



MINISTRY OF EDUCATION AND TRAINING

ĐẠI HỌC FPT

FPT UNIVERSITY

Capstone Project Document

NEXT BUS STATION

Group 2	
Group members	Tran Gia Quoc Hung – HungTGQSE60917 Nguyen Quoc Dat – DatNQSE60924 Man Huynh Khuong – KhuongMHSE61148 (Dropped out) Nguyen Tri Quang – QuangNT60420 (Dropped out)
Supervisor	Mr. Kieu Trong Khanh
Ext. Supervisor	N/A
Capstone Project code	TMST

-Ho Chi Minh City, 06/01/2014-

This page is intentionally left blank

Table of Contents

Table of Contents	1
List of Tables.....	5
List of Figures.....	6
Definitions, Acronyms, and Abbreviations.....	9
A. Introduction.....	10
1. Project Information	10
2. Introduction.....	10
3. Current Situation in Viet Nam	10
4. Problem Definition	10
5. Proposed Solution	11
5.1. Feature functions	11
5.2. Advantage and disadvantages.....	11
6. Functional Requirements	11
6.1. View Route	11
6.2. Create Route.....	11
6.3. Search Route	11
6.4. Feedback and Manage Saved Data	12
6.5. System Management.....	12
7. Role and Responsibility	12
B. Software Project Management Plan	13
1. Problem Definition	13
1.1. Name of this Capstone Project.....	13
1.2. Problem Abstract.....	13
1.3. Project Overview	13
2. Project Organization.....	15
2.1. Software Process Model.....	15
2.2. Roles and Responsibilities	15
2.3. Tools and Techniques	16
3. Project Management Plan	16
3.1. Software development life cycle	16
3.2. Phase Detail.....	17
3.3. All Meeting Minutes	19

4. Coding Convention	19
C. Software Requirement Specification.....	20
1. User Requirement Specification.....	20
1.1. Guest Requirement	20
1.2. Member requirement.....	20
1.3. Staff requirement	20
2. System Requirement Specification.....	20
2.1. External Interface Requirement.....	20
2.2. System Overview Use Case.....	21
2.3. List of Use Case.....	22
3. Software System Attribute	64
3.1. Usability.....	64
3.2. Reliability	64
3.3. Availability	64
3.4. Security.....	64
3.5. Maintainability	64
3.6. Portability	65
3.7. Performance.....	65
4. Conceptual Diagram	65
4.1. Web Conceptual Diagram.....	66
4.2. Mobile Conceptual Diagram.....	67
D. Software Design Description	67
1. Design Overview.....	67
2. System Architectural Design.....	68
2.1. Web Application Architecture Description.....	68
2.2. Web Service Architecture Description	68
2.3. Windows Phone Application Architecture Description.....	69
3. Component Diagram	69
4. Detailed Description	70
4.1. Web Service Class Diagram	70
4.2. Web Service Class Diagram Explanation	71
4.3. Mobile Application Class Diagram	80
4.4. Mobile Application Diagram Explanation.....	80
4.5. Interaction Diagram.....	88

5. User Interface Design	109
5.1. Mobile Interface Design	109
5.2. Web Interface Design	127
6. Database Design	129
6.1. Web Service Logical Diagram	129
6.2. Web Service Data Dictionary.....	129
6.3. Mobile Application Logical Diagram.....	135
6.4. Mobile Application Data Dictionary	135
7. Algorithms	140
7.1. Find Next Station	140
7.2. Search Bus Rote.....	143
7.3. Automatically Create Route	149
7.4. Find stations around one location.....	154
7.5. Parse and Update data	155
E. System Implementation & Test.....	158
1. Introduction.....	158
1.1. Overview.....	158
1.2. Test Approach.....	158
2. Database Relationship Diagram	158
2.1. Web Service Physical Diagram	158
2.2. Web Service Data Dictionary.....	159
2.3. Mobile Application Physical Diagram	168
2.4. Mobile Application Data Dictionary	168
3. Performance Measures	175
3.1. Find next station performance.....	175
3.2. Search route performance	175
4. Test Plan	175
4.1. Features to be tested	175
4.2. Features not to be tested	176
5. System Testing Test Case	176
5.1. View Route Flow.....	177
5.2. Create Route Flow	181
5.3. Search Route Flow.....	184
F. Software User's Manual	186

1. Installation Guide	186
1.1. Setting up environment at server side	186
1.2. Deployment at server.....	186
1.3. Setting up environment at client side	190
2. User Guide.....	192
2.1. Main page.....	192
2.2. Create route page.....	194
2.3. Search route page	197
G. Appendix.....	198

List of Tables

Table 1 Roles and Responsibility.....	12
Table 2 Hardware Requirement for Server.....	14
Table 3 Hardware Requirement for Mobile.....	14
Table 4 Roles and Responsibilities Details	16
Table 5 Phase 1: Requirement Analysis	17
Table 6 Phase 2: Design.....	18
Table 7 Phase 3: Implementation	18
Table 8 Phase 4: Testing.....	18
Table 9 Phase 5: Maintenance	19
Table 10 Web Entity Data Dictionary.....	129
Table 11 Web Detail Data Dictionary.....	135
Table 12 Mobile Application Entity Data Dictionary.....	136
Table 13 Mobile Detail Data Dictionary	140
Table 14 Relationship between LocalDatabase and ReferenceDatabase.....	140
Table 15 Web Data Dictionary	159
Table 16 Web Attribute Data Dictionary.....	167
Table 17 Mobile Application Data Dictionary	169
Table 18 Mobile Application Attribute Data Dictionary.....	175
Table 19 Main page.....	192
Table 20 View route page	193
Table 21 Create route page.....	194
Table 22 Edit route menu.....	195
Table 23 Save created route	196
Table 24 Search route page	197
Table 25 Location menu.....	198

List of Figures

Figure 1 Modified Waterfall Development Model.....	15
Figure 2 Mobile Application Overview Use Case	21
Figure 3 Website Overview Use Case.....	22
Figure 4 Member Overview Use Case.....	23
Figure 5 <Member> Search route.....	23
Figure 6 <Member> Save searched route.....	25
Figure 7 <Member> Create own route	27
Figure 8 <Member> View route list	29
Figure 9 <Member> View route	30
Figure 10 <Member> Add reminder	32
Figure 11 <Member> Find next station.....	33
Figure 12 <Member> Show reminder.....	35
Figure 13 <Member> Set reminder by average time	37
Figure 14 <Member> Search location.....	38
Figure 15 <Member> Save bus path history	40
Figure 16 <Member> Upload bus path history	41
Figure 17 <Member> Edit own route.....	42
Figure 18 <Member> Delete own route	46
Figure 19 <Member> Edit reminder	47
Figure 20 <Member> Remove reminder.....	49
Figure 21 <Member> Feedback station	50
Figure 22 <Member> Feedback bus route.....	52
Figure 23 <Member> Feedback location	54
Figure 24 <Staff> Overview Use Case	56
Figure 25 <Staff> Edit bus route.....	57
Figure 26 <Staff> Edit bus station	58
Figure 27 <Staff> Edit bus path	60
Figure 28 <Staff> Add location.....	62
Figure 29 Web Conceptual Diagram	66
Figure 30 Mobile Conceptual Diagram	67
Figure 31 TMST System Architectural.....	68
Figure 32 Component Diagram.....	69
Figure 33 Web Service Class Diagram	70
Figure 34 Mobile Application Class Diagram	80
Figure 35 <Member> Search route	88
Figure 36 <Member> Save searched route.....	89
Figure 37 <Member> Create own route	90
Figure 38 <Member> View route list	91
Figure 39 <Member> View route	92
Figure 40 <Member> Add reminder to station.....	93
Figure 41 <Member> Add reminder to location	94
Figure 42 <Member> Search location.....	95
Figure 43 <Member> Find next station.....	95

Figure 44 <Member> Set reminder by location	96
Figure 45 <Member> Set reminder by average time	97
Figure 46 <Member> Search walking route.....	98
Figure 47 <Member> Save bus path history	98
Figure 48 <Member> Upload bus path history	99
Figure 49 <Member> Edit own route.....	100
Figure 50 <Member> Delete own route	101
Figure 51 <Member> Edit reminder	101
Figure 52 <Member> Remove reminder.....	102
Figure 53 <Member> Feedback station	103
Figure 54 <Member> Feedback bus route.....	105
Figure 55 <Member> Feedback location	106
Figure 56 <Staff> Edit bus route.....	107
Figure 57 <Staff> Edit bus station	108
Figure 58 <Staff> Edit bus path.....	108
Figure 59 <Staff> Add location.....	109
Figure 60 Main page.....	109
Figure 61 Choose bus route menu	110
Figure 62 Choose own route menu.....	111
Figure 63 Settings page	112
Figure 64 View route page	113
Figure 65 Add reminder popup.....	114
Figure 66 Edit reminder page.....	115
Figure 67 Create route page	116
Figure 68 Location menu.....	117
Figure 69 Add route menu	117
Figure 70 Edit route menu	118
Figure 71 Save route popup	119
Figure 72 Search route page	120
Figure 73 Location menu.....	121
Figure 74 Search settings	121
Figure 75 Save route	123
Figure 76 Feedback bus route page	124
Figure 77 Feedback bus station page.....	125
Figure 78 Feedback location page.....	126
Figure 79 Manage Bus Route page	127
Figure 80 Edit bus path popup	128
Figure 81 Web Service Logical Diagram	129
Figure 82 Mobile Application Logical Diagram	135
Figure 83 Find next station algorithm	143
Figure 84 Search route algorithm	147
Figure 85 Create route algorithm - Select turns	150
Figure 86 Create route algorithm - Adjust route	152
Figure 87 Create route algorithm - Adjust bus route and location.....	153
Figure 88 Create route algorithm - Adjust two bus routes.....	154

Figure 89 Web Service Physical Diagram	158
Figure 90 Mobile Application Physical Database	168
Figure 91 View Route Flow	176
Figure 92 Create Route Flow	176
Figure 93 Search Route Flow.....	176
Figure 94 Prepare deployment package folders	186
Figure 95 Prepare deployment package for web service	187
Figure 96 Prepare deployment package for web application.....	187
Figure 97 Connect database.....	188
Figure 98 IIS Control Panel.....	188
Figure 99 Add web service in IIS	189
Figure 100 Test deployment	189
Figure 101 Deployment result	189
Figure 102 Add Web Site on IIS.....	190
Figure 103 Next Bus Station on Windows Phone store website.....	191
Figure 104 Next Bus Station on Windows Phone store mobile	191
Figure 105 Main page.....	192
Figure 106 View route page	193
Figure 107 Create route page	194
Figure 108 Edit route menu	195
Figure 109 Save created route	196
Figure 110 Search route page	197
Figure 111 Location menu.....	198

Definitions, Acronyms, and Abbreviations

Name	Definition
NBS	Next Bus Station
GPS	Global Positioning System
Bus route	The bus itself.
Bus path	The path from one station to another station, travel by bus. One bus contains many bus paths in order to form the whole route of bus.
Walking path	The path from one station to another station, travel on foot.
OS	Operating System
API	Application Programming Interface
HTTP	Hyper Text Transfer Protocol

A. Introduction

1. Project Information

- Project name: **Next Bus Station**
- Project Code: **TMST**
- Product Type: **Windows Phone Application, Website**
- Start Date: **January 06th, 2014**
- End Date: **May 7th, 2014**

2. Introduction

Nowadays bus in Vietnam becomes a popular transportation, especially for students and workers. But it is not easy for people who do not have much traffic knowledge to travel by bus confidently, for example, people who have just move to a new city or foreign people who cannot communicate in Vietnamese fluently. There are several objective reasons make people afraid of bus: how to know which buses should get in, where to get off, what to do if buses are too crowded to recognize the street... Even some of regular bus users miss their expected stations.

In our project, we want to create a Windows Phone application that allows users to search for buses they need to go, create a route from buses they have known, alert them when they need to get off... Besides that, to ensure accuracy of data, we also build a Web Application to manage data of system.

3. Current Situation in Viet Nam

Below are some comparison applications and users behaviors:

- Windows phone applications (BusMap, BusHCM, Taxi – Xe Bus Hà Nội, VietBus...): they have functions such as look up buses' information, show buses' routes on map, find route between two points... but cannot save searched data, cannot alert users when they need to get off, and less helpful when helping users in real bus travelling.
- When users want to go to an address which is not very well-known, they may know which bus they have to get on somehow, but will be anxious about which station to get off.

4. Problem Definition

Below are the advantages and disadvantages of current bus application in Windows Phone store and website in Viet Nam:

- Advantages:
 - Easy to use. Most features are offline.
 - Provide users quick access to information about buses and stations' information.
- Disadvantages:
 - Cannot alert users when their expected station comes.

- Cannot let users create the route they want.
- Cannot let users recommend more attributes or download defined attributes which help them to recognize station easier.
- Require users to have network connection or GPS.
- Totally offline applications may contain out-of-date information.

5. Proposed Solution

The application allows users to view bus route and use their GPS or address to know when expected station comes near. Users can also create route or search route. Users can set reminder at station or location. By taking advantages of offline map of Windows Phone device, they can also use those functions in offline mode. Users can send feedback to server and update data.

5.1. Feature functions

- Users can view bus route and set reminders based on station or location.
- Users can use GPS to keep tracking about next station of bus.
- Users can determine next station by providing their current address.
- Users can create their own route to use as a bus route.
- Users can search bus route and save to use as a bus route.
- Users can send their feedback.
- Users can update data in application.

5.2. Advantage and disadvantages

- Advantages: Provide users many supports from searching for bus route to alert their expected stations. Most of functions can also be used in offline mode.
- Disadvantages: Users must have suitable Windows Phone device and offline map on device. Data which is gotten from <http://buytphcm.com.vn/> may be out-of-date, or that website may not available anymore, so it is essential to get actual data from users' feedbacks.

6. Functional Requirements

Function requirements of the system are listed as below:

6.1. View Route

- User can choose to view bus route or saved route and enable tracking to know which next station is and alert when set station or location comes near.
- User can add reminders at stations or location.
- User can get next station by providing address only.

6.2. Create Route

- User can create a route when (s)he know which buses (s)he has to get on but confuse about where to get off.
- User can save created route for using later.

6.3. Search Route

- User can search for route between 2 locations.

- User can save searched route for using later.

6.4. Feedback and Manage Saved Data

- User can send feedback server.
- User can manage their saved data such as location, unsent feedback, saved route,...

6.5. System Management

- Staff can manage all parsed data such as bus, station, location, user, feedback...
- Staff can manage parsing function have to approve changes between parsed data and current data to make changes

7. Role and Responsibility

No	Full Name	Role	Position	Contact
1	Kieu Trong Khanh	Project Manager	Instructor	khanhkkt@fpt.edu.vn
2	Tran Gia Quoc Hung	Developer	Leader	hungtgqse60917@gmail.com
3	Nguyen Quoc Dat	Developer	Member	datnqse60924@fpt.edu.vn

Table 1 Roles and Responsibility

B. Software Project Management Plan

1. Problem Definition

1.1. Name of this Capstone Project

- Next Bus Station (TMST).

1.2. Problem Abstract

As bus has become a popular transportation, the demands of searching for routes, stations of bus users are increasing. Not only users desire to be supported before they get on bus but also while they are travelling by bus. Next Bus Station is an application which help users to create the route they want, use it to travel, provide several utilities for helping users get to expected place. Users can also send their feedback to help system works better.

1.3. Project Overview

1.3.1. Current Situation

- Windows Phone applications: There are several applications which provide users buses and stations' information. Some applications have good search route function which provide users various recommend routes between 2 points.
- Advantages: easy to use, helpful for experienced bus users.
- Disadvantages: data is out-of-date so some buses and stations' information are wrong. Users cannot propose ideas or give feedback. Users cannot save searched route. Applications may not be helpful when users are on bus.

1.3.2. The Proposed System

The system is intended for both inexperienced and experienced bus users to help them view buses' routes, search and save searched routes, create routes based on their own for tracking for next station or set reminders at stations and locations. The system also encourage users to participate in recommending new data or correcting incorrect data to make data more practical.

1.3.2.1 Mobile Application

- Create route: User can create route which includes some bus routes they want.
- Search route: User can search for route between 2 locations and save searched route for using later.
- View route and tracking: Users can view bus route, view created route or saved searched route. Users can set reminders at stations or locations.
- User can send feedback about data in system.
- Various policies are apply to let users go online once in a while.

1.3.2.2 Web Application

- Staff can manage data such as bus, station, route, location, user...
All changes must be approved by staff before make changes in database.

1.3.3. Boundaries of the System

- The system can be used by every people with Windows Phone 8.0 devices or later, or with a laptop with internet connection.
- The language of application is Vietnamese.
- The complete product includes:
 - The mobile application, for member.
 - The website, for staff.
 - All the process document involved.

1.3.4. Development Environment

1.3.4.1 Hardware requirements

For Server		
Windows	Minimum Requirements	Recommended
Internet Connection	Cable, Wi-Fi (4 Mbps)	Cable, Wi-Fi (8 Mbps)
Operating System	Window Server 2008	Window Server 2008
Computer Processor	Intel® Xeon ® 1.4GHz	Intel® Xeon ® Quad Core (12M Cache, 2.50 GHz)
Computer Memory	1GB RAM	2GB or more

Table 2 Hardware Requirement for Server

For Mobile		
Mobile	Minimum Requirements	Recommended
Internet Connection	2 Mbps	4 Mbps
Operating System	Windows Phone 8.0	Windows Phone 8.0
Hardware		GPS supported
Memory	150MB	150MB or more

Table 3 Hardware Requirement for Mobile

1.3.4.2 Software requirements

- Window Server 2008: operating system and platform for development.
- SQL Server 2008 Enterprise R2: used to create and manage the database for system.
- Visual Studio 2012: used to implement Windows Phone application, website and web service.
- Google Code & TortoiseSVN: used for source control.
- StarUML version 5.0.2: used to create models and diagrams.
- www.lucidchart.com : used to create models and diagrams.

2. Project Organization

2.1. Software Process Model

Project is developed under modified waterfall model (SASHIMI)

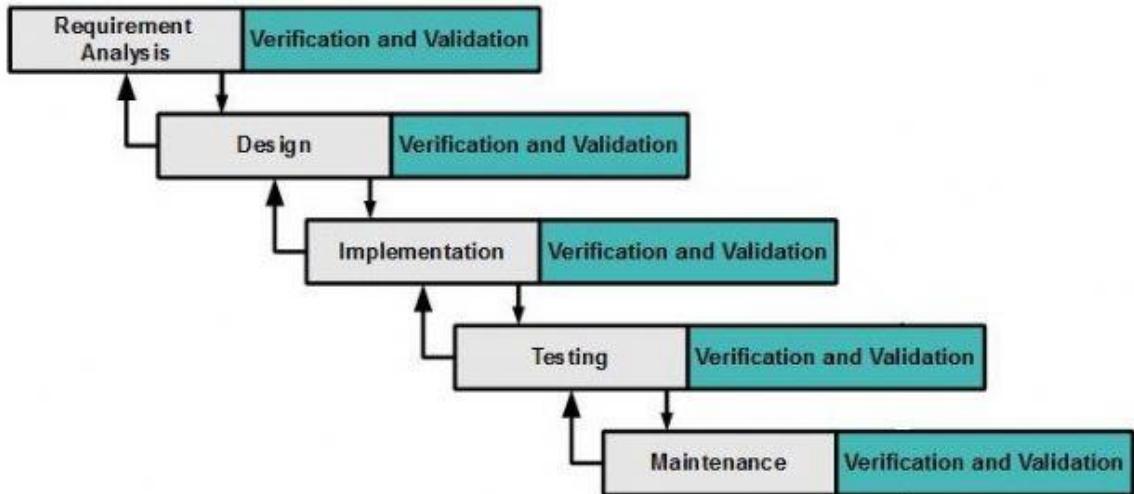


Figure 1 Modified Waterfall Development Model

For more information: <http://www.waterfall-model.com/sashimi-waterfall-model/>

Explanation: We use Modified Waterfall Development Model for Next Bus Station application development for several reasons, mostly because:

- List of functional requirement and non-functional requirement may be made cleared only during Design phase because main data about bus route is taken from <http://buyttphcm.com.vn/> which will decide which function can or cannot be done.
- Architecture which is made from Design phase may be changed during Implementation phase because some of Windows Phone application development's supporting features may be known during implementation, and it may change the architecture. Besides that, some functions may be updated or changed, requiring design document should be updated.
- Implementation may need to be updated during Testing phase because data which is parsed from <http://buyttphcm.com.vn/> may be inaccurate or change in structure. Some of functions may need to be updated during testing.

2.2. Roles and Responsibilities

No	Full name	Role in Group	Responsibilities
1	Kiều Trọng Khánh	Project manager	<ul style="list-style-type: none">• Specify user requirement• Control the development process• Give out technique and business analysis support
2	Trần Gia Quốc Hưng	Team Leader,	<ul style="list-style-type: none">• Managing process

		BA, DEV, Tester	<ul style="list-style-type: none"> Clarifying requirements Prepare documents GUI Design Designing database Coding Create test plan Testing
3	Nguyễn Quốc Đạt	Team Member, BA, DEV, Tester	<ul style="list-style-type: none"> Clarifying requirements Prepare documents GUI Design Designing database Coding Create test plan Testing

Table 4 Roles and Responsibilities Details

2.3. Tools and Techniques

- Mobile Application: Windows Phone SDK 8.0.
- Database in Mobile Application: SQL Server Compact Edition 3.5
- Front-end technologies: HTML5, CSS3, JavaScript, jQuery, AJAX.
- Back-end:
 - Website: ASP.NET MVC4 + Entity Framework 5.
 - Scheduler: Quartz version 2.3.2.
 - Parse data from website: LINQ to XML version 4.0.
- Web server: Microsoft IIS version 8.5.
- Web service: WCF Service Application in .NET Framework 4.5.
- Database Management System: Microsoft SQL Server 2008 Enterprise R2.

3. Project Management Plan

3.1. Software development life cycle

Phase	Description	Deliverables	Resource needed	Dependencies and Constraints	Risks
Requirement Analysis	<ul style="list-style-type: none"> - Collect requirements from customer. - Identify and clarify requirements for the system in general. 	<ul style="list-style-type: none"> -Introduction of proposed system. -Software requirement specification. -Project Task Plan. - Prototypes 	20 man-days	N/A	<ul style="list-style-type: none"> - Missing requirement - Unclear scope of project - Lack of member share of understanding

Design	<ul style="list-style-type: none"> - Architecture design for the system - Detail design using top-down break down - Choose Architecture style 	<ul style="list-style-type: none"> - Software Design Document - Base code structure - Technology notes 	20 man-days	Depend on "Requirement Analysis"	<ul style="list-style-type: none"> - Lack of experience - Not fulfil requirement.
Implementation	<ul style="list-style-type: none"> - Coding system core functions and other feature with GUI - Unit test 	<ul style="list-style-type: none"> - Main user's functions on mobile and website 	50 man-days	Depend on "Design".	<ul style="list-style-type: none"> - Lack of experience and knowledge. - Human mistake.
Testing	<ul style="list-style-type: none"> - Integration test the system - Alpha test - Correct bugs - Beta test - Acceptance test 	<ul style="list-style-type: none"> - Test document 	20 man-days	Depend on "Implementation"	<ul style="list-style-type: none"> - Lack of experience - Missing test case
Maintenance	<ul style="list-style-type: none"> - Deploy on sever and mobile 	<ul style="list-style-type: none"> - Installation guide. - User Manual 	10 man-days	Depend on "Testing"	<ul style="list-style-type: none"> - Lack of experience

3.2. Phase Detail

3.2.1. Phase 1: Requirement Analysis

Task	Description	Author
1. Collect requirements	Find which systems currently provide similar service, their strengths and weakness.	HungTGQ, DatNQ
2. Identify and clarify main functions.	Define which main functions system should provide.	HungTGQ, DatNQ
3. Create System Introduction.	Complete Introduction Report.	HungTGQ
4. Software Project Management Plan.	Prepare Project Management Plan.	HungTGQ
5. Prototype.	Build a prototype of proposed system (Website/Mobile).	HungTGQ, DatNQ
6. SRS	Create SRS document.	HungTGQ, DatNQ

Table 5 Phase 1: Requirement Analysis

3.2.2. Phase 2: Design

Task	Description	Author
1. Detailed Design	Compare new document with existed documents of system.	HungTGQ, DatNQ,
2. Database Design	Based on parsed data to recommendation. Based on other needs to recommendation.	HungTGQ, DatNQ,
3. Technology research	Study Windows Phone Application Development Create find next bus station engine. Create search engine for online search and offline search route.	HungTGQ, DatNQ,
4. Design Document	Create software design document	HungTGQ, DatNQ,

Table 6 Phase 2: Design

3.2.3. Phase 3: Implementation

Task	Description	Author
1. Front-end web functions	Implement front-end functions on web	
2. Back-end web functions	Implement back-end functions on web	DatNQ
3. Mobile functions	Implement mobile application	HungTGQ,
4. Unit testing	Write test case and testing for web functions	HungTGQ, DatNQ
	Write test case and testing for mobile functions	HungTGQ,

Table 7 Phase 3: Implementation

3.2.4. Phase 4: Testing

Task	Description	Author
1. Integration testing	Write test case and testing systems	HungTGQ, DatNQ,
2. Alpha testing	Do alpha test with customer	HungTGQ, DatNQ,

Table 8 Phase 4: Testing

3.2.5. Phase 5: Maintenance

Task	Description	Author

1. Installation guide	Write installation guide	HungTGQ
2. User Manual	Write user manual	HungTGQ, DatNQ

Table 9 Phase 5: Maintenance

3.3. All Meeting Minutes

Refer to Meeting Minutes folder.

