses the account to login.



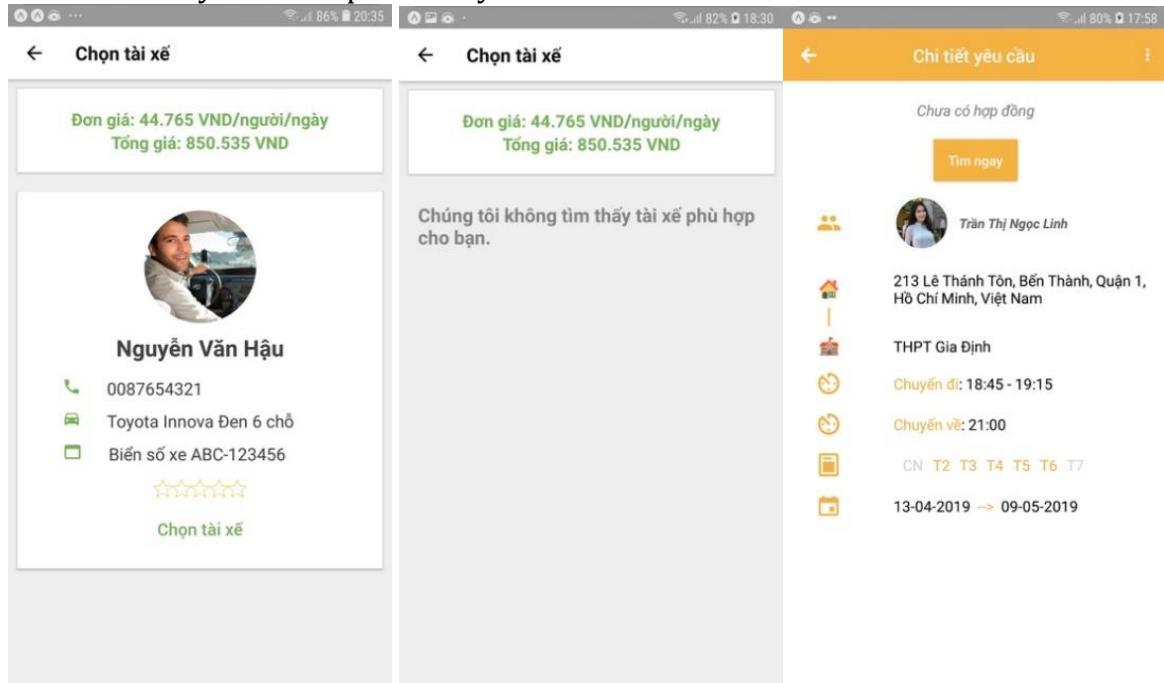
Then, customer/driver is required to update their profile. Also, driver must update their car's information.



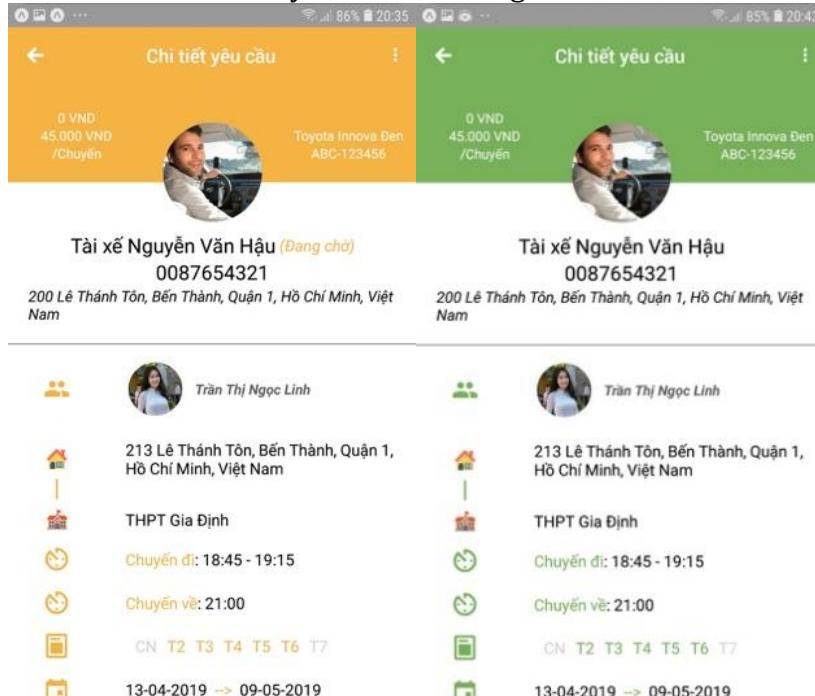
After updating information, customer/driver would be redirected to the home screen.

2.2 For Customer

A customer who wants to request a transporting service for their children from home to school would create a requirement. Customer must add some information about their children if they have not provided yet.



After creating a requirement, customers would be shown a list of appropriate driver's services matching their requirement. If there's no matching driver, the list is clear, customer could always find drivers again.



Customer will choose from the available services, then wait for response from the driver. If the driver accepts the request, a transporting contract would be formed.