4. Coding Convention

C#: Using to develop website.

Summary:

- Naming Convention:
 - For variable's name, use camel case. Example: routeType, nextStation...
 - For function name, class name, use pascal case. Example: GetAllStations, GetCurrentLocation...
- Layout Convention:
 - Write only one statement/declaration per line.
 - Indent continuation one tab stop (four spaces).
 - Add at least one blank line between method definitions and property definitions.
 - Use parentheses to make clauses in an expression apparent.
- Language Guidelines:

Using C# Code Convention From:

<http://msdn.microsoft.com/en-us/library/vstudio/ff926074.aspx>

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 use some functions in the system. To use all functions, guest must login. These are some functions guest can use:
 - Register.
 - Login in Website to become staff.
 - Login in Mobile Application to become member.

1.2. Member requirement

- Member is guest who uses his account to login to the system. These are some functions member can use:
 - View route, check next station, alert next station, set reminder on station or location.
 - Search route.
 - Create route.
 - Give feedback.
 - Save data.
 - Update data.

1.3. Staff requirement

- Staff is the person who manages data on server. These are some functions staff can use:
 - Manage location
 - Manage bus route.
 - Manage bus station.
 - Manage bus path.
 - Manage account.
 - Manage parsing.

2. System Requirement Specification

2.1. External Interface Requirement

2.1.1. User Interface

- General requirement for graphics user interface of website and application on mobile device is the GUI should be simple, clear, intuitive, and reminiscent.
- Some design principle will be taken into consideration:
 - Microsoft Design Principle - Microsoft [Ref: Appendix 1]
 - UX/UI Guidelines for Windows Phone 8 - Dina Helmy [Ref: Appendix 2]

2.1.2. Hardware Interface

- N/A

2.1.3. Software Interface

- Chromes (version 41.0.2272.118), Firefox (version 37.0.1) with Resolution (1024 x 768) or bigger and must support JavaScript and HTML5.

2.1.4. Communication Protocol

- Using HTTP/HTTPS protocol.

2.2. System Overview Use Case

2.2.1. Mobile Application Overview Use Case

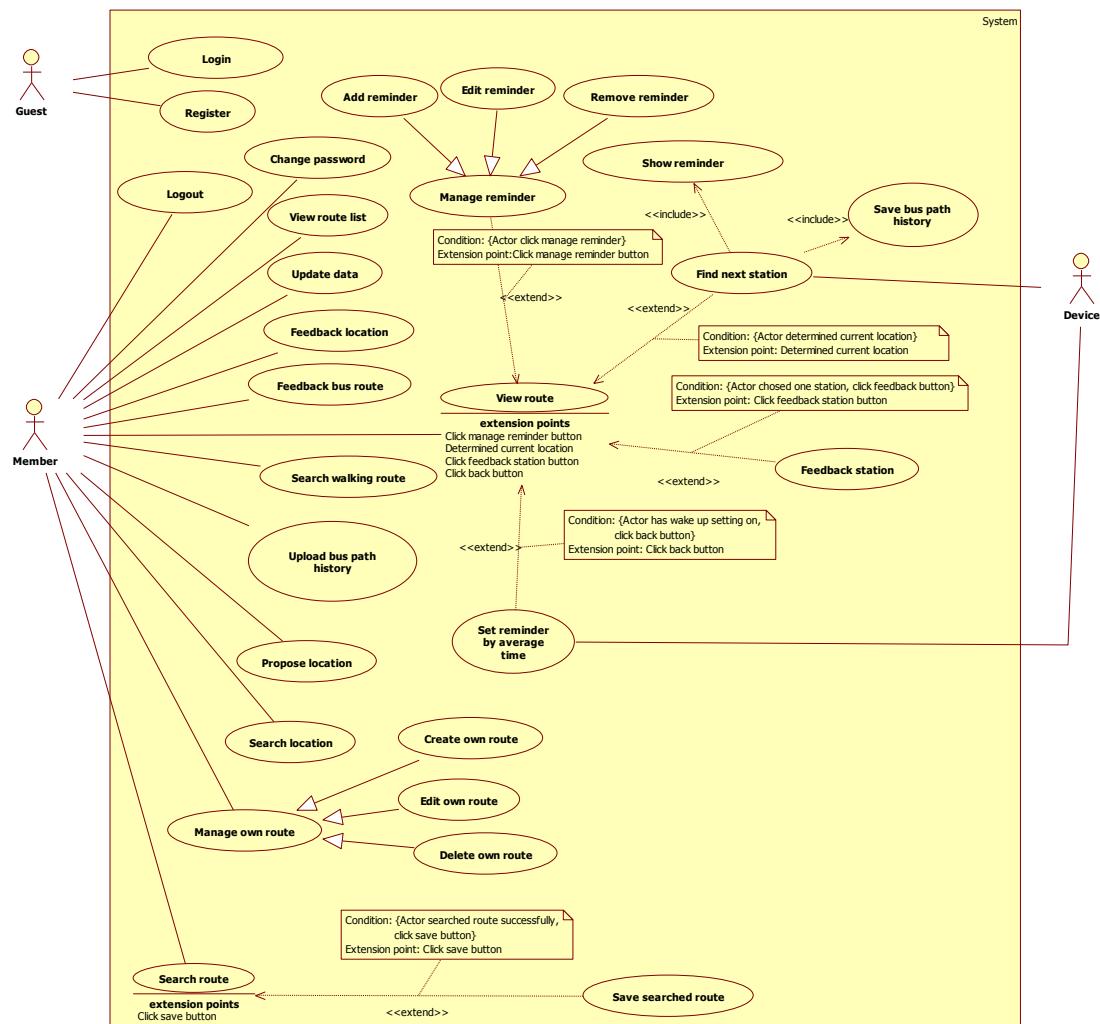


Figure 2 Mobile Application Overview Use Case

2.2.2. Website Overview Use Case

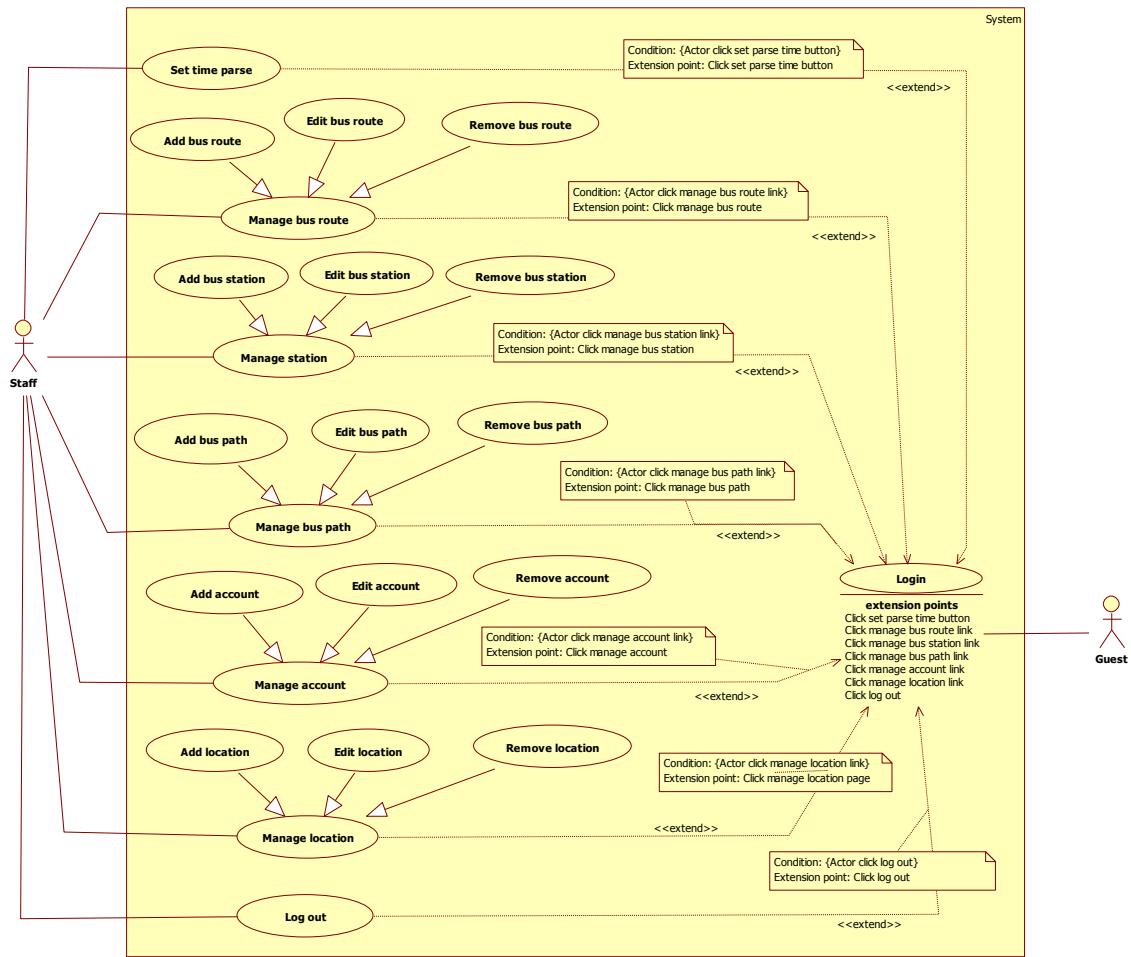


Figure 3 Website Overview Use Case

2.3. List of Use Case

Below are some use case specifications of frequently used functions, in order of frequency.

2.3.1. <Member> Overview Use Case

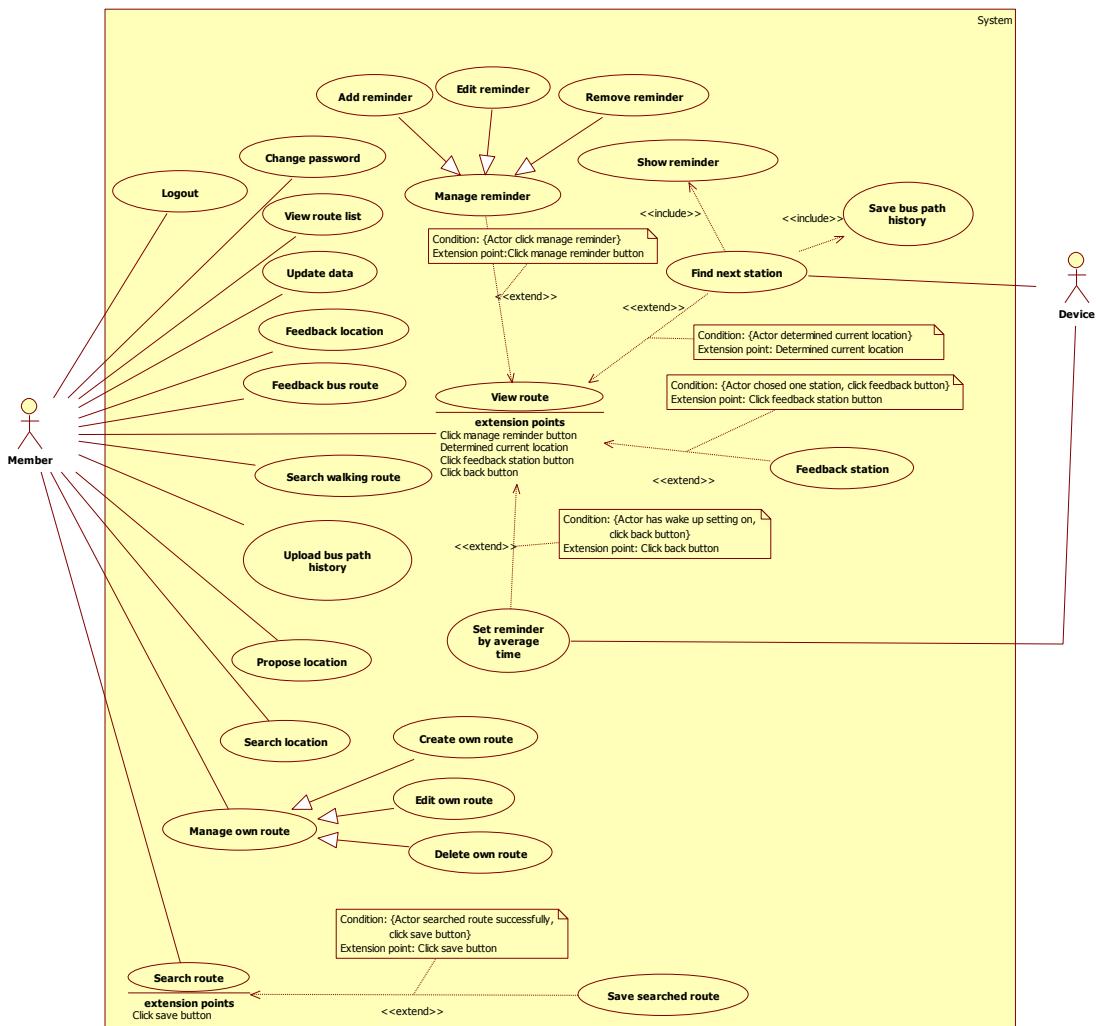


Figure 4 Member Overview Use Case

2.3.1.1 <Member> Search route Use case diagram

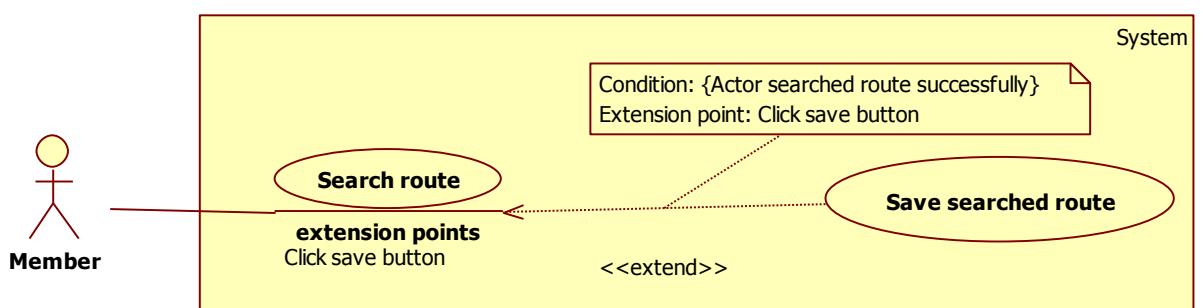


Figure 5 <Member> Search route

Use Case Specification

USE CASE – TMST001			
Use Case No.	TMST001	Use Case Version	2.0
Use Case Name	Search route		

Author	Tran Gia Quoc Hung		
Date	17/04/2015	Priority	High

Actor:

- Member

Summary:

- This use case allows member to search route between two locations.

Goal:

- Search route results is shown.

Triggers:

- Member clicks “tìm đường” button.

Preconditions:

- Guest is logged in as Member.

Post condition:

- **Success:** List of search results is shown.
- **Fail:** Show error message.

Main Success Scenario:

Step	Actor Action	System Response
1	Member chooses one location as start location.	System marks start location.
2	Member chooses another location as stop location.	System marks stop location.
3	Member clicks “tìm đường” button.	<ul style="list-style-type: none"> - System searches for routes from start location to stop location: <ul style="list-style-type: none"> • Search for bus stations around start location. • Search for bus stations around stop location. • Search route from start stations to stop stations. - System shows first searched result on map: bus stations is shown in form of markers, bus paths is shown in form of lines. - System shows list of results. [Alternative 1,2,3] [Exception 1]

Alternative:

No	Actor Action	System Response
1	There is no bus station around start location.	System shows message “Không có trạm xe nào xung quanh điểm bắt đầu.”
2	There is no bus station around stop location.	System shows message “Không có trạm xe nào xung quanh điểm kết thúc.”
3	No suitable route is found.	System shows message “Không có đường đi phù hợp nào được tìm thấy.”

Exception:

No	Actor Action	System Response
1	Error occur during searching process.	System shows message “Chức năng tìm kiếm không thể hoạt động, vui lòng nhấn nút sửa chữa và thử lại. Nếu vẫn hiện lỗi này xin liên hệ với chúng tôi.”

Relationships: extends to Save searched route (Click save button)

Business Rules:

- Start location and/or stop location may be chosen by searching location or holding on map.
- Searched route includes bus route only.
- In online mode, searched route can be saved. In offline mode, searched route can be found but cannot be saved.
- There are 2 search modes system allows member to choose in setting: “đường đi bộ ít nhất” and “đường đi ít xe nhất”.
- By default, system searches route in “đường đi ít xe nhất” mode with following settings:
 - Total distance of walking: no more than 300m.
- In “đường đi bộ ít nhất” search mode, systems allows member to modify following settings:
 - Number of bus route: from 1 to 5.
- In “đường đi ít xe nhất” search mode, system allows member to modify following setting:
 - Total distance of walking in route: from 0m to 1000m.
- System should show as many suitable results as possible.

2.3.1.2 <Member> Save searched route

Use case diagram

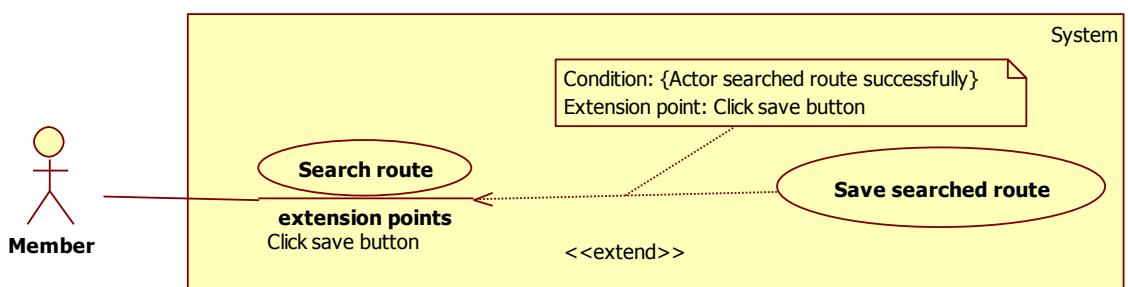


Figure 6 <Member> Save searched route

Use Case Specification

USE CASE – TMST002			
Use Case No.	TMST002	Use Case Version	2.0
Use Case Name	Save searched route		
Author	Tran Gia Quoc Hung		
Date	17/04/2015	Priority	High

Actor:									
- Member									
Summary:									
- This use case allows member to save searched routes between 2 locations.									
Goal:									
- Searched route is saved to database.									
Triggers:									
- Member taps on “lưu lại” button.									
Preconditions:									
- Guest login as Member. - Member has used search route function and get a result with at least 1 bus route. - Internet is available.									
Post condition:									
- Success: Searched route is saved to database. - 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>Member taps on “lưu lại” button.</td><td> <ul style="list-style-type: none"> - System shows a page: <ul style="list-style-type: none"> • “Tên”: textbox, max length 100 characters, required. • “lưu lại”: button • “quay lại”: button </td></tr> <tr> <td>2</td><td>Member fill in “Tên” textbox and taps on “lưu lại” button. [Alternative 1]</td><td> <ul style="list-style-type: none"> - System saves searched route data to database. - System navigate back to main page. [Exception 1,2] </td></tr> </tbody> </table>	Step	Actor Action	System Response	1	Member taps on “lưu lại” button.	<ul style="list-style-type: none"> - System shows a page: <ul style="list-style-type: none"> • “Tên”: textbox, max length 100 characters, required. • “lưu lại”: button • “quay lại”: button 	2	Member fill in “Tên” textbox and taps on “lưu lại” button. [Alternative 1]	<ul style="list-style-type: none"> - System saves searched route data to database. - System navigate back to main page. [Exception 1,2]
Step	Actor Action	System Response							
1	Member taps on “lưu lại” button.	<ul style="list-style-type: none"> - System shows a page: <ul style="list-style-type: none"> • “Tên”: textbox, max length 100 characters, required. • “lưu lại”: button • “quay lại”: button 							
2	Member fill in “Tên” textbox and taps on “lưu lại” button. [Alternative 1]	<ul style="list-style-type: none"> - System saves searched route data to database. - System navigate back to main page. [Exception 1,2] 							
Alternative:									
<table border="1"> <thead> <tr> <th>No</th><th>Actor Action</th><th>System Response</th></tr> </thead> <tbody> <tr> <td>1</td><td>Member taps back key.</td><td>System navigates back to previous page.</td></tr> </tbody> </table>	No	Actor Action	System Response	1	Member taps back key.	System navigates back to previous page.			
No	Actor Action	System Response							
1	Member taps back key.	System navigates back to previous page.							
Exception:									
<table border="1"> <thead> <tr> <th>No</th><th>Actor Action</th><th>System Response</th></tr> </thead> <tbody> <tr> <td>1</td><td>Name is not empty.</td><td>Show error message “Vui lòng điền tên của đường đi”.</td></tr> <tr> <td>2</td><td>Length of name is over 100 characters</td><td>Show error message “Tên đường đi tối đa 100 ký tự”.</td></tr> </tbody> </table>	No	Actor Action	System Response	1	Name is not empty.	Show error message “Vui lòng điền tên của đường đi”.	2	Length of name is over 100 characters	Show error message “Tên đường đi tối đa 100 ký tự”.
No	Actor Action	System Response							
1	Name is not empty.	Show error message “Vui lòng điền tên của đường đi”.							
2	Length of name is over 100 characters	Show error message “Tên đường đi tối đa 100 ký tự”.							
Relationships: extended by Search route (Click save button).									
Business Rules:									
<ul style="list-style-type: none"> - Searched route is saved to database as own route. All bus routes' data in searched route are reference data. - In online mode, if searched bus route includes any bus routes which has not been saved to database, system will save these bus routes to database automatically. 									

2.3.1.3 <Member> Create own route Use case diagram

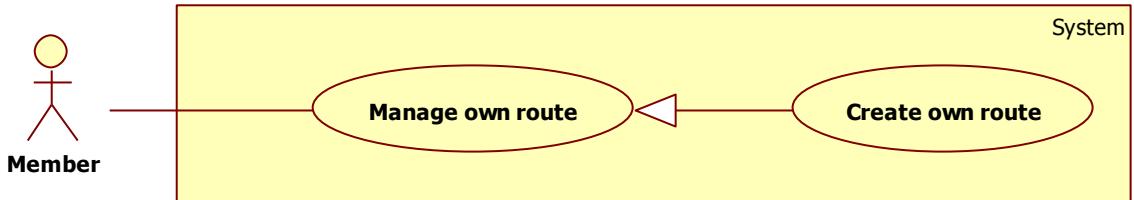


Figure 7 <Member> Create own route

Use Case Specification

USE CASE - TMST003

Use Case No.	TMST003	Use Case Version	2.0
Use Case Name	Create own route		
Author	Tran Gia Quoc Hung		
Date	17/04/2015	Priority	High

Actor:

- Member

Summary:

- This use case allows member creates new route.

Goal:

- New route is created.

Triggers:

- Member clicks “tạo đường” button.

Preconditions:

- Guest is logged in as Member.

Post condition:

- **Success:** New route is created.
- **Fail:** Show error message.

Main Success Scenario:

Step	Actor Action	System Response
1	Member chooses some buses of expecting route in order from route list in main page. Member taps “tạo đường” button.	System navigates to create route page.
2	Member chooses one location as start location.	System marks start location.
3	Member chooses another location as stop location.	System marks stop location.
4	Member taps on “tự động tạo đường” button.	System shows confirm message to confirm the order of chosen buses.
5	Member taps OK button. [Alternative 1]	<ul style="list-style-type: none"> - System creates route from start location, list of buses, stop location. - System auto adjust start station and stop station of each bus route in own route. - System shows created route on map.
6	Member taps on “lưu lại”	System shows popup:

	button.	- “Tên đường đi”: textbox, max length: 100 characters, required.
7	Member fill in name and taps on save button.	System saves the route. System navigates back to main page. [Exception 1,2]

Alternative:

No	Actor Action	System Response
1	Member taps Cancel button.	System closes confirm message.

Exception:

No	Actor Action	System Response
1	Name is empty.	System shows message “Xin vui lòng điền tên đường đi.”
2	Length of name is over 100 characters.	System shows message “Tên đường đi tối đa 100 ký tự.”

Relationships: N/A

Business Rules:

- Own route can only be created from bus routes, not from other own routes. Own route must have one bus route to be saved. Maximum number of bus routes in own route is 5.
- Own route can also be created manually. Start location and stop location are optional.
- Each bus route can only be added to own route once.
- Start location and/or stop location may be chosen by searching location or holding on map.
- Data of bus routes in own route is reference data: only start station index and stop station index of each bus route are stored for each bus route in own route.
- When saving data of own route to database:
 - If “Bắt đầu” / “Kết thúc” location is server location and has not been downloaded, location is downloaded to database first.
 - If “Bắt đầu” / “Kết thúc” location is address and has not been saved, location is saved to database first.
- In system, different own routes can have same name.
- Automatically adjust bus route: When one bus route in own route is changed, or start location / stop location is changed, system will adjust creating route automatically:
 - If start location is changed: Find nearest “Trạm lên” of first bus route.
 - If bus route is first bus route: Find nearest “Trạm lên” from start location. Find nearest “Trạm xuống” to next bus route.
 - If bus route is last bus route: Find nearest “Trạm xuống” to stop location.
 - If bus route is neither first nor last bus route: Find nearest “Trạm lên” to previous bus route. Find nearest “Trạm xuống” to next bus

- route.
- If stop location is changed: Find nearest “Trạm xuống” of last bus route.
 - When auto adjust bus route: if member has chosen “Trạm xuống” or “Trạm lên” manually for one bus route, system will prioritize it. Otherwise system will automatically adjust.
 - When one bus route is chosen in bus route list, the bus route with the same number but different turn is hidden, the chosen bus route is marked with different color. When the bus route is removed, both will be shown in default color.
 - In online mode, walking line is searched by service. In offline mode, walking line is searched by query offline map. If walking line cannot be shown, system won't show anything.
 - In online mode, if created route includes any bus routes which has not been saved to database, system will save these bus routes to database automatically.
 - System will create reminders at stop station of each bus route in own route automatically.

2.3.1.4 <Member> View route list

Use case diagram

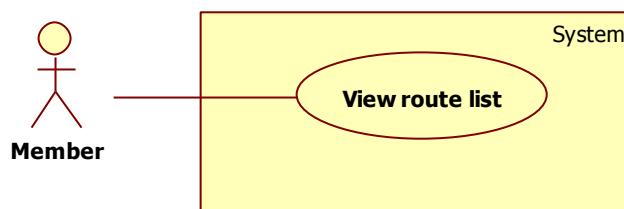


Figure 8 <Member> View route list

Use Case Specification

USE CASE – TMST004			
Use Case No.	TMST004	Use Case Version	2.0
Use Case Name	View route list		
Author	Tran Gia Quoc Hung		
Date	17/04/2015	Priority	Normal
Actor:	<ul style="list-style-type: none"> - Member 		
Summary:	<ul style="list-style-type: none"> - This use case allows member to view list of routes. 		
Goal:	<ul style="list-style-type: none"> - List of routes is shown successfully. 		
Triggers:	<ul style="list-style-type: none"> - Member is navigated to main page. 		
Preconditions:	<ul style="list-style-type: none"> - Guest is logged in as member. 		
Post condition:	<ul style="list-style-type: none"> - Success: List of route is viewed successfully. - Fail: Do nothing. 		

Main Success Scenario:

Step	Actor Action	System Response
1	Member is navigated to main page.	<p>System shows list of route, each item includes:</p> <ul style="list-style-type: none"> - Bus number: label (Number of bus route, blank of route is own route). - Name: label. - Turn: label. <p>[Alternative 1]</p>

Alternative:

No	Actor Action	System Response
1	Internet is not available and no bus route is stored.	System shows label: "Không có xe buýt nào được lưu lại".

Exception: N/A

Relationships: N/A

Business Rules:

- If internet is available: List of route includes: List of own route (sort by created date), list of saved bus route (sort by bus number), list of unsaved bus route (sort by bus number), in order.
- If internet is not available: List of route includes: List of own route (sort by created date), list of saved bus route (sort by bus number), in order.

2.3.1.5 <Member> View route

Use case diagram

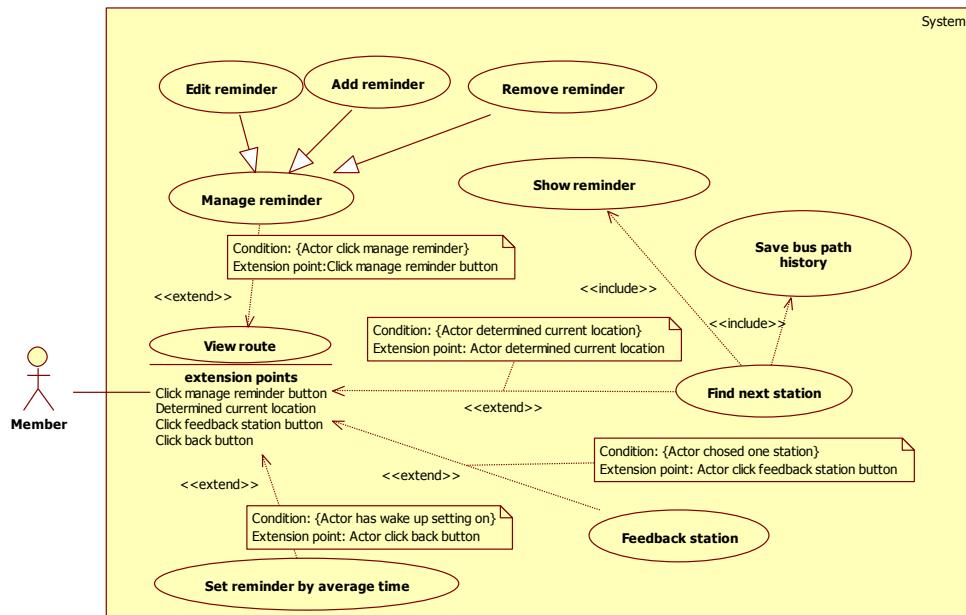


Figure 9 <Member> View route

Use Case Specification

USE CASE – TMST005

Use Case No.	TMST005	Use Case Version	2.0
Use Case Name	View route		
Author	Tran Gia Quoc Hung		
Date	17/04/2015	Priority	High
Actor:			
- Member			
Summary:			
<ul style="list-style-type: none"> - This use case allows member to view a route. - List of related reminders is loaded. 			
Goal:			
<ul style="list-style-type: none"> - Route is shown in map. 			
Triggers:			
<ul style="list-style-type: none"> - Member chooses one route to show. 			
Preconditions:			
<ul style="list-style-type: none"> - Guest is logged in as Member. 			
Post condition:			
<ul style="list-style-type: none"> - Success: Route is shown. - Fail: Do nothing. 			
Main Success Scenario:			
Step	Actor Action	System Response	
1	Member chooses one route and click view button.	System navigates to show route page and shows route on map: <ul style="list-style-type: none"> - Lines to show paths between stations, markers to show stations. 	System gets related reminders data from database.
Alternative Scenario: N/A			
Exceptions: N/A			
Relationships: Extends to Find next station (Actor determined current location), extends to Feedback station (Actor click feedback station button), extends to Set reminder by average time (Actor click back button).			
Business Rules:			
<ul style="list-style-type: none"> - Show own route: each bus routes' lines has different color. - Active reminded stations' markers has different color. - List of stations is loaded in extend menu. - Corrupted data: Data may be corrupted if member stops application forcefully while updating, or parsed data on server is corrupted. - In online mode, if chosen bus route has not been saved to database, system will save it automatically. 			

2.3.1.6 <Member> Add reminder

Use case diagram

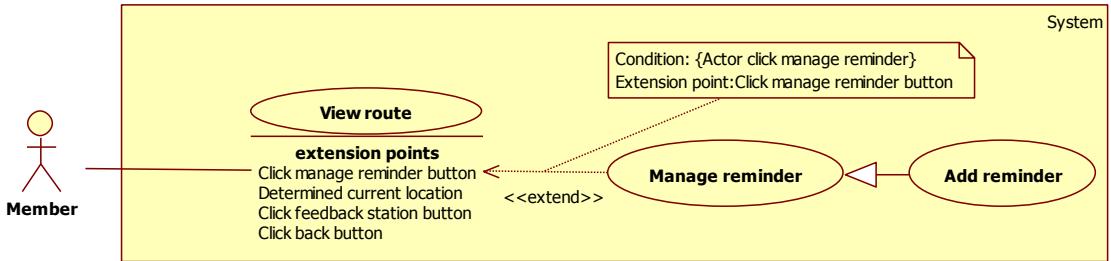


Figure 10 <Member> Add reminder

Use Case Specification

USE CASE - TMST006		
Use Case No.	TMST006	Use Case Version
Use Case Name	Add reminder	
Author	Tran Gia Quoc Hung	
Date	17/04/2015	Priority
Actor:	<ul style="list-style-type: none"> - Member 	
Summary:	<ul style="list-style-type: none"> - This use case allows member to add reminder to one route. 	
Goal:	<ul style="list-style-type: none"> - Reminder is added. 	
Triggers:	<ul style="list-style-type: none"> - Member adds reminder. 	
Preconditions:	<ul style="list-style-type: none"> - Guest is logged in as Member. 	
Post condition:	<ul style="list-style-type: none"> - Success: Reminder is added. - Fail: Show error message. 	
Main Success Scenario:		
Step	Actor Action	System Response
1	Member taps on location marker or station marker on map.	System displays menu contains “Đặt nhắc nhở” option.
2	Member taps on “Đặt nhắc nhở” option.	System opens popup: <ul style="list-style-type: none"> - Data of location: name, category, alternate name. - “Báo trước”: combobox, raw source: “1 trạm”, “2 trạm”, “3 trạm”, default value: “1 trạm”. - “Đặt lúc”: combobox, raw source: “ngay bây giờ”, “sau này”, default value: “ngay bây giờ”. - “Thêm”: button - “Quay lại”: button
3	Member chooses options and taps on “Thêm” button. [Alternative 1]	System adds reminder to database and close popup. [Exception 1,2]

Alternative:

No	Actor Action	System Response
1	Member taps on “Quay lại” button.	System close popup.

Exception:

No	Actor Action	System Response
1	The reminded location is too far from all stations of route.	System shows message “Xe buýt không thể dừng gần điểm này.”
2	Reminder is set on server location and server location cannot be downloaded.	System shows message “Không thể tải được địa điểm. Xin vui lòng thử lại sau.”

Relationships: N/A

Business Rules:

- Reminder is used to alert member when expecting station come near while travelling on bus.
- Member can add as many reminders as they want.
- If “Báo lúc” is “ngay bây giờ”, reminder’s state will be active. If “Báo lúc” is “sau này”, reminder’s state will be inactive.
- Reminder can be set on station or location.
 - Reminder cannot be set on first station.
 - Reminder can only be set on:
 - Second station without “Báo trước” property.
 - Third station with “Báo trước” is “1 trạm”.
 - Fourth station with “Báo trước” is “1 trạm” or “2 trạm”.
- If reminder is set on location, system searches for nearest station of route to set reminder on.
- If reminder is set on location, location must be saved in database first.
 - If location is searched from server and location has not been downloaded: system downloads location to database.
 - If location is an address: system adds location to database.

2.3.1.7 <Member> Find next station

Use case diagram

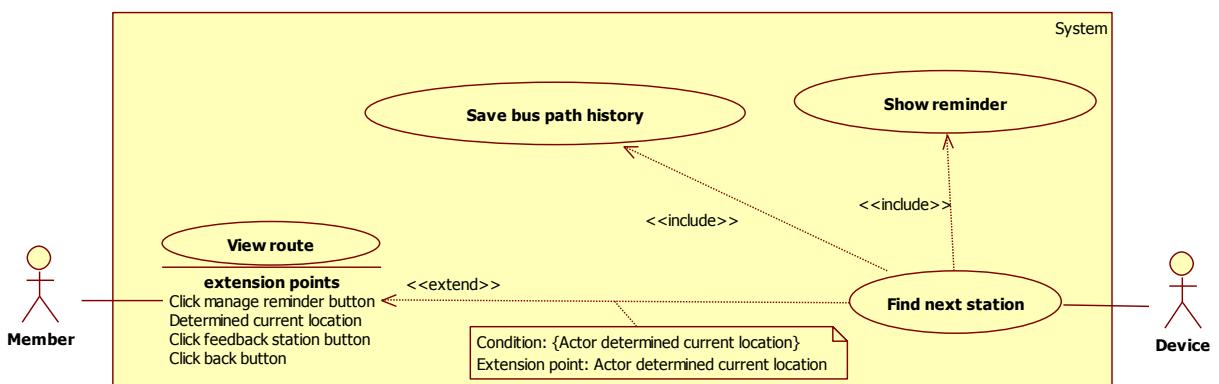


Figure 11 <Member> Find next station

Use Case Specification

USE CASE – TMST007												
Use Case No.	TMST007	Use Case Version	2.0									
Use Case Name	Find next station											
Author	Tran Gia Quoc Hung											
Date	17/04/2015	Priority	High									
Actor:	<ul style="list-style-type: none"> - Member, Device 											
Summary:	<ul style="list-style-type: none"> - This use case allows member to know which station is the next based on their current location. - Device is used to get GPS data. 											
Goal:	<ul style="list-style-type: none"> - Next station is found. 											
Triggers:	<ul style="list-style-type: none"> - Member chooses “theo dõi” button. 											
Preconditions:	<ul style="list-style-type: none"> - Guest is logged in as Member. - Member must enable GPS on device or know current address. 											
Post condition:	<ul style="list-style-type: none"> - Success: Next station is found. - Fail: Show error message. 											
Main Success Scenario:	<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: left; padding: 5px;">Step</th><th style="text-align: left; padding: 5px;">Actor Action</th><th style="text-align: left; padding: 5px;">System Response</th></tr> </thead> <tbody> <tr> <td style="padding: 10px;">1</td><td style="padding: 10px;"> Member taps “theo dõi” button on show route page. [Alternative 1,2] </td><td style="padding: 10px;"> <ul style="list-style-type: none"> - System gets current location of member’s device and calculate which path is nearest, the next station is the station that path lead to. - System keeps tracking for location of member’s device each time GPS is changed and shows next station continuously. <p>[Exception 1]</p> </td></tr> </tbody> </table>			Step	Actor Action	System Response	1	Member taps “theo dõi” button on show route page. [Alternative 1,2]	<ul style="list-style-type: none"> - System gets current location of member’s device and calculate which path is nearest, the next station is the station that path lead to. - System keeps tracking for location of member’s device each time GPS is changed and shows next station continuously. <p>[Exception 1]</p>			
Step	Actor Action	System Response										
1	Member taps “theo dõi” button on show route page. [Alternative 1,2]	<ul style="list-style-type: none"> - System gets current location of member’s device and calculate which path is nearest, the next station is the station that path lead to. - System keeps tracking for location of member’s device each time GPS is changed and shows next station continuously. <p>[Exception 1]</p>										
Alternative Scenario:	<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: left; padding: 5px;">No</th><th style="text-align: left; padding: 5px;">Actor Action</th><th style="text-align: left; padding: 5px;">System Response</th></tr> </thead> <tbody> <tr> <td style="padding: 10px;">1</td><td style="padding: 10px;"> Member taps current location button on show route page. </td><td style="padding: 10px;"> System gets current location of member’s device and calculate which path is nearest, the next station is the station that path lead to. [Exception 2] </td></tr> <tr> <td style="padding: 10px;">2</td><td style="padding: 10px;"> Member chooses one location as current location. </td><td style="padding: 10px;"> System gets location data and calculate which path is nearest, the next station is the station that path lead to. [Exception 2] </td></tr> </tbody> </table>			No	Actor Action	System Response	1	Member taps current location button on show route page.	System gets current location of member’s device and calculate which path is nearest, the next station is the station that path lead to. [Exception 2]	2	Member chooses one location as current location.	System gets location data and calculate which path is nearest, the next station is the station that path lead to. [Exception 2]
No	Actor Action	System Response										
1	Member taps current location button on show route page.	System gets current location of member’s device and calculate which path is nearest, the next station is the station that path lead to. [Exception 2]										
2	Member chooses one location as current location.	System gets location data and calculate which path is nearest, the next station is the station that path lead to. [Exception 2]										
Exceptions:												

No	Actor Action	System Response
1	Next station cannot be found while tracking.	System keeps using recent next station for 3 tracking times. If next station is then found, system run normally. [Exception 2]
2	Next station cannot be found.	Show message: “Không thể tìm thấy trạm kế tiếp, có thể xe buýt đang đi một lộ trình khác với bình thường. Vui lòng thử lại sau.”

Relationships: extended by View route (Actor determined current location), includes Set reminder by location, includes Save travelled bus path history.

Business Rules:

- System is intended to receive GPS data each 2 seconds.
- Method to find next station by GPS:
 - o Location will be taken.
 - o Calculate distance from the location to projection point of location onto each bus path in route.
 - o The bus path which has minimum distance to location and has projection point of location on it is then taken.
 - o Next station is the station that bus path is heading to.
- Method to find next station by location: each location has its' coordinate values. Coordinate values is taken to do method likes find next station by GPS.
- If there are more than one next station is found, system shows popup for member to choose which is correct bus routes. The chosen bus route will be prioritized.
- When app is running background, a notification may show instead of popup.
- When next station is found:
 - o System searches for suitable reminder and shows it.
 - o If user has tapped “theo dõi” button: If user enable save history mode, time to travel between 2 station (a bus path) is saved automatically.

2.3.1.8 <Member> Show reminder

Use case diagram

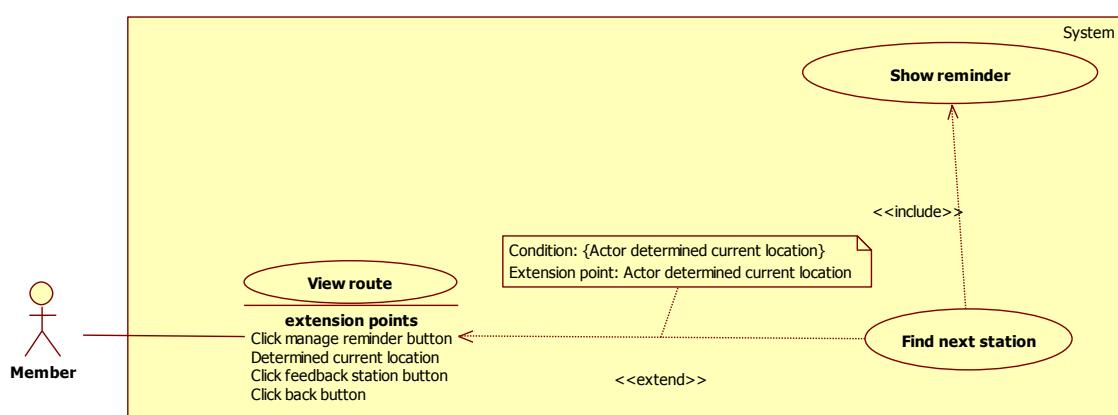


Figure 12 <Member> Show reminder

Use Case Specification

USE CASE – TMST008												
Use Case No.	TMST008	Use Case Version	2.0									
Use Case Name	Show reminder											
Author	Tran Gia Quoc Hung											
Date	17/04/2015	Priority	High									
Actor:	<ul style="list-style-type: none"> - Member 											
Summary:	<ul style="list-style-type: none"> - This use case demonstrate how system shows reminders. 											
Goal:	<ul style="list-style-type: none"> - Reminder is shown. 											
Triggers:	<ul style="list-style-type: none"> - Next station is found. 											
Preconditions:	<ul style="list-style-type: none"> - Guest is logged in as Member. - Member has added at least one reminder. - Next station is found. 											
Post condition:	<ul style="list-style-type: none"> - Success: Reminder is registered. - Fail: Do nothing. 											
Main Success Scenario:	<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="background-color: #cccccc;">Step</th><th style="background-color: #cccccc;">Actor Action</th><th style="background-color: #cccccc;">System Response</th></tr> </thead> <tbody> <tr> <td>1</td><td>Member determined current location.</td><td> System find next station. System selects suitable reminders from reminder list based on next station information and shows on a popup: <ul style="list-style-type: none"> - Title: "Next bus station" - Content: "Sắp tới trạm..." - "Báo lại": button - "Tắt": button </td></tr> <tr> <td>2</td><td>Member taps "Tắt" button. [Alternative 1]</td><td>System close popup.</td></tr> </tbody> </table>			Step	Actor Action	System Response	1	Member determined current location.	System find next station. System selects suitable reminders from reminder list based on next station information and shows on a popup: <ul style="list-style-type: none"> - Title: "Next bus station" - Content: "Sắp tới trạm..." - "Báo lại": button - "Tắt": button 	2	Member taps "Tắt" button. [Alternative 1]	System close popup.
Step	Actor Action	System Response										
1	Member determined current location.	System find next station. System selects suitable reminders from reminder list based on next station information and shows on a popup: <ul style="list-style-type: none"> - Title: "Next bus station" - Content: "Sắp tới trạm..." - "Báo lại": button - "Tắt": button 										
2	Member taps "Tắt" button. [Alternative 1]	System close popup.										
Alternative Scenario:	<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="background-color: #cccccc;">No</th><th style="background-color: #cccccc;">Actor Action</th><th style="background-color: #cccccc;">System Response</th></tr> </thead> <tbody> <tr> <td>1</td><td>Member taps "Báo lại" button.</td><td>System will set shown reminders at next station.</td></tr> </tbody> </table>			No	Actor Action	System Response	1	Member taps "Báo lại" button.	System will set shown reminders at next station.			
No	Actor Action	System Response										
1	Member taps "Báo lại" button.	System will set shown reminders at next station.										
Exceptions: N/A												
Relationships: included from Find next station.												
Business Rules:	<ul style="list-style-type: none"> - Each reminder can only be shown once each time member views a route, unless member taps "Báo lại" button. - Suitable reminders are reminders which have set at found next station. Suitable reminders are reminders which haven't been shown. - System prioritize latest reminders. If popup has opened before and another reminder need to be shown, popup data will be updated. 											

- If there are more than one reminder at station, popup's content will list all names, each once.

2.3.1.9 <Member> Set reminder by average time

Use case diagram

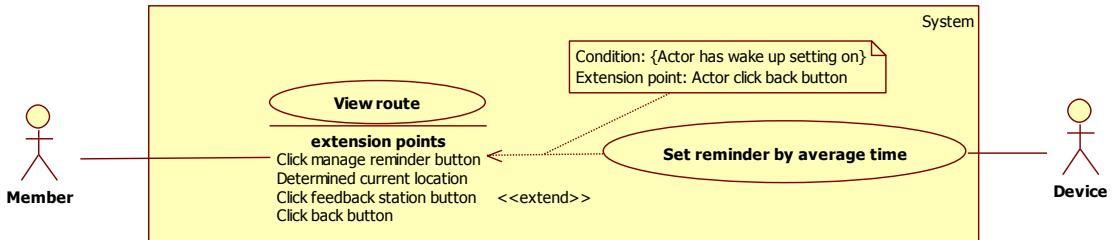


Figure 13 <Member> Set reminder by average time

Use Case Specification

USE CASE - TMST009

Use Case No.	TMST009	Use Case Version	2.0
Use Case Name	Set reminder by average time		
Author	Tran Gia Quoc Hung		
Date	17/04/2015	Priority	Normal

Actor:

- Member, Device

Summary:

- This use case allows member set reminder based on average time.
- Reminder is registered in device. Device will show reminder later.