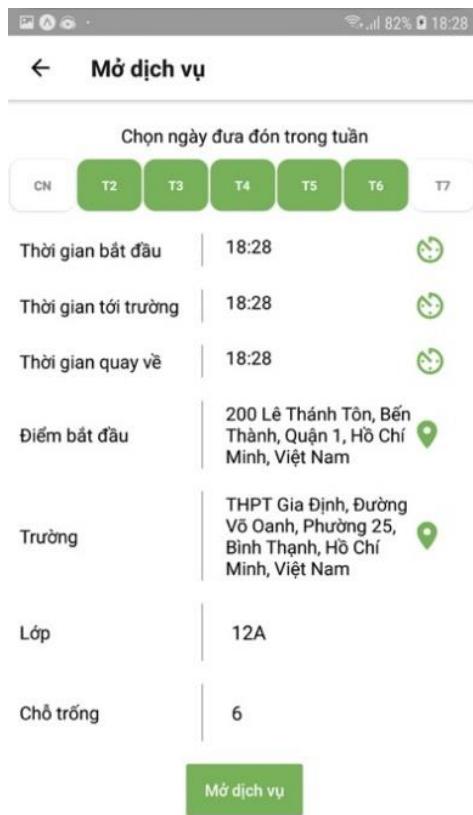


For every school day of their children, customers would be notified when the driver comes, when their children are picked up, and arrive at school. Customers also know their driver's current location on the trip.

Hóa Đơn	
T6, 19 Thg.4, 2019 17:20 Mã Chuyển: e8966585...	
Ngô Thế Vin 0932657789 84M-01543	
Chi tiết chuyến	
16:57	Bắt đầu
17:00	Bé Linh 110 Xóm Chiểu, Quận 4, Hồ Chí Minh, Việt Nam 29.000 VND
17:00	Bé Lan 110 Xóm Chiểu, Quận 4, Hồ Chí Minh, Việt Nam 29.000 VND
17:20	Trường THPT Trung Vương 3A Nguyễn Bình Khiêm, Bến Nghé, Quận 1, Hồ Chí Minh, Vietnam
Tổng cộng 58.000 VND	
Xem xác nhận của tài xế	

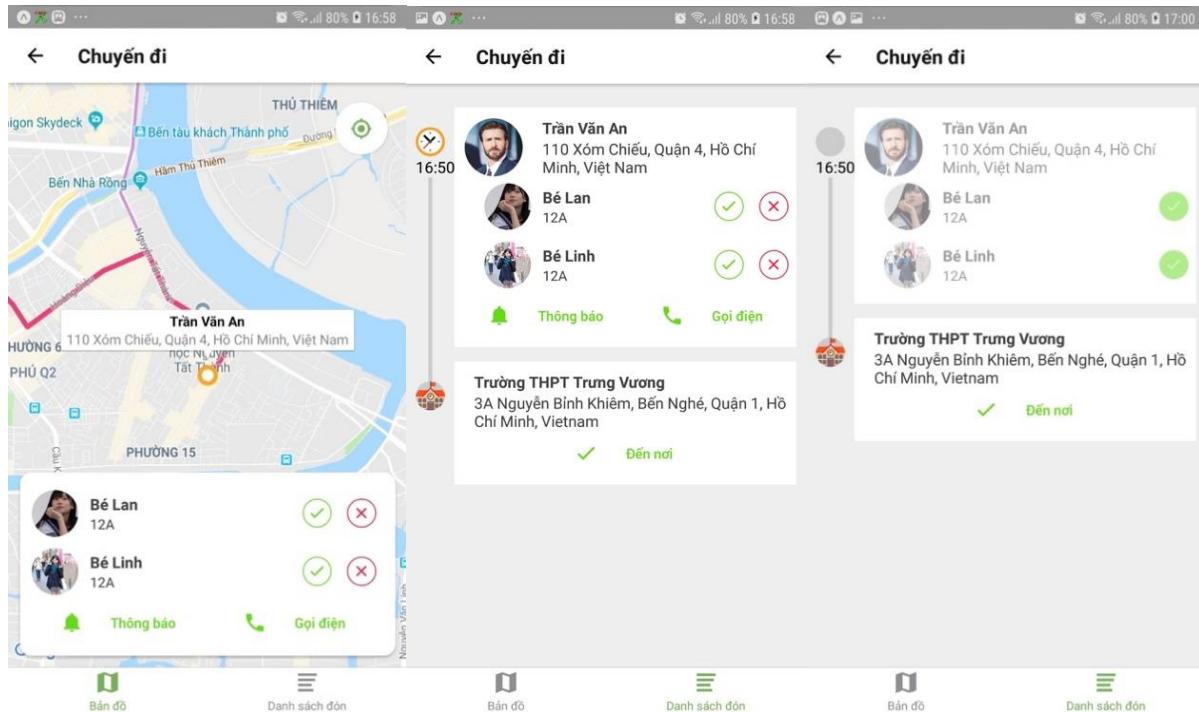
After their children arriving at school, customers would receive bill information about their children's school trip and an optional verification from the drivers.

2.3 For Driver



A driver who wants to open a transporting service would create a service with their car's available capacity.

When receiving requests for transporting, they would accept or reject. By accepting, the transporting contract would be created.



On school days of customer's children, drivers would receive the trip information with direction and information of their pick-up targets. When drivers reach a pick-up point, they would notify the customer and confirm that the child is picked up.

Chi tiết chuyến	Bắt đầu	
16:57		
17:00	Bé Linh 110 Xóm Chiểu, Quận 4, Hồ Chí Minh, Việt Nam	29.000 VND
17:00	Bé Lan 110 Xóm Chiểu, Quận 4, Hồ Chí Minh, Việt Nam	29.000 VND
17:00	Trường THPT Trung Vương 3A Nguyễn Bình Khiêm, Bến Nghé, Quận 1, Hồ Chí Minh, Vietnam	
Tổng cộng		58.000 VND

After reaching school, the trip would finish, and the drivers would get the bill information about their trip, verification is optional.