Goal:

- Reminder is registered on device with begin time at a calculated time.

Triggers:

- Member wants to set reminder without tracking location.

Preconditions:

- Guest is logged in as Member
- Member has at least one reminder added.
- Next station has been found.

Post condition:

- **Success:** Reminder is registered on device.
- **Fail:** Show error message.

Main Success Scenario:

Step	Actor Action	System Response
1	Member taps on "Nhắc nhở" button on show route page.	System navigates to reminder page which shows "đánh thức" checkbox.
2	Member checks "đánh thức" checkbox and taps back button on device.	System activate "set reminder by average time" feature. System navigate back to show route page.
3	Member taps on back button from show route page.	<ul style="list-style-type: none"> - System takes next station property. - System calculates relative time needed to travel from current

		<p>location to each suitable reminder's location.</p> <ul style="list-style-type: none"> - System registers suitable reminders on device based on calculated time. <p>[Exception 1]</p>
--	--	--

Alternative: N/A

Exception:

No	Actor Action	System Response
1	Reminder cannot be set on device.	System will try to register later.

Relationships: extended by View route (Actor click back button)

Business Rules:

- Average time to travel from current location to reminded location = time to travel from current location to next station + sum of average times to travel on each bus path heading to reminded station.
- If average attribute of bus path is available, it is used. Otherwise average time will be calculated by formula: average time = distance of bus path / default average speed. Default average speed is 30km/h (8.33m/s).
- In own route: only reminders of running bus route are set.
- Suitable reminders are reminders which haven't been shown or set.
- Device is in charge of alert reminder, system is only registered reminder on device. Each reminder is distinguished by name for each application.
- Reminders will be registered on device with following attributes: Name: id of reminder, Title: "Next bus station", content: "Sắp tới trạm...", begin time: calculated time, expiration time: 60 minutes from calculated time.
- When members click on alerted reminder, members will be navigated to show route map with corresponding information of reminders.

2.3.1.10 <Member> Search location

Use case diagram



Figure 14 <Member> Search location

Use Case Specification

USE CASE - TMST010			
Use Case No.	TMST010	Use Case Version	2.0
Use Case Name	Search location		
Author	Tran Gia Quoc Hung		
Date	17/04/2015	Priority	Normal

Actor:

- Member

Summary:

- This use case allows member to search location.

Goal:

- Search results is shown.

Triggers:

- Member types in search location textbox.

Preconditions:

- Guest is logged in as Member.

Post condition:

- **Success:** List of suitable result is shown.
- **Fail:** Do nothing.

Main Success Scenario:

Step	Actor Action	System Response
1	Member types in search location textbox.	<p>System searches for suitable location from: saved location in database, server location from server, bus station in database, address by service or map offline.</p> <p>Shows list of results, each item includes:</p> <ul style="list-style-type: none"> - Name of location. - Address of location. <p>[Alternative 1]</p>

Alternative:

No	Actor Action	System Response
1	No result is found.	System shown an item with name “Không có địa điểm nào được tìm thấy” in list of result.

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

- In offline mode, result is get from: location in database, bus station in database, address in map offline.

2.3.1.11 <Member> Save bus path history**Use case diagram**

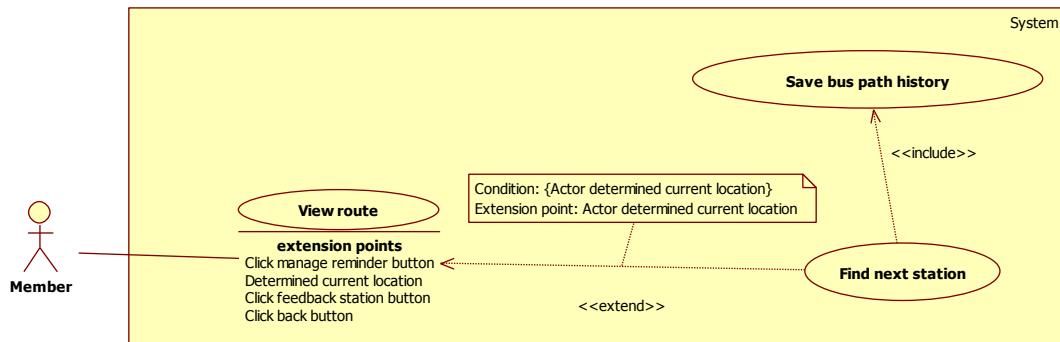


Figure 15 <Member> Save bus path history

Use Case Specification

USE CASE - TMST011									
Use Case No.	TMST011	Use Case Version	2.0						
Use Case Name	Save bus path history								
Author	Tran Gia Quoc Hung								
Date	17/04/2015	Priority	Normal						
Actor:	<ul style="list-style-type: none"> - Member 								
Summary:	<ul style="list-style-type: none"> - This use case demonstrate how system save bus path history. 								
Goal:	<ul style="list-style-type: none"> - Bus path history is saved. 								
Triggers:	<ul style="list-style-type: none"> - Member uses tracking function. 								
Preconditions:	<ul style="list-style-type: none"> - Guest is logged in as Member. - Member has enabled “lưu lại lịch sử đường đi” option. - Member uses tracking function. 								
Post condition:	<ul style="list-style-type: none"> - Success: Reminder is registered. - Fail: Do nothing. 								
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>Member taps “theo dõi” button</td><td> When next station property is changed, saves bus path with attributes: <ul style="list-style-type: none"> - BusPathID: id of last bus path. - Created time: current time. - Travelled time: number of seconds needed to travel from previous station to current station. </td></tr> </tbody> </table>			Step	Actor Action	System Response	1	Member taps “theo dõi” button	When next station property is changed, saves bus path with attributes: <ul style="list-style-type: none"> - BusPathID: id of last bus path. - Created time: current time. - Travelled time: number of seconds needed to travel from previous station to current station.
Step	Actor Action	System Response							
1	Member taps “theo dõi” button	When next station property is changed, saves bus path with attributes: <ul style="list-style-type: none"> - BusPathID: id of last bus path. - Created time: current time. - Travelled time: number of seconds needed to travel from previous station to current station. 							
Exceptions: N/A									
Relationships: included from Find next station.									
Business Rules:	<ul style="list-style-type: none"> - To ensure accuracy, history can only be saved if entire bus path is travelled while tracking. 								

2.3.1.12 <Member> Upload bus path history

Use case diagram

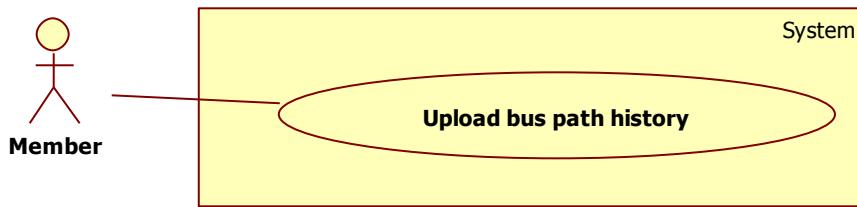


Figure 16 <Member> Upload bus path history

Use Case Specification

USE CASE - TMST012												
Use Case No.	TMST012	Use Case Version	2.0									
Use Case Name	Upload bus path history											
Author	Tran Gia Quoc Hung											
Date	17/04/2015	Priority	Normal									
Actor:	- Member											
Summary:	- This use case allows member upload bus path history to server.											
Goal:	- Member's bus path history is sent to server.											
Triggers:	- Member click “đăng lên” button.											
Preconditions:	- Guest login as Member.											
Post condition:	<ul style="list-style-type: none"> - Success: Member's bus path history is sent to server. - 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>Member clicks on bus path history item.</td><td>System shows menu includes “đăng lên” option.</td></tr> <tr> <td>2</td><td>Member clicks “đăng lên” option.</td><td> <ul style="list-style-type: none"> - System sends data to server. - System shows message and delete bus path history item from database. <p>[Exception 1]</p> </td></tr> </tbody> </table>			Step	Actor Action	System Response	1	Member clicks on bus path history item.	System shows menu includes “đăng lên” option.	2	Member clicks “đăng lên” option.	<ul style="list-style-type: none"> - System sends data to server. - System shows message and delete bus path history item from database. <p>[Exception 1]</p>
Step	Actor Action	System Response										
1	Member clicks on bus path history item.	System shows menu includes “đăng lên” option.										
2	Member clicks “đăng lên” option.	<ul style="list-style-type: none"> - System sends data to server. - System shows message and delete bus path history item from database. <p>[Exception 1]</p>										
Alternative: N/A												
Exception:	<table border="1"> <thead> <tr> <th>No</th><th>Actor Action</th><th>System Response</th></tr> </thead> <tbody> <tr> <td>1</td><td>Cannot connect to server.</td><td>Show error message “Không thể kết nối được với server. Xin vui lòng thử lại sau.”</td></tr> </tbody> </table>			No	Actor Action	System Response	1	Cannot connect to server.	Show error message “Không thể kết nối được với server. Xin vui lòng thử lại sau.”			
No	Actor Action	System Response										
1	Cannot connect to server.	Show error message “Không thể kết nối được với server. Xin vui lòng thử lại sau.”										
Relationships: N/A												
Business Rules:												

- History of bus path is used to calculate actual average time on server. Average time is used to set reminder by average time.
- When member upload successfully, member will get point immediately.
- History data does not include member's account data to ensure privacy.
- When history is uploaded successfully on server, server will calculate new average time and send it back. Then average time of uploaded bus path is updated in database.

2.3.1.13 <Member> Edit own route

Use case diagram

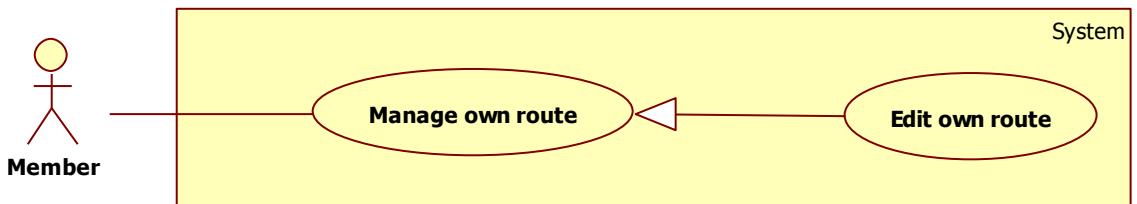


Figure 17 <Member> Edit own route

Use Case Specification

USE CASE – TMST013			
Use Case No.	TMST013	Use Case Version	2.0
Use Case Name	Edit own route		
Author	Tran Gia Quoc Hung		
Date	17/04/2015	Priority	Normal
Actor:			
- Member			
Summary:			
- This use case allows member edit own route.			
Goal:			
- Own route is edited.			
Triggers:			
- Member clicks “chỉnh sửa” button.			
Preconditions:			
- Guest is logged in as Member.			
Post condition:			
- Success: Own route is edited. - Fail: Show error message.			
Main Success Scenario:			
Step	Actor Action	System Response	
1	Member clicks on own route item in route list from main page.	Systems shows menu which includes “Chỉnh sửa” option.	
2	Member chooses “Chỉnh sửa” option.	<ul style="list-style-type: none"> - Systems navigates to create route page, list of bus route is shown, each item includes: <ul style="list-style-type: none"> - Bus number: label. - Turn: label. - Direction: label. 	

		<ul style="list-style-type: none"> - Load data of own route to map. Update list of bus route based on added bus route. - Application bar button: “lưu lại” button.
3	Member clicks on one bus route which has not been added.	System shows menu includes “thêm tuyến này” option.
4	Member choose “thêm tuyến này” option.	<ul style="list-style-type: none"> - Systems adds bus route to own route’s list of bus route. - Systems shows bus route’s full information.
5	Member clicks on one bus route which has not been added.	System shows menu includes “thêm tuyến này” option.
6	Member chooses “thêm tuyến này” option.	<ul style="list-style-type: none"> - Systems adds bus route to own route’s list of bus route. - For previous added bus route: Hide full information, show short information. - For current added bus route: Add bus route to own route’s list of bus route. Show bus route’s full information. - Auto adjust start station and stop station between 2 bus routes.
7	Member chooses one location as “bắt đầu”.	<ul style="list-style-type: none"> - System marks the location with “bắt đầu” template. - For previous added bus route: Hide full information, show short information. - System auto adjust start station of first bus route. - System draws walking lines from “Trạm lên” to start station of first bus route.
8	Member chooses one location as “kết thúc”.	<ul style="list-style-type: none"> - System marks the location with “kết thúc” template. - System auto adjusts stop station of last bus route. - System draws walking lines from “Trạm xuống” to stop station of last bus route.
9	Member clicks “Lưu lại” button. [Alternative 1]	System shows textbox for member to enter name of own route. Old name is shown in textbox.
10	Member clicks save button.	<ul style="list-style-type: none"> - System stored own route data to database. - System navigates to main page. [Exception 1,2,3]

Alternative:

- Alternative 1:

No	Actor Action	System Response
1	Member clicks on one bus route which has been added.	<p>System shows menu includes</p> <ul style="list-style-type: none"> - “Bỏ tuyến này” option. - “Chỉnh sửa” option. - “Đổi lượt” option. - If chosen bus route is not first bus route in own route: “Chuyển lên” option. - If chosen bus route is not last bus route in own route: “Chuyển xuống” option.
2	Member chooses “Bỏ tuyến này” option. [Alternative 2,3,4,5]	<ul style="list-style-type: none"> - System removes chosen bus route from own route. - System auto adjusts other bus routes in own route.

- Alternative 2

No	Actor Action	System Response
1	Member chooses “Chỉnh sửa” option.	<ul style="list-style-type: none"> - For previous added bus route: Hide full information, show short information. - For current added bus route: Show bus route's full information. - Change application bar buttons: “lưu lại” button, “hủy bỏ” button.
2	<ul style="list-style-type: none"> - Member changes “Trạm lên” and “Trạm xuống” of bus route. - Member clicks “lưu lại” button. <p>[Alternative 6]</p>	<ul style="list-style-type: none"> - For previous added bus route: Hide full information, show short information. - System auto adjusts bus routes in own route. - Change application bar buttons to main application bar: “lưu lại” button.

- Alternative 3

No	Actor Action	System Response
1	Member chooses “Đổi lượt” option.	<ul style="list-style-type: none"> - System removes chosen bus route from own route. Systems removes chosen bus route data from map. - System add bus route with same number as chosen bus route but different turn to own route. System adds new bus route data to map. - System auto adjusts bus routes in own route.

- Alternative 4

No	Actor Action	System Response
1	Member chooses “Chuyển lên” option.	<ul style="list-style-type: none"> - System swap data of chosen bus route and previous chosen bus route in own route. - System auto adjusts bus routes in own route.

- Alternative 5

No	Actor Action	System Response
1	Member chooses “Chuyển xuống” option.	<ul style="list-style-type: none"> - System swap data of chosen bus route and next chosen bus route in own route. - System auto adjusts bus routes in own route.

- Alternative 6

No	Actor Action	System Response
1	Member chooses “hủy bỏ” option.	<ul style="list-style-type: none"> - System hides full information, show short information of chosen bus route.

Exception:

No	Actor Action	System Response
1	Name is empty.	System shows message “Xin vui lòng điền tên đường đi.”
2	Length of name is over 100 characters.	System shows message “Tên đường đi tối đa 100 ký tự.”
3	Route cannot be saved.	System shows message “Đường đi không thể lưu lại. Xin vui lòng thử lại sau.”

Relationships: N/A

Business Rules:

- Own route can only be created from bus routes, not from other own routes. Own route must have one bus route to be saved. Maximum number of bus routes in own route is 5.
- Each bus route can only be added to own route once.
- Start location and/or stop location may be chosen by searching or holding on map. Start location and stop location are optional.
- Data of bus routes in own route is reference data: only start station index and stop station index of each bus route are stored for each bus route in own route.
- When saving data of own route to database:
 - If “Bắt đầu” / “Kết thúc” location is server location and has not been downloaded, location is downloaded to database first.
 - If “Bắt đầu” / “Kết thúc” location is address and has not been saved, location is saved to database first.
- In system, different own routes can have same name.
- Full information of bus route: All bus route line and all stations. Short information of bus route: Line between from “Trạm lên” and “Trạm xuống” properties of bus route. Only “Trạm lên” station and “Trạm xuống” station.
- Automatically adjust bus route: When one bus route in own route is changed, or start location / stop location is changed, system will adjust creating route automatically:
 - If start location is changed: Find nearest “Trạm lên” of first bus route.
 - If bus route is first bus route: Find nearest “Trạm lên” from start location. Find nearest “Trạm xuống” to next bus route.

- If bus route is last bus route: Find nearest “Trạm xuống” to stop location.
- If bus route is neither first nor last bus route: Find nearest “Trạm lên” to previous bus route. Find nearest “Trạm xuống” to next bus route.
- If stop location is changed: Find nearest “Trạm xuống” of last bus route.
- When auto adjust bus route: if member has chosen “Trạm xuống” or “Trạm lên” manually for one bus route, system will prioritize it. Otherwise system will automatically adjust.
- When one bus route is chosen in bus route list, the bus route with the same number but different turn is hidden, the chosen bus route is marked with different color. When one bus route is removed, both will be shown in default color.
- In online mode, walking line is searched by service. In offline mode, walking line is searched by query offline map. If walking line cannot be shown, system won't show anything.
- In online mode, list of routes show all route. In offline mode, only saved bus routes are listed.
- In online mode, if created route includes any bus routes which has not been saved to database, system will save these bus routes to database automatically.

2.3.1.14 <Member> Delete own route

Use case diagram

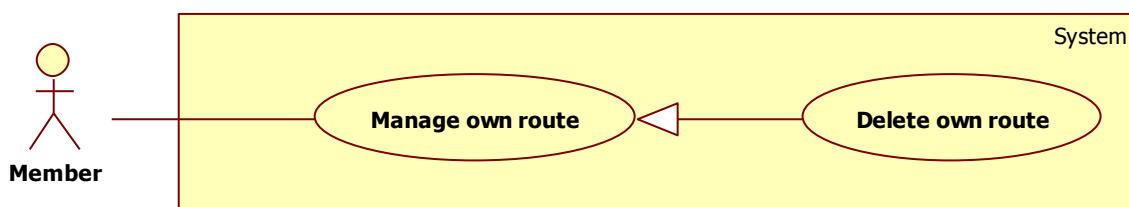


Figure 18 <Member> Delete own route

Use Case Specification

USE CASE - TMST014			
Use Case No.	TMST014	Use Case Version	2.0
Use Case Name	Delete own route		
Author	Tran Gia Quoc Hung		
Date	17/04/2015	Priority	Normal
Actor:	<ul style="list-style-type: none"> - Member 		
Summary:	<ul style="list-style-type: none"> - This use case allows member delete own route. 		
Goal:	<ul style="list-style-type: none"> - Own route is deleted. 		
Triggers:	<ul style="list-style-type: none"> - Member clicks “xóa” button. 		

Preconditions:

- Guest is logged in as Member.

Post condition:

- **Success:** Own route is deleted.
- **Fail:** Do nothing.

Main Success Scenario:

Step	Actor Action	System Response
1	Member clicks on own route item in route list from main page.	Systems shows menu which includes "Xóa" option.
2	Member chooses "Xóa" option.	System shows message to confirm member which includes "Đồng ý" and "Hủy bỏ" option.
3	Member chooses "Đồng ý" option. [Alternative 1]	System closes message and deletes own route data from database. List of route is reloaded.

Alternative:

No	Actor Action	System Response
1	Member chooses "Hủy bỏ" option.	System closes message.

Exception: N/A

Relationships: N/A

Business Rules:

- Data to be deleted: own route data, own route's reminders.
- Location which is "Bắt đầu" and "Kết thúc" is kept in database (if any).

2.3.1.15 <Member> Edit reminder

Use case diagram

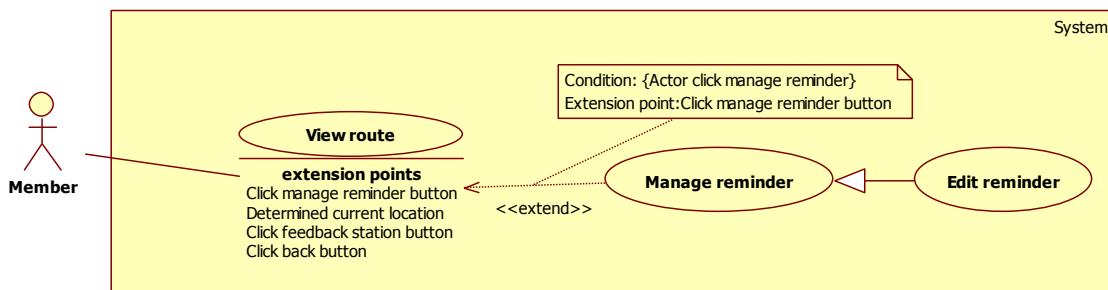


Figure 19 <Member> Edit reminder

Use Case Specification

USE CASE - TMST015			
Use Case No.	TMST015	Use Case Version	2.0
Use Case Name	Edit reminder		
Author	Tran Gia Quoc Hung		
Date	17/04/2015	Priority	Normal

Actor:

- Member

Summary:

- This use case allows member to edit reminder information.

Goal:

- Reminder is edited.

Triggers:

- Member taps on “nhắc nhở” button.

Preconditions:

- Guest is logged in as Member.

Post condition:

- **Success:** Reminder is edited.
- **Fail:** Do nothing.

Main Success Scenario:

Step	Actor Action	System Response
1	Member taps on “nhắc nhở” button on show route page.	<p>System navigates to reminder page. List of reminder is loaded, each item includes:</p> <ul style="list-style-type: none"> - Name of set station or location: label. - Reminder is on or off: Toggle button. - “Xem chi tiết” link.
2	Member taps on toggle button of reminder item to turn reminder on or off. [Alternative 1]	System saves change of reminder. [Exception 1]

Alternative:

- Alternative 1

Step	Actor Action	System Response
1	Member taps on “chi tiết” link.	<p>System shows more details of reminder:</p> <ul style="list-style-type: none"> - “Báo trước”: set of 3 radio buttons: “1 trạm”, “2 trạm” and “3 trạm”
2	Member chooses one radio button.	System save change of reminder.

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

- List of reminder is only includes related reminders of chosen route.
- Details of reminder is varied, based on:
 - Second station and third station: no “chi tiết” link.
 - Fourth station: “chi tiết” has “1 trạm” radio button or “2 trạm” radio button only.
- If reminder is changed to off: Reminded station’s color will change to normal station’s color.

2.3.1.16 <Member> Remove reminder

Use case diagram

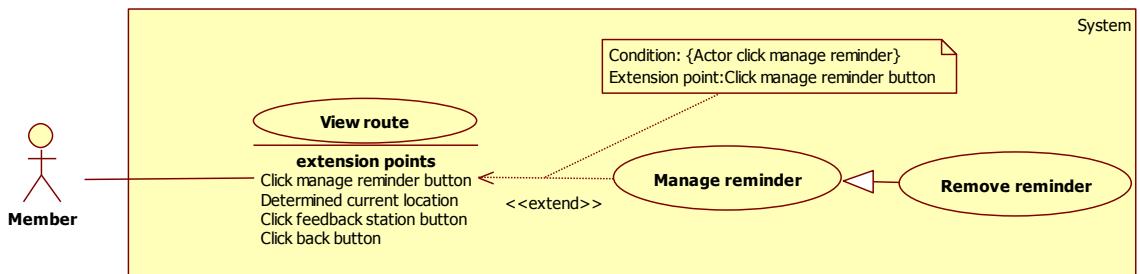


Figure 20 <Member> Remove reminder

Use Case Specification

USE CASE – TMST016

Use Case No.	TMST016	Use Case Version	2.0
Use Case Name	Remove reminder		
Author	Tran Gia Quoc Hung		
Date	17/04/2015	Priority	Normal

Actor:

- Member

Summary:

- This use case allows member to remove reminder.

Goal:

- Reminder is removed.

Triggers:

- Member holds on reminder item.

Preconditions:

- Guest is logged in as Member.

Post condition:

- **Success:** Reminder is removed.
- **Fail:** Do nothing.

Main Success Scenario:

Step	Actor Action	System Response
1	Member holds on reminder item.	System shows menu “xóa”
2	Member taps on “xóa”.	System shows message box “Xóa nhắc nhở này ?” with option “Xóa” and “Không”.
3	Member chooses option “Xóa”. [Alternative 1]	System deletes reminder in database and in chosen route’s list of reminder. [Exception 1]

Alternative:

No	Actor Action	System Response
1	Member chooses option “Không”.	System does nothing.

Exception: N/A

Relationships: N/A

Business Rules:

- Related locations won’t be deleted.

2.3.1.17 <Member> Feedback station

Use case diagram

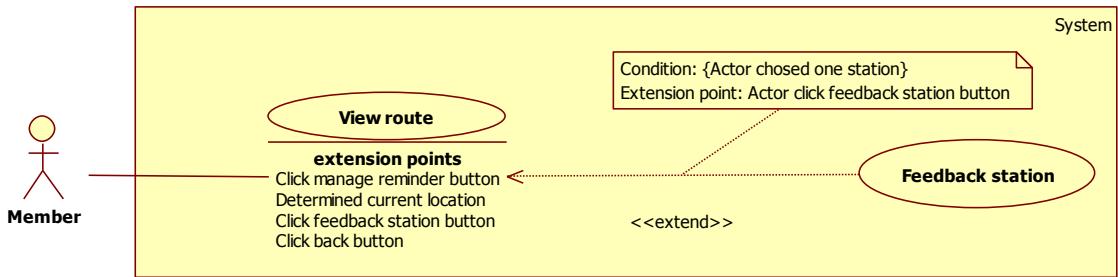


Figure 21 <Member> Feedback station

Use Case Specification

USE CASE – TMST017												
Use Case No.	TMST017	Use Case Version	2.0									
Use Case Name	Feedback station											
Author	Tran Gia Quoc Hung											
Date	17/04/2015	Priority	Low									
Actor:	<ul style="list-style-type: none"> - Member 											
Summary:	<ul style="list-style-type: none"> - This use case allows member to feedback for a station. 											
Goal:	<ul style="list-style-type: none"> - Member's feedback is sent. 											
Triggers:	<ul style="list-style-type: none"> - Member chooses “Góp ý trạm xe” option. 											
Preconditions:	<ul style="list-style-type: none"> - Guest is logged in as Member. 											
Post condition:	<ul style="list-style-type: none"> - Success: Feedback is sent. - 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>Member holds on station marker.</td><td>System shows menu contains “Góp ý trạm xe” option.</td></tr> <tr> <td>2</td><td>Member chooses “Góp ý địa điểm” option.</td><td> System navigates to feedback for bus station page. Page includes: <ul style="list-style-type: none"> - Station's name, station's address: labels. - “Đổi địa chỉ”: radio button, default is checked, with new address textbox: max 120 characters, required if “Đổi địa chỉ” radio button is chosen. - “Thêm tên khác”: radio button with new name textbox: max 100 characters, required if “Thêm tên khác” radio button is chosen. </td></tr> </tbody> </table>			Step	Actor Action	System Response	1	Member holds on station marker.	System shows menu contains “Góp ý trạm xe” option.	2	Member chooses “Góp ý địa điểm” option.	System navigates to feedback for bus station page. Page includes: <ul style="list-style-type: none"> - Station's name, station's address: labels. - “Đổi địa chỉ”: radio button, default is checked, with new address textbox: max 120 characters, required if “Đổi địa chỉ” radio button is chosen. - “Thêm tên khác”: radio button with new name textbox: max 100 characters, required if “Thêm tên khác” radio button is chosen.
Step	Actor Action	System Response										
1	Member holds on station marker.	System shows menu contains “Góp ý trạm xe” option.										
2	Member chooses “Góp ý địa điểm” option.	System navigates to feedback for bus station page. Page includes: <ul style="list-style-type: none"> - Station's name, station's address: labels. - “Đổi địa chỉ”: radio button, default is checked, with new address textbox: max 120 characters, required if “Đổi địa chỉ” radio button is chosen. - “Thêm tên khác”: radio button with new name textbox: max 100 characters, required if “Thêm tên khác” radio button is chosen. 										

		<p>chosen.</p> <ul style="list-style-type: none"> - “Sửa thành địa điểm hiện tại”: radio button - “Trạm xe không còn tồn tại”: radio button. - “Gửi đi”: button - “Quay lại”: button
3	Member chooses “Đổi địa chỉ”, fill in new address textbox and click “Gửi đi” [Alternative 1,2,3,4]	System gets data, then send to server. [Exception 1,2,3]

Alternative Scenario:

No	Actor Action	System Response
1	Member chooses “Thêm tên khác”, fill in new name textbox and click “Gửi đi” [Alternative 4]	System gets data, then send to server. [Exception 1,4,5]
2	Member chooses “Sửa thành địa điểm hiện tại”, then clicks “Gửi đi” button. [Alternative 4]	System get data and send to server. [Exception 1,6]
3	Member chooses “Trạm xe không còn tồn tại”, then taps “Gửi đi” button. [Alternative 4]	System get data and send to server. [Exception 1]
4	Member taps “Quay lại” button.	System navigates back to previous page.

Exceptions:

No	Actor Action	System Response
1	Cannot connect to server.	System saves feedback to database. Show error message: “Góp ý hiện tại chưa gửi được và sẽ được lưu lại. Xin vui lòng gửi lại sau. Xin chân thành cảm ơn.”
2	New address is empty.	Show error message: “Vui lòng điền địa chỉ mới”.
3	Length of new address is over 120 characters	Show error message: “Địa chỉ mới tối đa 120 ký tự.”
4	New name is empty.	Show error message: “Vui lòng điền tên mới”.
5	Length of new name is over 100 characters.	Show error message: “Tên mới tối đa 100 ký tự”.
6	GPS is not available.	Show message: “Không thể lấy được địa điểm hiện tại. Vui lòng bật chức năng định vị.”

Relationships: extended by View route (Actor click feedback station button).

Business Rules:

- Feedback of member does not need to provide exact data in textbox, staff need to examine the changes in real.
- When staff approved a feedback, member will get point. Staff approve a feedback when it contains useful information.
- If feedback cannot be sent to server, it is saved in database. Member can send it later.
- Member can send as many feedbacks as they want.
- Member can send many feedbacks for the same stations. But staff may deactivate user if they send too many unuseful data.

2.3.1.18 <Member> Feedback bus route

Use case diagram

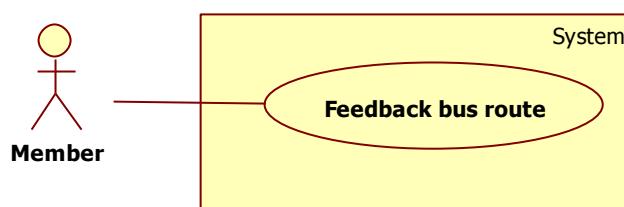


Figure 22 <Member> Feedback bus route

Use Case Specification

USE CASE – TMST018									
Use Case No.	TMST018	Use Case Version	2.0						
Use Case Name	Feedback bus route								
Author	Tran Gia Quoc Hung								
Date	17/04/2015	Priority	Low						
Actor:	<ul style="list-style-type: none"> - Member 								
Summary:	<ul style="list-style-type: none"> - This use case allows member to feedback for a bus route. 								
Goal:	<ul style="list-style-type: none"> - Member's feedback is sent to server. 								
Triggers:	<ul style="list-style-type: none"> - Member click "Góp ý thông tin xe buýt". 								
Preconditions:	<ul style="list-style-type: none"> - Guest is logged in as Member. 								
Post condition:	<ul style="list-style-type: none"> - Success: Member's feedback is sent to server. - Fail: Show message error. 								
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>Member clicks on "Góp ý thông tin xe buýt" menu on view route page.</td><td> System navigates to feedback which shown bus' number, bus' name and offer member 2 options to report <ul style="list-style-type: none"> - "Thêm tên khác": radio button, default is checked, with a new name textbox max </td></tr> </tbody> </table>			Step	Actor Action	System Response	1	Member clicks on "Góp ý thông tin xe buýt" menu on view route page.	System navigates to feedback which shown bus' number, bus' name and offer member 2 options to report <ul style="list-style-type: none"> - "Thêm tên khác": radio button, default is checked, with a new name textbox max
Step	Actor Action	System Response							
1	Member clicks on "Góp ý thông tin xe buýt" menu on view route page.	System navigates to feedback which shown bus' number, bus' name and offer member 2 options to report <ul style="list-style-type: none"> - "Thêm tên khác": radio button, default is checked, with a new name textbox max 							

		<p>length 100 characters, required if this option is chosen.</p> <ul style="list-style-type: none"> - “Xe này không còn chạy”: radio button. <p>And two buttons:</p> <ul style="list-style-type: none"> - “Gửi đi”: button - “Quay lại”: button
2	Member chooses “Thêm tên khác” option, fill in new name textbox and taps on “Gửi đi” button. [Alternative 1,2]	System get data and send to server. [Exception 1,2,3]

Alternative:

No	Actor Action	System Response
1	Member chooses “Xe này không còn chạy” option and taps on “Gửi đi” button. [Alternative 2]	System get data and send to server. [Exception 1]
2	Member taps on “Quay lại” button.	System turns back on previous page.

Exception:

No	Actor Action	System Response
1	Cannot connect to server.	System saves feedback. Show error message: “Góp ý hiện tại chưa gửi được và sẽ được lưu lại. Xin vui lòng gửi lại sau. Xin chân thành cảm ơn.”
2	New name is empty.	Show error message: “Vui lòng điền tên khác”.
3	Length of new name is over 100 characters	Show error message: “Tên khác tối đa 100 ký tự.”

Relationships: N/A

Business Rules:

- Feedback of member does not need to provide exact data in textbox, staff need to examine the changes in real.
- When staff approved a feedback, member will get point. Staff approve a feedback when it contains useful information.
- If feedback cannot be sent to server, it is saved in database instead. Member can send it later.
- Member can send as many feedbacks as they want.
- Member can send many feedbacks for the same stations. But staff may deactivate user if they send too many unuseful data.

2.3.1.19 <Member> Feedback location

Use case diagram

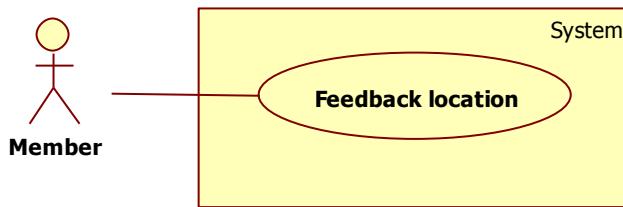


Figure 23 <Member> Feedback location

Use Case Specification

USE CASE - TMST019												
Use Case No.	TMST019	Use Case Version	2.0									
Use Case Name	Feedback location											
Author	Tran Gia Quoc Hung											
Date	17/04/2015	Priority	Low									
Actor:	<ul style="list-style-type: none"> - Member 											
Summary:	<ul style="list-style-type: none"> - This use case allows member to feedback for a location. 											
Goal:	<ul style="list-style-type: none"> - Member's feedback is sent. 											
Triggers:	<ul style="list-style-type: none"> - Member clicks on “Góp ý địa điểm” button. 											
Preconditions:	<ul style="list-style-type: none"> - Guest is logged in as Member. 											
Post condition:	<ul style="list-style-type: none"> - Success: Feedback is sent. - 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>Member holds on location marker.</td><td>System shows menu contains “Góp ý địa điểm” option.</td></tr> <tr> <td>2</td><td>Member taps on marker and chooses “góp ý địa điểm”.</td><td> System navigates to feedback for location page. Page includes: <ul style="list-style-type: none"> - Location's name, location's labels. - “Đổi địa chỉ”: radio button, default is checked, with new address textbox: max 120 characters, required if “Đổi địa chỉ” radio button is chosen. - “Thêm tên khác”: radio button with new name textbox: max 80 characters, required if “Thêm tên khác” radio button is chosen. - “Sửa thành địa điểm hiện tại”: </td></tr> </tbody> </table>			Step	Actor Action	System Response	1	Member holds on location marker.	System shows menu contains “Góp ý địa điểm” option.	2	Member taps on marker and chooses “góp ý địa điểm”.	System navigates to feedback for location page. Page includes: <ul style="list-style-type: none"> - Location's name, location's labels. - “Đổi địa chỉ”: radio button, default is checked, with new address textbox: max 120 characters, required if “Đổi địa chỉ” radio button is chosen. - “Thêm tên khác”: radio button with new name textbox: max 80 characters, required if “Thêm tên khác” radio button is chosen. - “Sửa thành địa điểm hiện tại”:
Step	Actor Action	System Response										
1	Member holds on location marker.	System shows menu contains “Góp ý địa điểm” option.										
2	Member taps on marker and chooses “góp ý địa điểm”.	System navigates to feedback for location page. Page includes: <ul style="list-style-type: none"> - Location's name, location's labels. - “Đổi địa chỉ”: radio button, default is checked, with new address textbox: max 120 characters, required if “Đổi địa chỉ” radio button is chosen. - “Thêm tên khác”: radio button with new name textbox: max 80 characters, required if “Thêm tên khác” radio button is chosen. - “Sửa thành địa điểm hiện tại”: 										

		radio button - “Địa điểm không còn tồn tại”: radio button. - “Gửi đi”: button - “Quay lại”: button
3	Member chooses “Đổi địa chỉ”, fill in new address textbox and click “Gửi đi” [Alternative 1,2,3,4]	System gets data, then send to server. [Exception 1,2,3]

Alternative Scenario:

No	Actor Action	System Response
1	Member chooses “Thêm tên khác”, fill in new name textbox and click “Gửi đi” [Alternative 4]	System gets data, then send to server. [Exception 1,4,5]
2	Member chooses “Sửa thành địa điểm hiện tại”, then clicks “Gửi đi” button. [Alternative 4]	System get data and send to server. [Exception 1,6]
3	Member chooses “Địa điểm không còn tồn tại”, then taps “Gửi đi” button. [Alternative 4]	System get data and send to server. [Exception 1]
4	Member taps “Quay lại” button.	System navigates back to previous page.

Exceptions:

No	Actor Action	System Response
1	Cannot connect to server.	System saves feedback to database. Show error message: “Góp ý hiện tại chưa gửi được và sẽ được lưu lại. Xin vui lòng gửi lại sau. Xin chân thành cảm ơn.”
2	New address is empty.	Show error message: “Vui lòng điền địa chỉ mới”.
3	Length of new address is over 120 characters	Show error message: “Địa chỉ mới tối đa 120 kí tự.”
4	New name is empty.	Show error message: “Vui lòng điền tên mới”.
5	Length of new name is over 80 characters.	Show error message: “Tên mới tối đa 80 kí tự”.
6	GPS is not available.	Show message: “Không thể lấy được địa điểm hiện tại. Vui lòng bật chức năng định vị.”

Relationships: N/A

Business Rules:

- Feedback of member does not need to provide exact data in textbox, staff need

to examine the changes in real.

- When staff approved a feedback, member will get point. Staff approve a feedback when it contains useful information.
- If feedback cannot be sent to server, it is saved in database instead. Member can send it later.
- Member can send as many feedbacks as they want.
- Member can send many feedbacks for the same stations. But staff may deactivate user if they send too many unuseful data.

2.3.2. <Staff> Overview Use Case

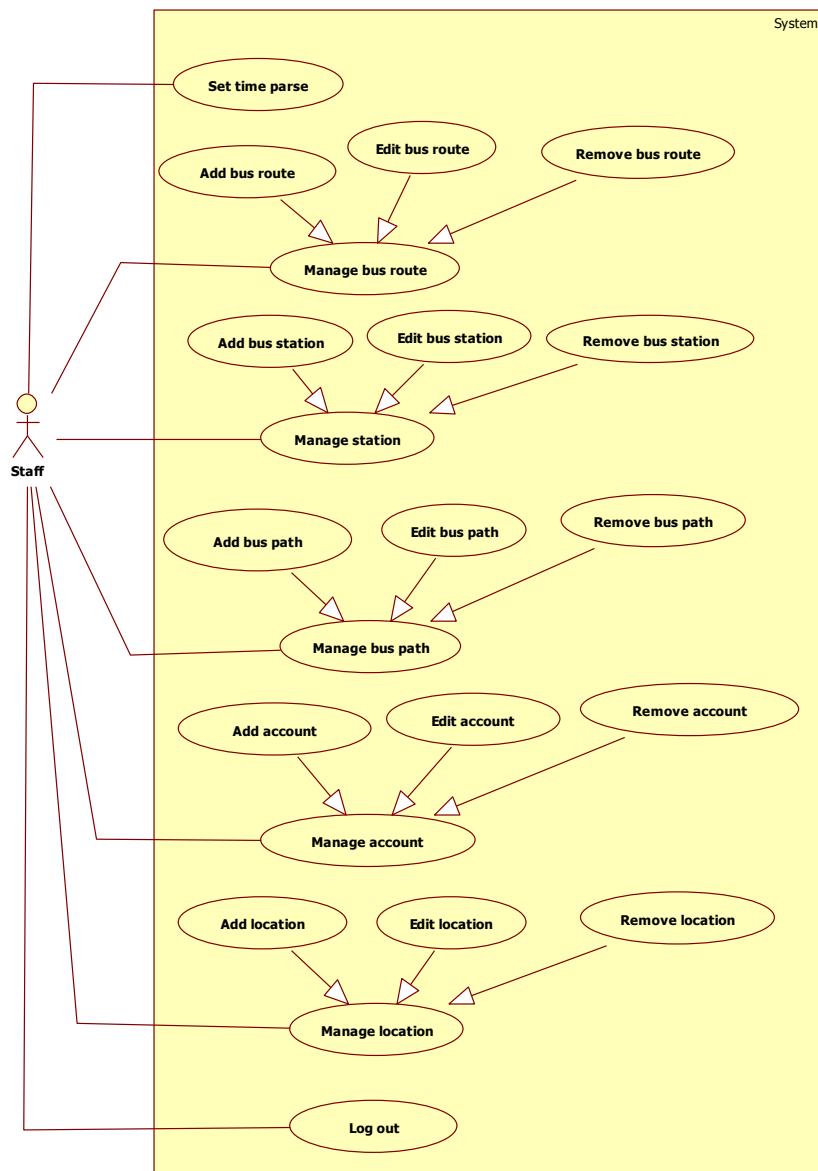


Figure 24 <Staff> Overview Use Case

2.3.2.1 <Staff> Edit bus route

Use case diagram

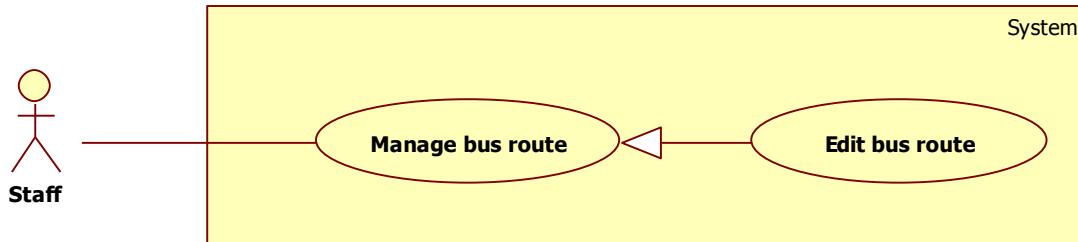


Figure 25 <Staff> Edit bus route

Use case specification

USE CASE - TMST020			
Use Case No.	TMST020	Use Case Version	2.0
Use Case Name	Edit bus route		
Author	Nguyen Quoc Dat		
Date	17/04/2015	Priority	Normal
Actor:			
- Staff.			
Summary:			
- This use case allows staff to edit BusRoute information.			
Goal:			
- BusRoute information is updated in database.			
Triggers:			
- Staff clicks button "Sửa".			
Preconditions:			
- Guest must login with role Staff.			
Post condition:			
- Success: BusRoute information is updated in database.			
- Fail: Show error message.			
Main Success Scenario:			
Step	Actor Action	System Response	
1	Staff clicks button "Sửa".	Show pop-up and get bus information in database for staff edit include the following information: - "Số Xe" : textbox, min length: 2,max length: 10, required - "Tên Xe" : textbox, min length: 5,max length: 100, required - "Tên Khác" : textbox. - "Xác Nhận": button - "Hủy": button	
2	Staff input information and clicks button "Đồng Ý" [Alternative 1]	Update new information into database and show success message. [Exception 1,2,3]	

Alternative Scenario:

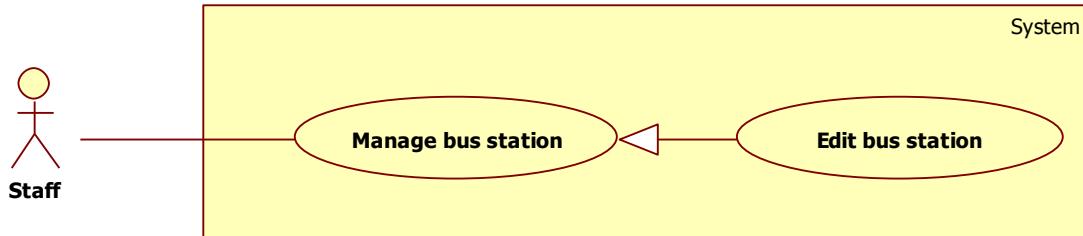
No	Actor Action	System Response
1	Staff clicks button "Hủy"	Pop-up disappear and nothing change

Exceptions:

No	Actor Action	System Response
1	Length of "Tên Xe" is not in range	Show error message: "Độ dài tên xe phải từ 5-100 ký tự! Xin vui lòng sửa lại."
2	Length of "Số Xe" is not in range	Show error message: "Độ dài số xe phải từ 2-10 ký tự! Xin vui lòng sửa lại."
3	"Tên Xe" is not match regular expression	Show error message: "Tên Xe phải có dạng bến xe đi-bến xe cuối"

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

- ID of BusRoute will be changed to form of "M-" + bus number + "-" + turn. (M for "Manual")
- Status is set to "Normal", last update date is set to current time.

2.3.2.2 <Staff> Edit bus station**Use case diagram****Figure 26 <Staff> Edit bus station****Use case specification****USE CASE – TMST021**

Use Case No.	TMST021	Use Case Version	2.0
Use Case Name	Edit bus station		
Author	Nguyen Quoc Dat		
Date	17/04/2015	Priority	Normal

Actor:

- Staff.

Summary:

- This use case allows staff to edit BusStation information.

Goal:

- Update BusStation information success to database..

Triggers:

- Staff clicks button “Sửa”.

Preconditions:

- Guest must login with role Staff.

Post condition:

- **Success:** BusStation information is updated to database.
- **Fail:** Show error message.

Main Success Scenario:

Step	Actor Action	System Response
1	Staff clicks button “Sửa”.	Show pop-up and get station information in database for staff edit include the following information: <ul style="list-style-type: none"> - “Tên Trạm” : textbox, min length: 5,max length: 80, required - “Tên Khác” : textbox. - “Kiểu” : drop down list, raw source (“Trạm Dừng”, “Nhà Chờ”),required - “Địa Chỉ” : textbox, min length: 5, max length: 120 - “Xe buýt tại trạm”: Multiselect, raw source from database-Bus table, required. - “Mã Trạm” : textbox, min length: 1, max length: 20, required - “Tiếp”: button - “Hủy”: button
2	Staff input information and clicks button “Tiếp” [Alternative 1]	Show new pop-up include: <ul style="list-style-type: none"> - Map - “Xác Nhận”: button - “Hủy”: button - “Ẩn Hiện Các Trạm Khác”: button [Exception 1,2,3,4,5,6]
3	Staff choose location for new station and clicks button “Xác Nhận” [Alternative 1,2]	Update new information into database and show success message.

Alternative Scenario:

No	Actor Action	System Response
1	Staff clicks button “Hủy”	Pop-up disappear and nothing change.

2	Staff clicks button “Ẩn Hiện Các Trạm Khác”	If show will show others BusStation. If hide will hide others BusStation.
---	---	---

Exceptions:

No	Actor Action	System Response
1	Length of “Tên Trạm” is not in range	Show error message: “Độ dài tên trạm phải từ 5-80 ký tự! Xin vui lòng sửa lại.”
2	“Kiểu” is not specified	Show error message: “Hãy chọn kiểu! Xin vui lòng chọn lại.”
3	“Xe Buýt Tại Trạm” is not specified	Show error message: “Hãy chọn xe buýt tại trạm! Xin vui lòng chọn lại.”
4	Length of “Địa Chỉ” is not in range	Show error message: “Độ dài địa chỉ phải từ 5-120 ký tự! Xin vui lòng sửa lại.”
5	Length of station “Mã Trạm” is not in range	Show error message: “Độ dài mã trạm phải từ 1-20 ký tự! Xin vui lòng sửa lại.”
6	Staff doesn't choose location for new station	Show error message: “Hãy chọn địa điểm của trạm trên bản đồ! Xin vui lòng chọn lại.”

Relationships: N/A

Business Rules:

- Status is set to “Normal”, last update date is set to current time.
- ID of BusStation will be changed to form of “M-“ + id. (M for “Manual”)

2.3.2.3 <Staff> Edit bus path

Use case diagram

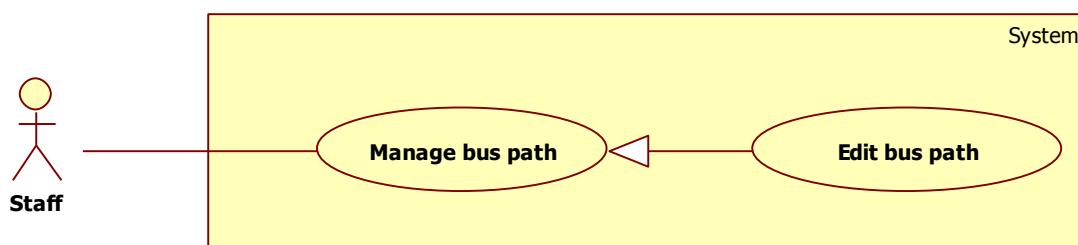


Figure 27 <Staff> Edit bus path

Use case specification

USE CASE – TMST022			
Use Case No.	TMST022	Use Case Version	2.0
Use Case Name	Edit bus path		
Author	Nguyen Quoc Dat		
Date	17/04/2015	Priority	Normal
Actor:	- Staff.		

Summary:

- This use case allows staff to edit BusPath information.

Goal:

- BusPath information have been updated to database.

Triggers:

- Staff clicks button “Sửa”.

Preconditions:

- Guest must login with role Staff.

Post condition:

- **Success:** BusPath information is updated to database.
- **Fail:** Show error message.

Main Success Scenario:

Step	Actor Action	System Response
1	Staff clicks button “Sửa”.	Show pop-up and get BusPath information in database for staff edit include the following information: <ul style="list-style-type: none"> - “Trạm Đầu” : drop down list, raw source from database- BusStation table, required - “Trạm Cuối” : drop down list, raw source from database- BusStation table, required - “Thứ tự tuyến” : textbox, number, required - “Số Xe Buýt”: drop down list, raw source from database- BusRoute table, required - “Tiếp”: button - “Hủy”: button
2	Staff input information and clicks button “Tiếp” [Alternative 1]	Show new pop-up include: <ul style="list-style-type: none"> - Map - “Xác Nhận”: button - “Hủy”: button [Exception 1,2,3,4]
3	Staff clicks button “Xác Nhận” [Alternative 1]	Update new information into database and show success message

Alternative Scenario:

No	Actor Action	System Response
1	Staff clicks button “Hủy”	Pop-up disappear and nothing change

Exceptions:

No	Actor Action	System Response
1	“Trạm Đầu” is not specified	Show error message: “ Hãy chọn trạm đầu! Xin vui lòng chọn lại.”

2	"Trạm Cuối" is not specified	Show error message: "Hãy chọn trạm cuối! Xin vui lòng chọn lại."
3	"Thứ Tự Tuyến" is not a number	Show error message: "Thứ tự tuyến phải là số! Xin vui lòng sửa lại."
4	"Số Xe Buýt" is not specified	Show error message: "Xin vui lòng chọn số xe buýt."

Relationships: N/A

Business Rules:

- "Trạm Đầu" drop down list, "Trạm Cuối" drop down list raw source from database-BusStation table.
- System will remove BusStation ID which selected in "Trạm Đầu" drop down list from "Trạm Cuối" drop down list. System will remove BusStation ID which selected in "Trạm Cuối" drop down list from "Trạm Đầu" drop down list.
- System auto catch latitude and longitude of StationFrom, StationTo to get path details and distance of BusPath.
- "Số Xe Buýt" drop down list raw source from database-BusRoute table.
- ID of BusPath will be in form of "M-" + automatic increased id. (M for Manual)

2.3.2.4 <Staff> Add location

Use case diagram

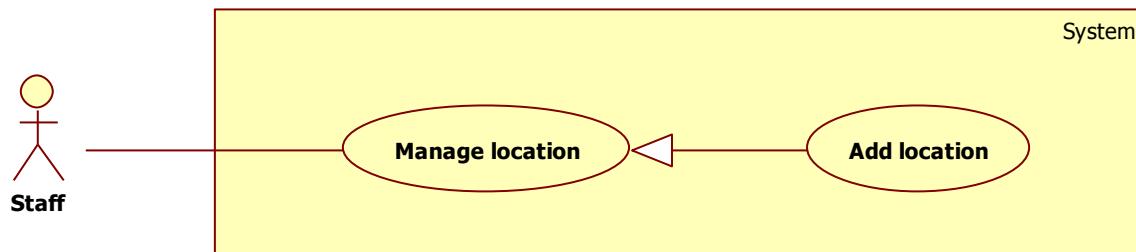


Figure 28 <Staff> Add location

Use case specification

USE CASE – TMST023			
Use Case No.	TMST023	Use Case Version	2.0
Use Case Name	Add location		
Author	Nguyen Quoc Dat		
Date	17/04/2015	Priority	Normal
Actor:			
- Staff.			
Summary:			
- This use case allows staff to add new Location.			
Goal:			
- New Location is added to database.			
Triggers:			
- Staff clicks button "Tạo Mới".			
Preconditions:			

- Guest must login with role Staff.

Post condition:

- **Success:** New Location information is inserted to database.
- **Fail:** Nothing is changed in database.

Main Success Scenario:

Step	Actor Action	System Response
1	Staff clicks button “Tạo Mới”	Show pop-up for staff input include the following information: <ul style="list-style-type: none"> - “Tên Địa Điểm” : textbox, min length: 5, max length: 100, required - “Địa chỉ” : textbox, min length: 5, max length: 120, required - “Thể Loại”: drop down list(Ẩm Thực, Cơ Quan, Giải Trí, Giáo Dục), required - “Tên Khác” : textbox. - “Tiếp”: button - “Hủy”: button
2	Staff input information and clicks button “Tiếp” [Alternative 1]	Show new pop-up include: <ul style="list-style-type: none"> - Map - “Xác Nhận”: button - “Hủy”: button - “Ẩn Hiện Các Địa Điểm Khác”: button [Exception 1,2,3]
3	Staff choose location for new station and clicks button “Xác Nhận” [Alternative 1,2]	Send data of new Location to server to add to database. [Exception 5]

Alternative Scenario:

No	Actor Action	System Response
1	Staff clicks button “Hủy”	Pop-up disappear and nothing change.
2	Staff clicks button “Ẩn Hiện Các Trạm Khác”	If show will show others Location. If hide will hide others Location.

Exceptions:

No	Actor Action	System Response
1	Length of “Tên Địa Điểm” is not in range	Show error message: “Độ dài tên địa điểm phải từ 5-100 ký tự! Xin vui lòng sửa lại.”
2	Length of “Địa Chỉ” is not in range	Show error message: “Độ dài tên địa chỉ phải từ 5-120 ký tự! Xin vui lòng

		sửa lại."
3	"Thể Loại' is not specified	Show error message: "Hãy chọn thể loại! Xin vui lòng chọn lại."
4	Staff doesn't choose coordinates for Location	Show error message: "Hãy chọn địa điểm trên bản đồ! Xin vui lòng sửa lại."
Relationships: N/A		
Business Rules:		
- Status is set to "Normal", last update date is set to current time.		

3. Software System Attribute

3.1. Usability

3.1.1. Graphic User Interface

- Texts which are name of buttons, messages, and labels are written in Vietnamese.

3.1.2. Usability

- Staff should need less than one week of training to be productive with the system.

3.1.3. Installation

- The system should be easy to deploy. Customer can deploy successfully and learn to configure, maintain the system within one day of training.
- The attached manual guide must be clear. Installation can be done by users themselves.

3.2. Reliability

- Latest update date of application in device can be viewed by user. Bus route data is updated daily, user can update data of application.
- User can give feedbacks. Staff can adjust data of system.

3.3. Availability

- In online mode, system provides all features for user.
- In offline mode, system provides most of main features for user: View route, search route, create route... with less accuracy in walking path related tasks.

3.4. Security

- Each role of user has a specific permission to interact with system.
- History of user's travels is uploaded without user's identity.

3.5. Maintainability

- The system is divided into separated modules.

3.6. Portability

- The system can be deployed into many type of servers those have IIS server.

3.7. Performance

- Search route should be within 1 minute each times.

4. Conceptual Diagram

- Because web and mobile application need different entities, so conceptual diagram on web will be different to conceptual diagram on mobile application. We decide to design two separated conceptual diagrams.
- Compare conceptual diagram on web and mobile:
 - Some entities is used in mobile application only but not on website: Reminder, Own route.
 - Some entities has the same name but different meanings:
 - Locations on web are often famous places for everyone to search. Locations on mobile are often personal places for member to search.
 - Bus Path History on web are meant to use for calculating average time of related bus path. Bus Path History on mobile are meant to store history data only.
 - Some entities has the same meanings but have to be appeared at both web and mobile application: bus route, bus path, bus station.
 - Loading one bus route includes both loading all related bus paths and related bus stations. If mobile application has to download those data each time member view a new route, process on mobile might be slowdown. Moreover, create route function and search route function's performance will be affected too.

4.1. Web Conceptual Diagram

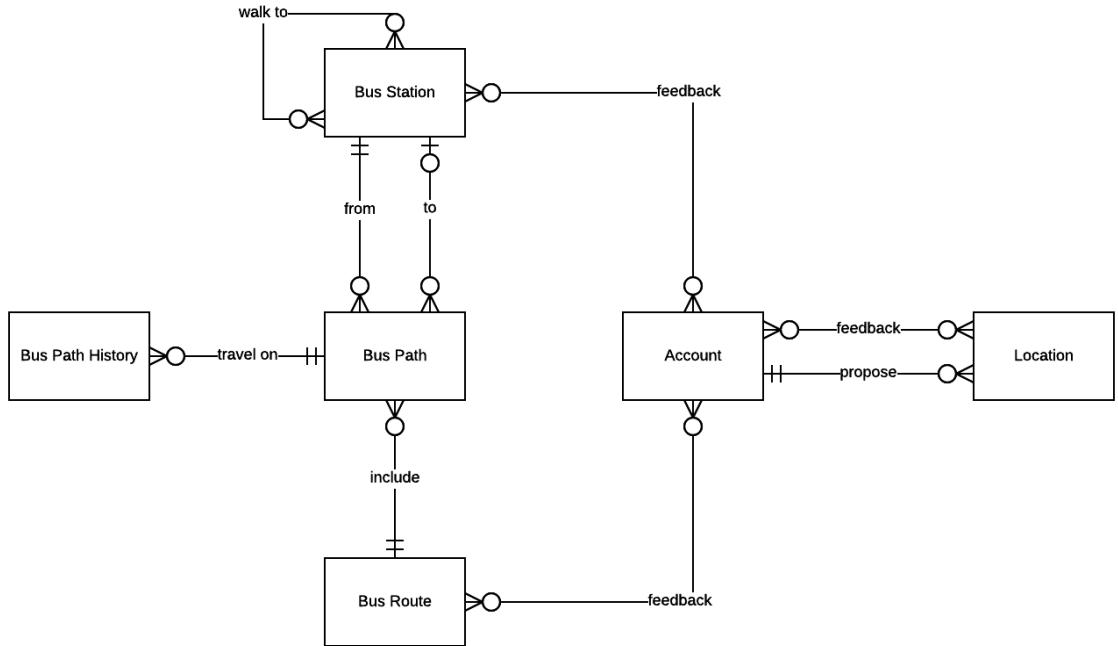


Figure 29 Web Conceptual Diagram

Data Dictionary

Entity Data dictionary: describe content of all entities	
Entity Name	Description
Account	Describe all accounts in the system.
Bus Path	Describe all bus paths in the system.
Bus Route	Describe all bus routes in the system.
Bus Station	Describe all bus stations in the system.
Location	Describe all locations in the system.
Bus Path History	Describe all histories of bus paths member sent to server.

4.2. Mobile Conceptual Diagram

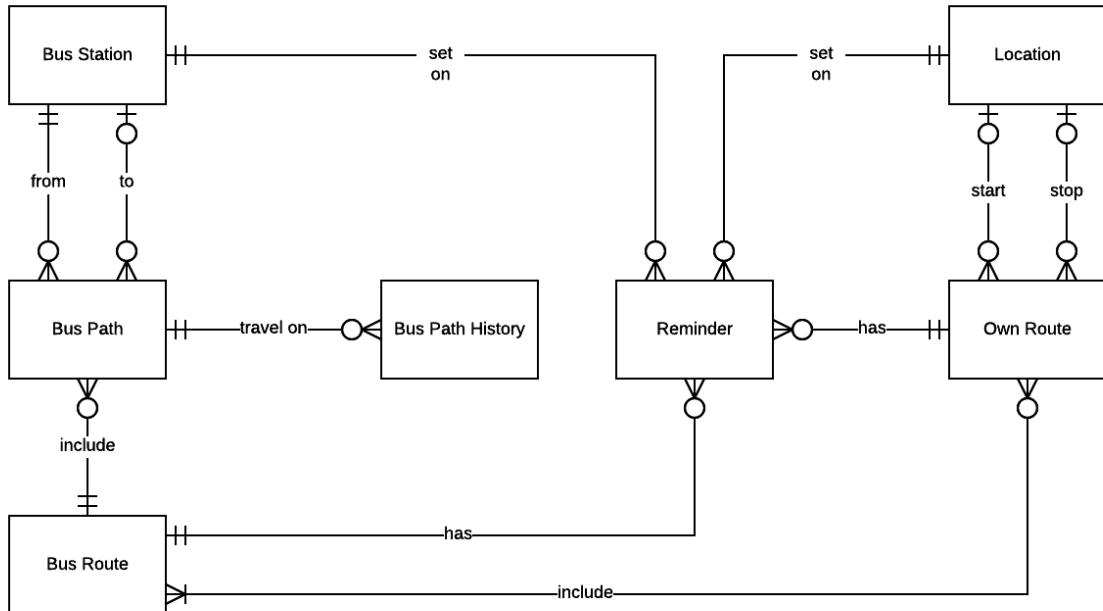


Figure 30 Mobile Conceptual Diagram

Data Dictionary

Entity Data dictionary: describe content of all entities	
Entity Name	Description
Bus Path	Describe all bus paths in the system.
Bus Route	Describe all bus routes in the system.
Bus Station	Describe all bus stations in the system.
Location	Describe all locations member saved.
Own Route	Describe all own route member created.
Reminder	Describe all reminders member created.
Bus Path History	Describe all histories member saved.

D. Software Design Description

1. Design Overview

- This document describes the technical and user interface design TMST 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:

- Section 2: gives an overall description of the system architecture design.
- Section 3: gives component diagrams that describe the connection and integration of the system.
- Section 4: gives the detail design description which includes class diagram, class explanation, activity diagram and sequence diagram to details the application functions.
- Section 5: describe screen design.
- Section 6: describe a fully attribute ERD.
- Section 7: describe algorithms.

2. System Architectural Design

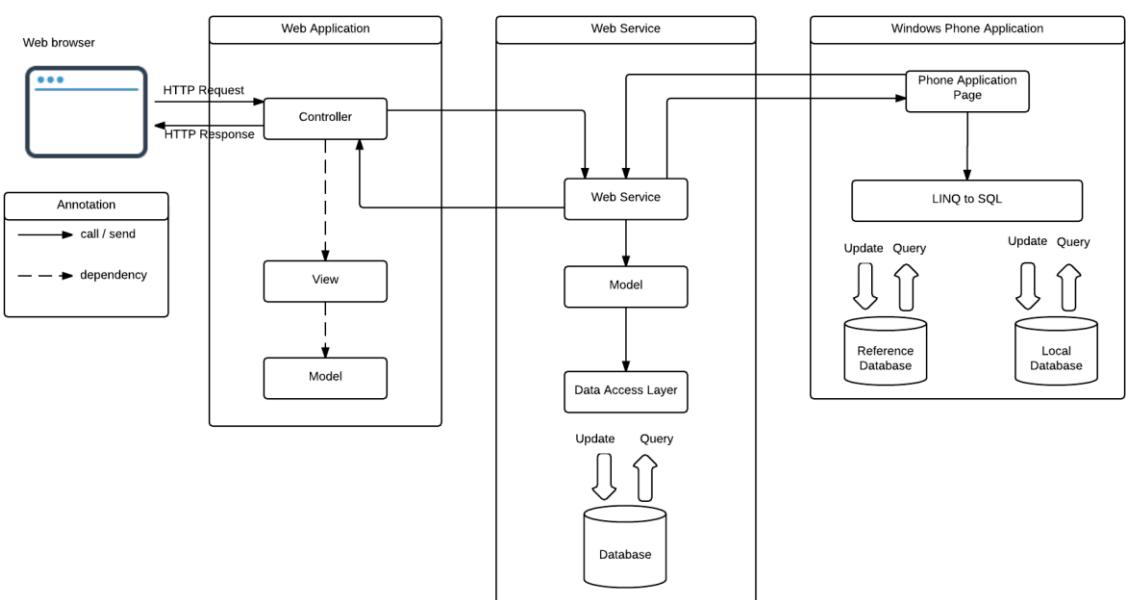


Figure 31 TMST System Architectural

2.1. Web Application Architecture Description

Web Application is developed from MVC architecture style because:

- Model: Easy to choose needed data, which is gotten from web service, to show on page, make data transferring more flexible.
- View: Freely and easy to design, customize page and show essential data.
- Controller: Easy to handle user's request: choose which view to show, which data is needed for view, get data from web service.

2.2. Web Service Architecture Description

- Web Service: Because Mobile application and Web application are separated, so web service is needed as a centre to get, update data in database. Web service makes it easier to change business logic.

2.3. Windows Phone Application Architecture Description

Mobile application is developed in Windows Phone OS because Windows Phone device can provide stable GPS when user is travelling. If user download map offline, GPS can be get even when device does not have internet connection, and map can be queried to get walking path and address offline, make it possible to run almost core function of Next Bus Station application in offline mode.

- Phone Application Page: Handle all events of user on a page.
- Reference Database: contains initial data which is needed for application (bus route data, bus station data and bus path data). Reference Database is separated from Local Database for updating application in future (just delete old reference database and use new one, do not need any further steps).
- Local Database: contains user's personal data.

3. Component Diagram

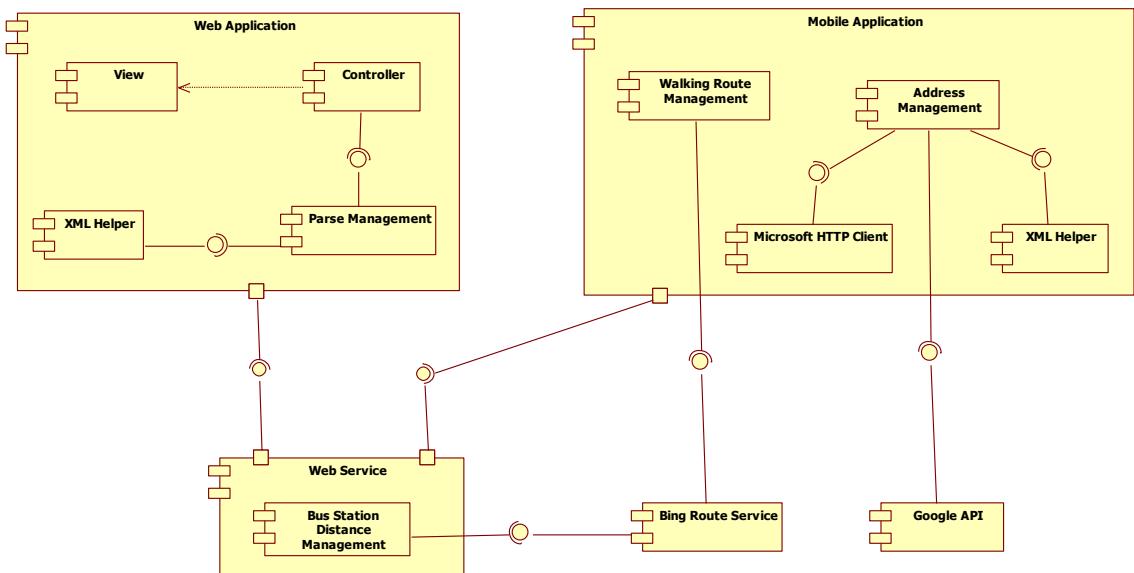


Figure 32 Component Diagram

Component dictionary: describe component	
Component Name	Description
View	All views of the system.
Controller	All controllers of the system.
Parse Management	Business logic processing for parsing function.
XML Helper	Handle XML processing, use in parsing function, get from https://www.nuget.org/packages/System.Xml.XPath.SL4/
Bus Station Distance Management	Business logic processing for create distance between bus stations.
Bing Route Service	Get route data, such as walking route, get from

	http://dev.virtualearth.net/webservices/v1/routeservice/routeService.svc/mex
Walking Route Management	Business logic processing for get route function.
Microsoft HTTP Client	Handle HTTP processing, get from https://www.nuget.org/packages/Microsoft.Net.Http
XML Helper	Handle XML processing, use in parsing function, get from https://www.nuget.org/packages/System.Xml.XPath.SL4/
Address Management	Business logic processing for search address function.
Google API	REST service: use to get address when user search address in online mode.

4. Detailed Description

4.1. Web Service Class Diagram

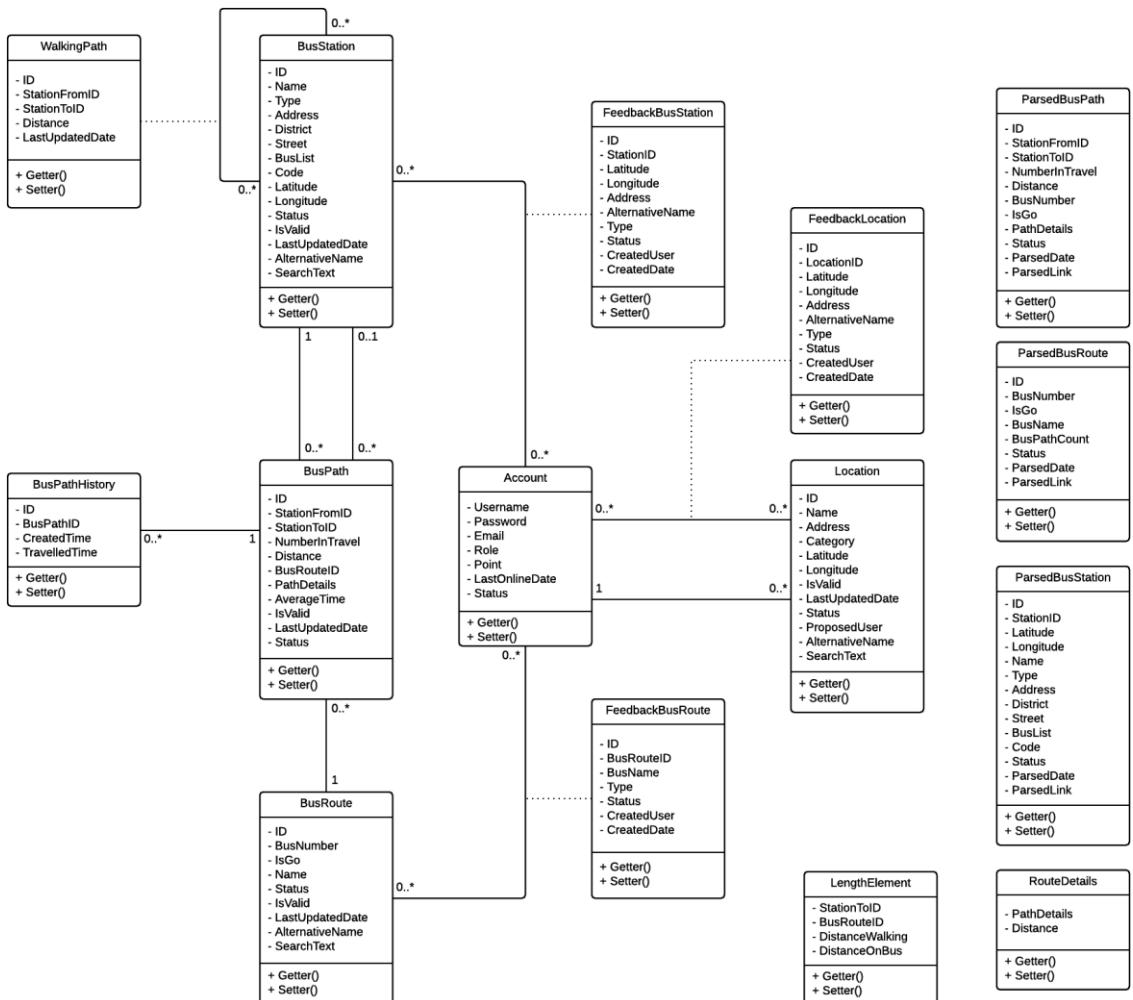


Figure 33 Web Service Class Diagram

4.2. Web Service Class Diagram Explanation

4.2.1. BusPath

Attribute

Attribute	Type	Visibility	Description
ID	String	Private	Unique identifier of each bus path.
StationFromID	String	Private	ID of station at which bus path begins.
StationToID	String	Private	ID of station at which bus path ends.
NumberInTravel	Integer	Private	The order of bus path in bus route.
Distance	Double	Private	Distance of bus path.
BusRouteID	String	Private	ID of bus route which contains bus path.
PathDetails	String	Private	Collection of coordinates which form the path.
AverageTime	Datetime	Private	Average time a bus has to spend on bus path.
IsValid	Boolean	Private	Validity of bus path.
LastUpdatedDate	DateTime	Private	Last time bus path is updated.
Status	String	Private	Status of bus path.

Method

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

4.2.2. BusRoute

Attribute

Attribute	Type	Visibility	Description
ID	String	Private	Unique identifier of each bus route.
BusNumber	String	Private	Number of bus.
IsGo	Boolean	Private	Type of bus, "Lượt đi" or "Lượt về".
Name	String	Private	Official name of bus.
Status	String	Private	Status of bus route.

IsValid	Boolean	Private	Validity of bus route.
LastUpdatedDate	DateTime	Private	Last time bus route is updated.
AlternativeName	String	Private	Other name of bus route.
SearchText	String	Private	Text which is used to search for bus.

Method

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

4.2.3. BusStation

Attribute

Attribute	Type	Visibility	Description
ID	String	Private	Unique identifier of each bus station.
Name	String	Private	Name of station
Type	String	Private	Type of station, “Nhà chờ” or “Trạm chờ” or “Bến xe buýt”.
Address	String	Private	Address of station.
District	String	Private	District of station.
Street	String	Private	Street of station.
BusList	String	Private	List of bus which is available at station.
Code	String	Private	Code of station.
Latitude	Double	Private	Latitude of station.
Longitude	Double	Private	Longitude of station.
Status	String	Private	Status of bus station.
IsValid	Boolean	Private	Validity of bus station.
LastUpdatedDate	DateTime	Private	Last time bus station is updated.
AlternativeName	String	Private	Other name of bus station.
SearchText	String	Private	Text which is used to search for station.

Method

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

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

4.2.4. WalkingPath

Attribute

Attribute	Type	Visibility	Description
ID	Integer	Private	Unique identifier of each walking path.
StationFromID	String	Private	ID of start station.
StationToID	String	Private	ID of end station
Distance	Double	Private	Walking distance from start station to end station.
LastUpdatedDate	DateTime	Private	Last time walking path is updated.

Method

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

4.2.5. Location

Attribute

Attribute	Type	Visibility	Description
ID	Integer	Private	Unique identifier of each location.
Name	String	Private	Name of location.
Address	String	Private	Address of location.
Category	String	Private	Category of location.
Latitude	Double	Private	Latitude of location.
Longitude	Double	Private	Longitude of location.
IsValid	Boolean	Private	Validity of bus location.
LastUpdatedDate	DateTime	Private	Last time bus location is updated.
Status	String	Private	Status of bus location.
ProposedUser	String	Private	User who proposed the location.
AlternativeName	String	Private	Other name of bus location.
SearchText	String	Private	Text which is used to search for location.

Method

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

4.2.6. FeedbackBusRoute

Attribute

Attribute	Type	Visibility	Description
ID	String	Private	Unique identifier of each feedback.
BusRouteID	String	Private	ID of bus route id which feedback is about.
BusName	String	Private	Proposed new name of bus in feedback.
Status	String	Private	Status of feedback
Type	String	Private	Type of feedback.
CreatedUser	String	Private	Username of user who send feedback.
CreatedDate	Datetime	Private	Created date of feedback.

Method

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

4.2.7. FeedbackBusStation

Attribute

Attribute	Type	Visibility	Description
ID	String	Private	Unique identifier of each feedback.
StationID	String	Private	ID of bus station id which feedback is about.
Latitude	Double	Private	Proposed new latitude of station in feedback.
Longitude	Double	Private	Proposed new longitude of station in feedback.
Address	String	Private	Proposed new

			address of station in feedback.
AlternativeName	String	Private	Alternative name of bus station in feedback.
Type	String	Private	Type of feedback.
Status	String	Private	Status of feedback
CreatedUser	String	Private	Username of user who send feedback.
CreatedDate	Datetime	Private	Created date of feedback.

Method

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

4.2.8. FeedbackLocation

Attribute

Attribute	Type	Visibility	Description
ID	String	Private	Unique identifier of each feedback.
LocationID	Integer	Private	ID of location which feedback is about.
Latitude	Double	Private	Proposed new latitude of location in feedback.
Longitude	Double	Private	Proposed new longitude of location in feedback.
Address	String	Private	Proposed new address of location in feedback.
AlternativeName	String	Private	Alternative name of location in feedback.
Status	String	Private	Status of feedback.
Type	String	Private	Type of feedback.
CreatedUser	String	Private	Username of user who send feedback.
CreatedDate	Datetime	Private	Created date of feedback.

Method

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

4.2.9. BusPathHistory

Attribute

Attribute	Type	Visibility	Description
ID	Integer	Private	Unique identifier of each travel history bus path.
BusPathID	String	Private	ID of bus path which travel history details is about.
CreatedTime	Datetime	Private	Time which travel history details is created.
TravelledTime	Datetime	Private	Taken time to travel on bus path.

Method

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

4.2.10. Account

Attribute

Attribute	Type	Visibility	Description
Username	String	Private	Username of account.
Password	String	Private	Password of account.
Email	String	Private	Email of account.
Role	String	Private	Role of account. (Member / Staff).
Point	Integer	Private	Point of account.
LastOnlineDate	DateTime	Private	Last date account connect to server.
Status	String	Private	Status of account (Ban / Unban).

Method

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

4.2.11. ParsedBusPath

Attribute

Attribute	Type	Visibility	Description
ID	String	Private	Unique identifier of each parsed bus path.
StationFromID	String	Private	Parsed ID of station at which bus path begins.
StationToID	String	Private	Parsed ID of station at which bus path ends.
NumberInTravel	Integer	Private	The parsed order of bus path in bus route.
Distance	Double	Private	Parsed distance of bus path.
BusNumber	String	Private	Parsed number of bus of bus path.
IsGo	Boolean	Private	Parsed turn of bus of bus path.
PathDetails	String	Private	Parsed collection of coordinates which form the path.
Status	String	Private	Status of parsed bus path.
ParsedDate	Boolean	Private	Date of parse.
ParsedLink	DateTime	Private	Parsed link.

Method

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

4.2.12. ParsedBusRoute

Attribute

Attribute	Type	Visibility	Description
ID	String	Private	Unique identifier of each parsed bus route.
BusNumber	String	Private	Parsed number of bus.
IsGo	Boolean	Private	Parsed turn of bus route.

BusName	String	Private	Parsed official name of bus.
BusPathCount	Integer	Private	Number of parsed bus path belong to this bus route.
Status	String	Private	Status of parsed bus route.
ParsedDate	Boolean	Private	Date of parse.
ParsedLink	DateTime	Private	Parsed link.
Method			
Method	Return type	Visibility	Description
Getter	Attribute type	Public	Get attribute value
Setter	Void	Public	Set value of attribute

4.2.13. ParsedBusStation

Attribute

Attribute	Type	Visibility	Description
ID	String	Private	Unique identifier of each parsed bus station.
StationID	String	Private	ID of parsed bus station.
Latitude	Double	Private	Parsed latitude of station.
Longitude	Double	Private	Parsed longitude of station.
Name	String	Private	Parsed name of station
Type	String	Private	Parsed type of station, “Nhà chờ” or “Trụ chờ” or “Bến xe buýt”.
Address	String	Private	Parsed address of station.
District	String	Private	Parsed district of station.
Street	String	Private	Parsed street of station.
BusList	String	Private	Parsed list of bus which is available at station.
Code	String	Private	Parsed code of station.
Status	String	Private	Status of parsed bus station.
ParsedDate	Boolean	Private	Date of parse.

ParsedLink	DateTime	Private	Parsed link.
Method			
Method	Return type	Visibility	Description
Getter	Attribute type	Public	Get attribute value
Setter	Void	Public	Set value of attribute

4.2.14. RouteDetails

Attribute

Attribute	Type	Visibility	Description
PathDetails	String	Private	Collection of coordinates which form the path.
Distance	Double	Private	Distance of the path.

Method

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

4.2.15. LengthElement

Attribute

Attribute	Type	Visibility	Description
StationToID	String	Private	Station to which path travel.
BusRouteID	String	Private	Bus which is used to travel.
DistanceWalking	Double	Private	Distance if walking.
DistanceOnBus	Double	Private	Distance if travel by bus.

Method

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

4.3. Mobile Application Class Diagram

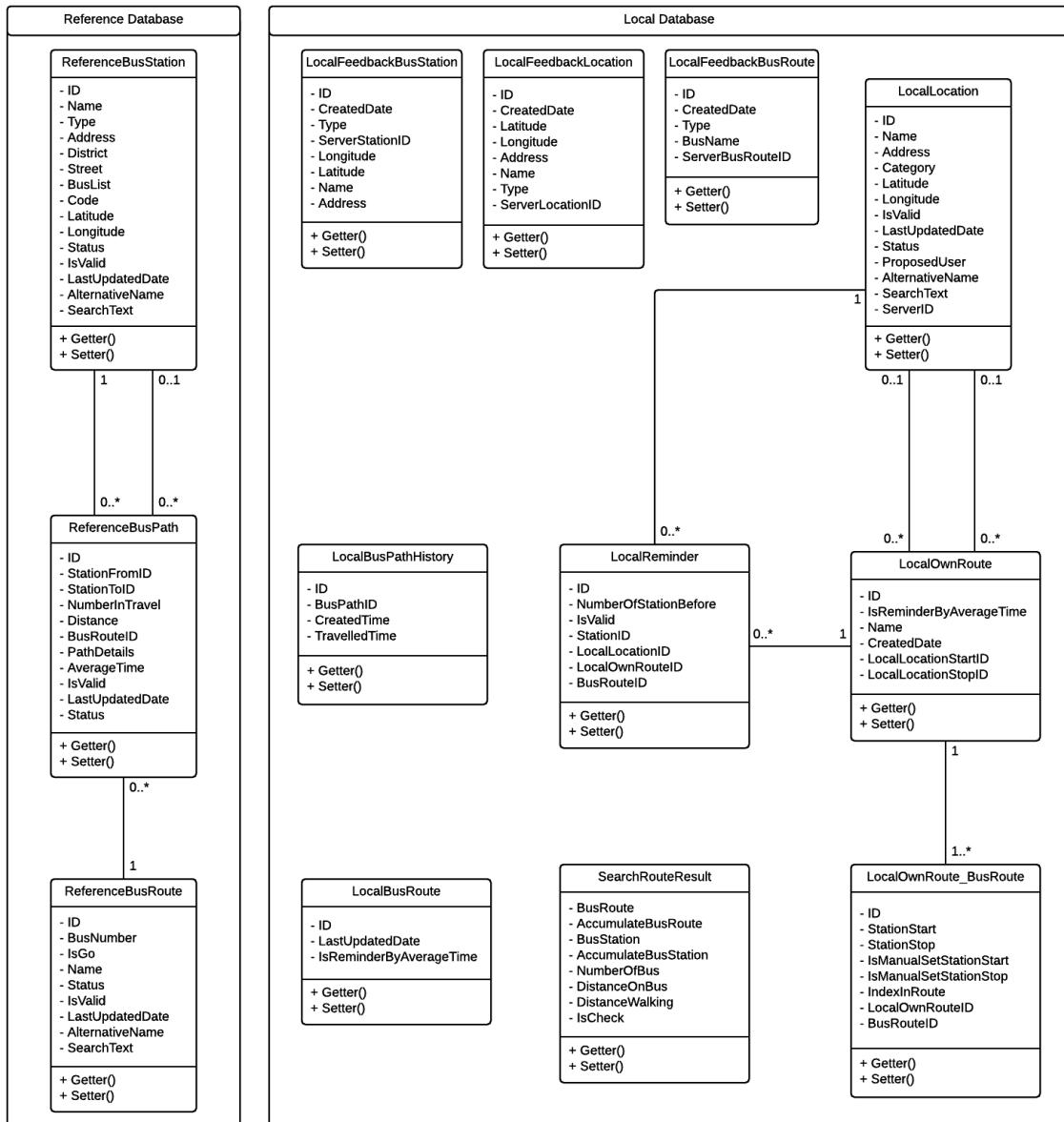


Figure 34 Mobile Application Class Diagram

4.4. Mobile Application Diagram Explanation

4.4.1. ReferenceBusStation

Attribute

Attribute	Type	Visibility	Description
ID	String	Private	Unique identifier of each bus station.
Name	String	Private	Name of station
Type	String	Private	Type of station, “Nhà chờ” or “Trạm chờ” or “Bến xe buýt”.
Address	String	Private	Address of station.

District	String	Private	District of station.
Street	String	Private	Street of station.
BusList	String	Private	List of bus which is available at station.
Code	String	Private	Code of station.
Latitude	Double	Private	Latitude of station.
Longitude	Double	Private	Longitude of station.
Status	String	Private	Status of bus station.
IsValid	Boolean	Private	Validity of bus station.
LastUpdatedDate	DateTime	Private	Last time bus station is updated.
AlternativeName	String	Private	Other name of bus station.
SearchText	String	Private	Text which is used to search for station.

Method

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

4.4.2. ReferenceBusPath

Attribute

Attribute	Type	Visibility	Description
ID	String	Private	Unique identifier of each bus path.
StationFromID	String	Private	ID of station at which bus path begins.
StationToID	String	Private	ID of station at which bus path ends.
NumberInTravel	Integer	Private	The order of bus path in bus route.
Distance	Double	Private	Distance of bus path.
BusRouteID	String	Private	ID of bus route which contains bus path.
PathDetails	String	Private	Collection of coordinates which form the path.
AverageTime	Datetime	Private	Average time a bus

			has to spend on bus path.
IsValid	Boolean	Private	Validity of bus path.
LastUpdatedDate	DateTime	Private	Last time bus path is updated.
Status	String	Private	Status of bus path.
Method			
Method	Return type	Visibility	Description
Getter	Attribute type	Public	Get attribute value
Setter	Void	Public	Set value of attribute

4.4.3. ReferenceBusRoute

Attribute

Attribute	Type	Visibility	Description
ID	String	Private	Unique identifier of each bus route.
BusNumber	String	Private	Number of bus.
IsGo	Boolean	Private	Type of bus, "Lượt đi" or "Lượt về".
Name	String	Private	Official name of bus.
Status	String	Private	Status of bus route.
IsValid	Boolean	Private	Validity of bus route.
LastUpdatedDate	DateTime	Private	Last time bus route is updated.
AlternativeName	String	Private	Other name of bus route.
SearchText	String	Private	Text which is used to search for bus.

Method

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

4.4.4. LocalFeedbackBusStation

Attribute

Attribute	Type	Visibility	Description
ID	String	Private	Unique identifier of each feedback.
CreatedDate	DateTime	Private	Created date of feedback.
Type	String	Private	Type of feedback.
ServerStationID	String	Private	ID of bus station id

			which feedback is about.
Longitude	Double	Private	Proposed new longitude of station in feedback.
Latitude	Double	Private	Proposed new latitude of station in feedback.
Name	String	Private	New name of bus station in feedback.
Address	String	Private	Proposed new address of station in feedback.

Method

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

4.4.5. LocalFeedbackLocation

Attribute

Attribute	Type	Visibility	Description
ID	String	Private	Unique identifier of each feedback.
CreatedDate	DateTime	Private	Created date of feedback.
Latitude	Double	Private	Proposed new latitude of location in feedback.
Longitude	Double	Private	Proposed new longitude of location in feedback.
Address	String	Private	Proposed new address of location in feedback.
Name	String	Private	New name of location in feedback.
Type	String	Private	Type of feedback.
ServerLocationID	Integer	Private	ID of location which feedback is about.

Method

Method	Return type	Visibility	Description

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

4.4.6. LocalFeedbackBusRoute

Attribute

Attribute	Type	Visibility	Description
ID	String	Private	Unique identifier of each feedback.
CreatedDate	DateTime	Private	Created date of feedback.
Type	String	Private	Type of feedback.
BusName	String	Private	New name of bus in feedback.
ServerBusRouteID	String	Private	ID of bus route id which feedback is about.

Method

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

4.4.7. LocalLocation

Attribute

Attribute	Type	Visibility	Description
ID	Integer	Private	Unique identifier of each location.
Name	String	Private	Name of location.
Address	String	Private	Address of location.
Category	String	Private	Category of location.
Latitude	Double	Private	Latitude of location.
Longitude	Double	Private	Longitude of location.
IsValid	Boolean	Private	Validity of bus location.
LastUpdatedDate	DateTime	Private	Last time bus location is updated.
Status	String	Private	Status of bus location.
ProposedUser	String	Private	User who proposed the location.
AlternativeName	String	Private	Other name of bus location.
SearchText	String	Private	Text which is used

			to search for location.
ServerID	Integer	Private	ID of location in server.

Method

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

4.4.8. LocalOwnRoute

Attribute

Attribute	Type	Visibility	Description
ID	Integer	Private	Unique identifier of each own route.
IsReminderByAverageTime	Boolean	Private	Check if own route is enable reminder by average time option.
Name	String	Private	Name of own route.
CreatedDate	Datetime	Private	Date when the own route is created.
LocalLocationStartID	Integer	Private	Start location of own route.
LocalLocationStopID	Integer	Private	Stop location of own route.

Method

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

4.4.9. LocalOwnRoute_BusRoute

Attribute

Attribute	Type	Visibility	Description
ID	Integer	Private	Unique identifier of each own route details.
StationStart	Integer	Private	Index of bus route' start station in own route.
StationStop	Integer	Private	Index of bus route'

			stop station in own route.
IsManualSetStationStart	Boolean	Private	Check if user had set start station manually or not.
IsManualSetStationStop	Boolean	Private	Check if user had set stop station manually or not.
IndexInRoute	Integer	Private	Index of own route details in own route.
LocalOwnRouteID	String	Private	ID of own route which contains own route details.
BusRouteID	String	Private	ID of bus route which own route contains.

Method

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

4.4.10. LocalReminder

Attribute

Attribute	Type	Visibility	Description
ID	Integer	Private	Unique identifier of each reminder.
NumberOfStationBefore	Integer	Private	Number of station which reminder is alerted before the expected station comes.
IsValid	Boolean	Private	Check if reminder is active or not.
StationID	String	Private	ID of station which station is set on.
LocalLocationID	Integer	Private	ID of location which reminder is set on.
LocalOwnRouteID	String	Private	ID of own route which contains reminder.
BusRouteID	String	Private	ID of bus route which contains reminder.

Method

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

4.4.11. LocalBusPathHistory

Attribute

Attribute	Type	Visibility	Description
ID	Integer	Private	Unique identifier of each bus path history.
BusPathID	String	Private	ID of bus path which bus path history is about.
CreatedTime	Datetime	Private	Time which bus path history is created.
TravelledTime	Datetime	Private	Taken time to travel on bus path.

Method

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

4.4.12. LocalBusRoute

Attribute

Attribute	Type	Visibility	Description
ID	String	Private	Unique identifier of each bus route.
LastUpdatedDate	DateTime	Private	Latest date bus route is updated.
IsReminderByAverageTime	Boolean	Private	Check if own route is enable reminder by average time option.

Method

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

4.4.13. SearchRouteResult

Attribute

Attribute	Type	Visibility	Description
------------------	-------------	-------------------	--------------------

BusRoute	String	Private	Latest bus route of result.
AccumulateBusRoute	String	Private	All bus route in result.
BusStation	String	Private	Lasted bus station of result.
AccumulateBusStation	String	Private	All bus station in result
NumberOfBus	Integer	Private	Number of bus in result.
DistanceOnBus	Double	Private	Distance travelled by bus.
DistanceWalking	Double	Private	Distance travelled on foot.
IsCheck	Boolean	Private	Check if result had checked or not.

Method

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

4.5. Interaction Diagram

4.5.1. <Member> Search route

Summary: This diagram show how member searches route.

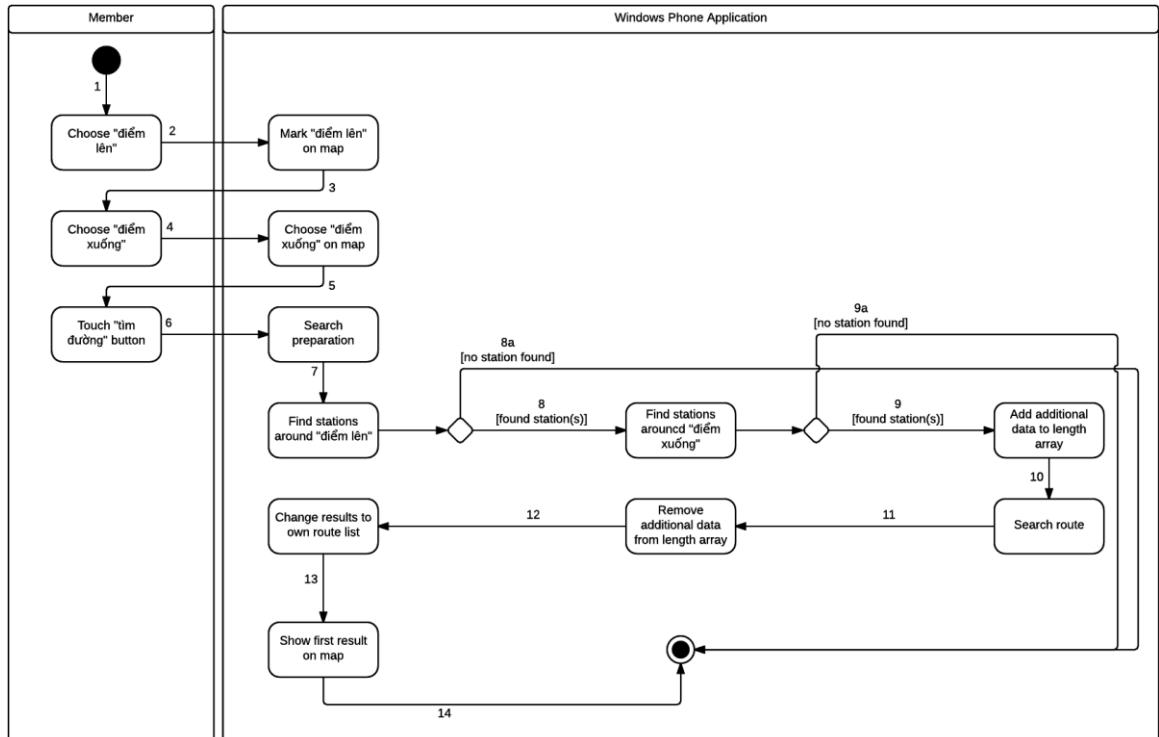


Figure 35 <Member> Search route

Step	Description
1	Start: Choose start location
2	Mark start location
3	Choose stop location
4	Mark stop location
5	Touch button
6	Search preparation
7	Find Station Around with start location
8	Find Station Around with stop location
8a	Finish
9	Add additional data to length array
9a	Finish
10	Search route
11	Remove additional data from length array
12	Change result to own route list
13	Show first result
14	Finish

4.5.2. <Member> Save searched route

Summary: This diagram show how member save searched route.

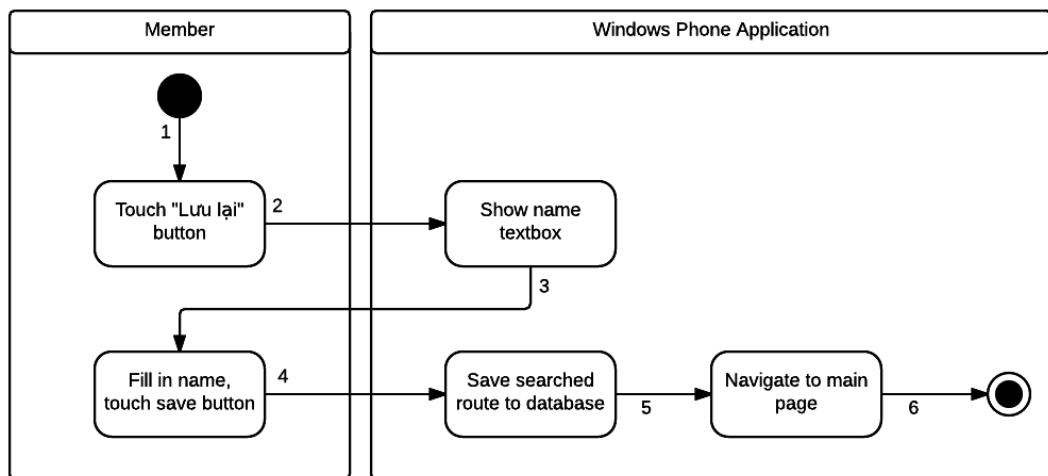


Figure 36 <Member> Save searched route

Step	Description
1	Start
2	Show textbox
3	Fill in textbox, touch button
4	Save route
5	Navigate
6	Finish

4.5.3. <Member> Create own route

Summary: This diagram shows how member creates their own route.

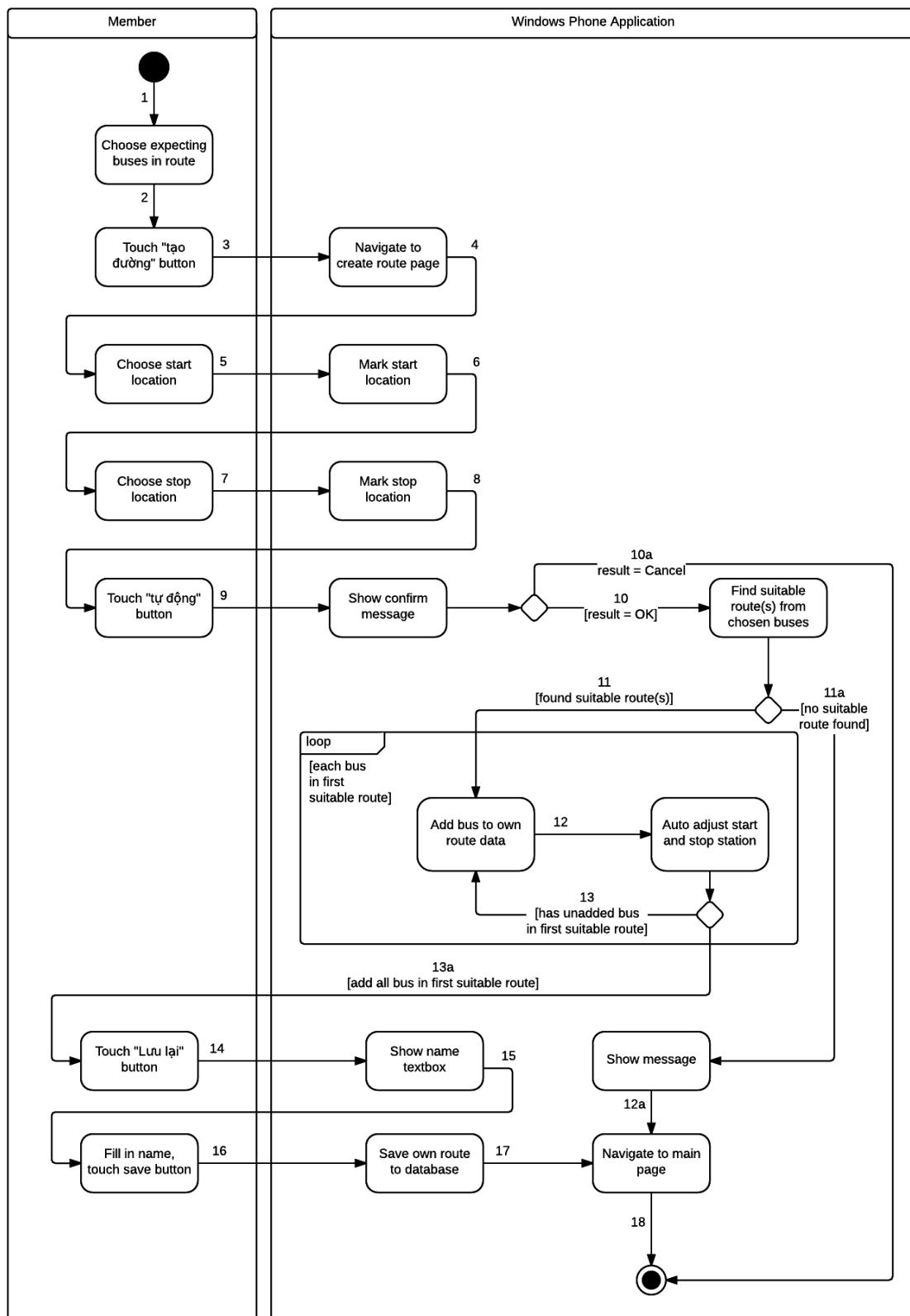


Figure 37 <Member> Create own route

Step	Description
1	Start
2	Touch button
3	Navigate
4	Choose start location
5	Mark start location
6	Choose stop location
7	Mark stop location
8	Touch button
9	Show confirm message
10	Find suitable bus routes' turn
10a	Finish
11	Add bus route to creating route
11a	Show message
12	Adjust added bus route
12a	Navigate
13	Add bus route to creating route
13a	Touch button
14	Show textbox
15	Fill in textbox
16	Save own route
17	Navigate
18	Finish

4.5.4. <Member> View route list

Summary: This diagram show how member view route list.

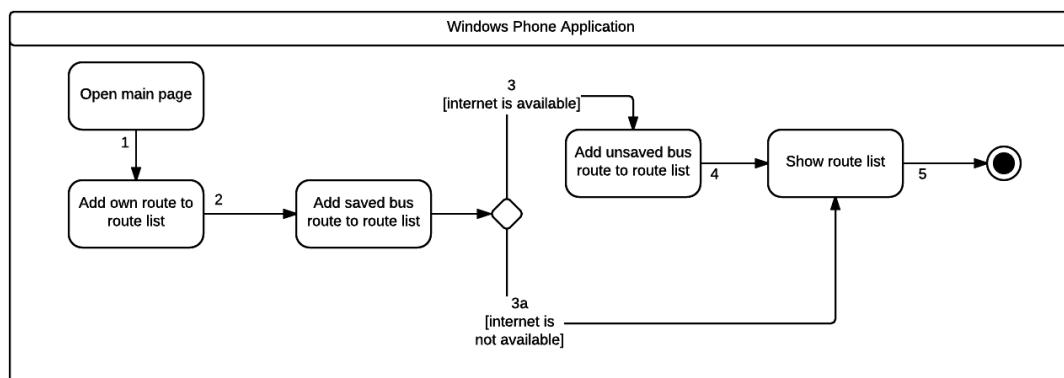


Figure 38 <Member> View route list

Step	Description
1	Start
2	Get saved bus route list
3	Get unsaved bus route list
3a	Show route list
4	Show route list
5	Finish

4.5.5. <Member> View route

Summary: This diagram shows how member view route.

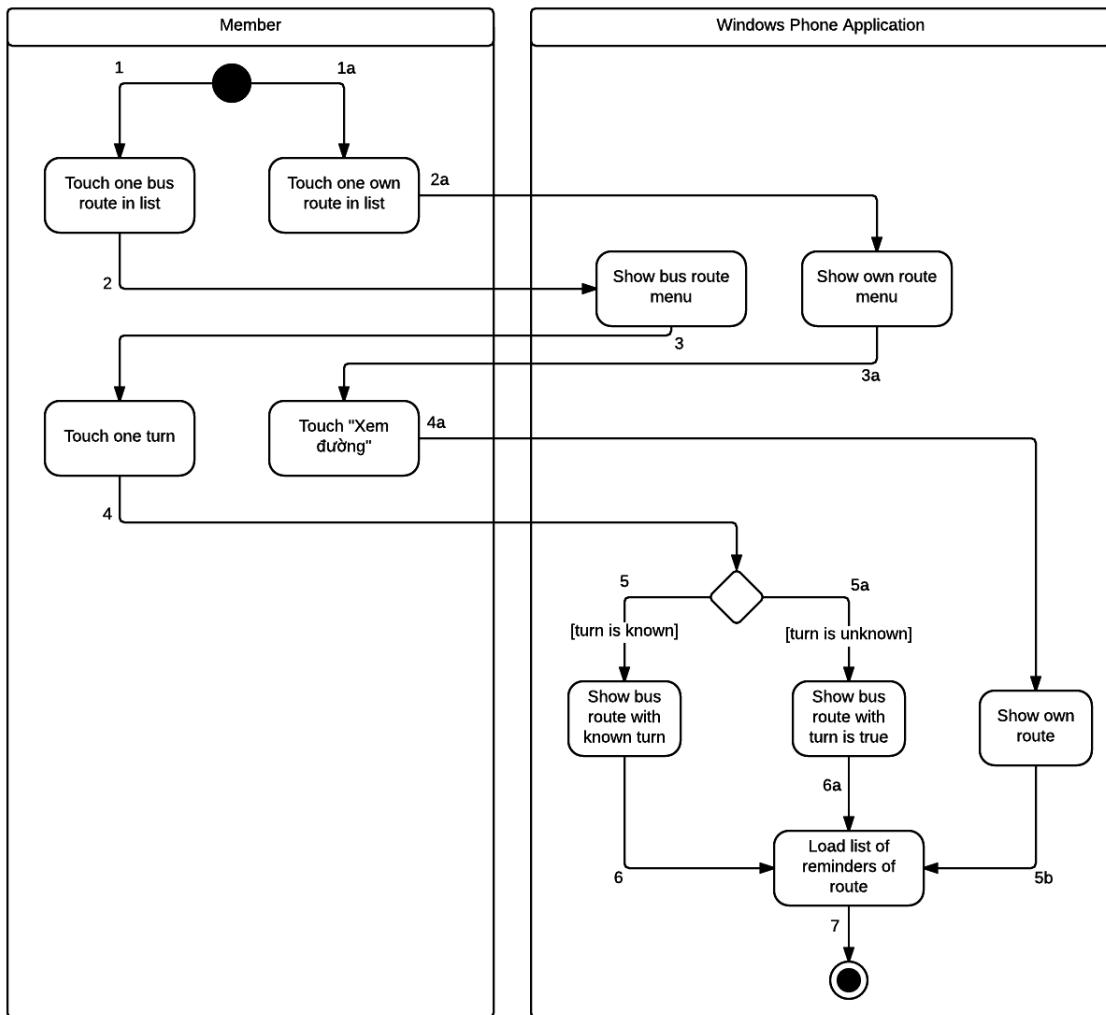


Figure 39 <Member> View route

Step	Description
1	Start
1a	Start
2	Show Bus Route Menu
2a	Show Own Route Menu
3	Touch on item
3a	Touch on item
4	Navigate to show route page
4a	Navigate to show route page
5	Load Bus Route
5a	Load Bus Route
5b	Load Bus Route
6	Load Bus Route
6a	Load Bus Route
7	Finish

4.5.6. <Member> Add reminder

Summary: This diagram shows how member add reminder.

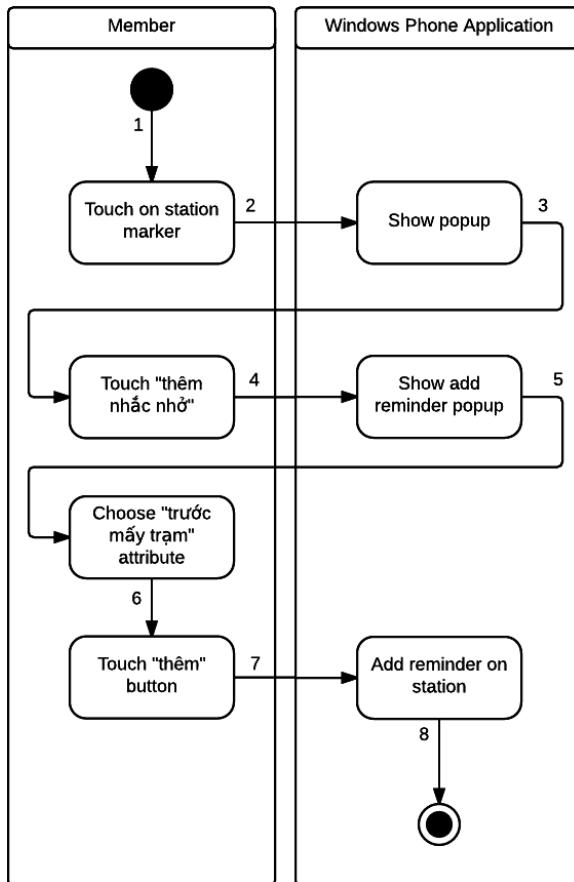


Figure 40 <Member> Add reminder to station

Step	Description
1	Start
2	Show station popup
3	Touch item
4	Show add reminder popup
5	Choose attribute
6	Touch button
7	Add reminder
8	Finish

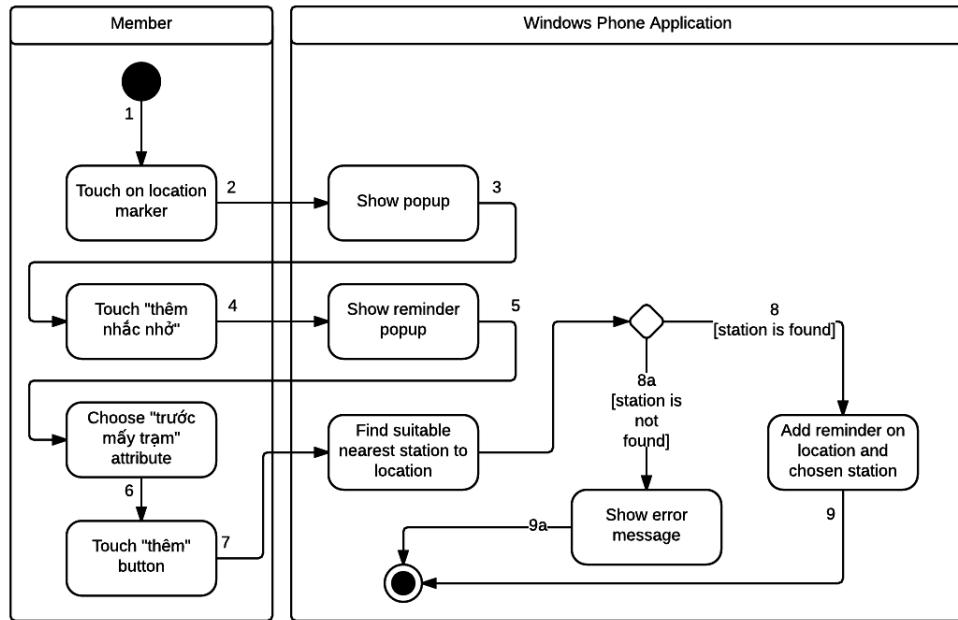


Figure 41 <Member> Add reminder to location

Step	Description
1	Start
2	Show station popup
3	Touch item
4	Show add reminder popup
5	Choose attribute
6	Touch button
7	Find nearest station to location
8	Add reminder
8a	Show message
9	Finish
9a	Finish

4.5.7. <Member> Search location

Summary: This diagram show how member search location.

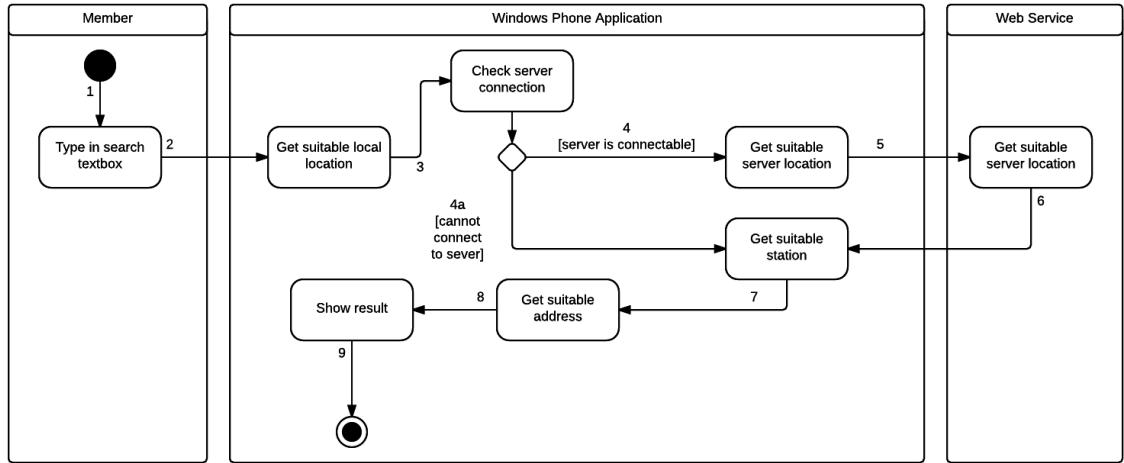


Figure 42 <Member> Search location

Step	Description
1	Start
2	Get suitable local location
3	Check server connection
4	Get suitable server location
4a	Get suitable station
5	Get data from web service
6	Get suitable station
7	Get suitable address
8	Show result
9	Finish

4.5.8. <Member> Find next station

Summary: This diagram show how member find next station.

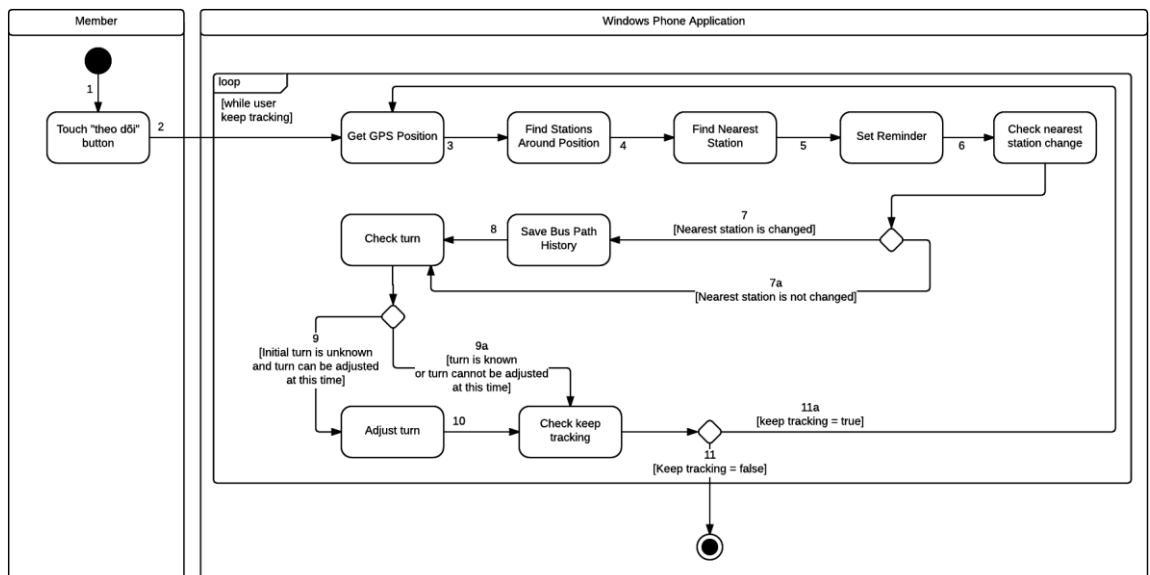


Figure 43 <Member> Find next station

Step	Description
1	Start
2	Tracking, get GPS location
3	Find Station Around
4	Find Nearest Station
5	Show Reminders
6	Check is nearest station changed?
7	Save Bus Path History
7a	Check is turn need to adjust or not.
8	Check is turn need to adjust or not.
9	Adjust turn
9a	Check is tracking or not.
10	Check is tracking or not
11	Get GPS location.
11a	Finish

4.5.9. <Member> Show reminder

Summary: This diagram show how system set reminder by location.

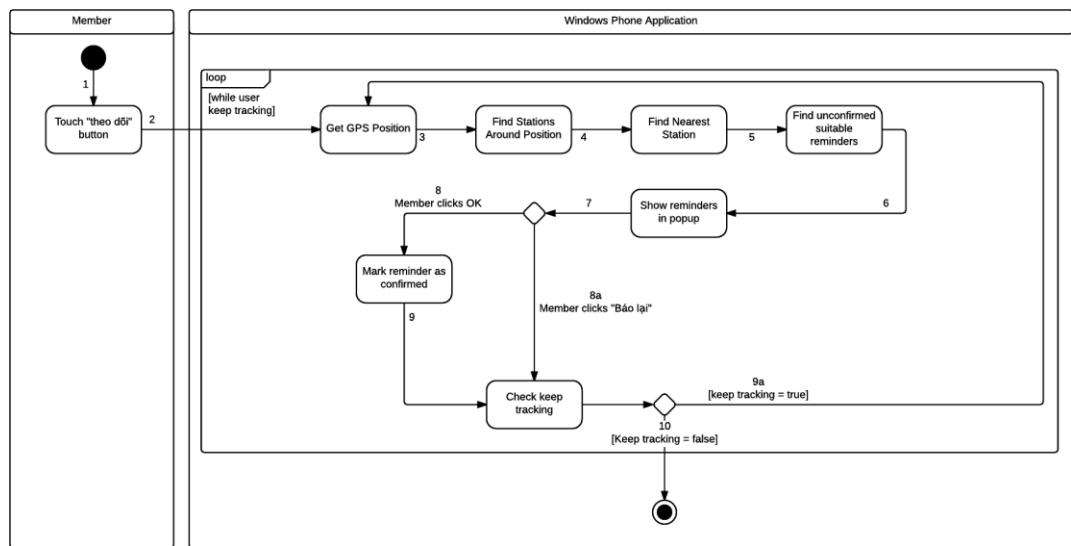


Figure 44 <Member> Set reminder by location

Step	Description
1	Start
2	Tracking, get GPS location
3	Find Station Around
4	Find Nearest Station
5	Show Reminder
6	Show popup
7	Check member response
8	Member click OK
8a	Member click "Báo lại"
9	Check keep tracking
9a	Get GPS location
10	Finish

4.5.10. <Member> Set reminder by average time

Summary: This diagram show how member set reminder by average time.

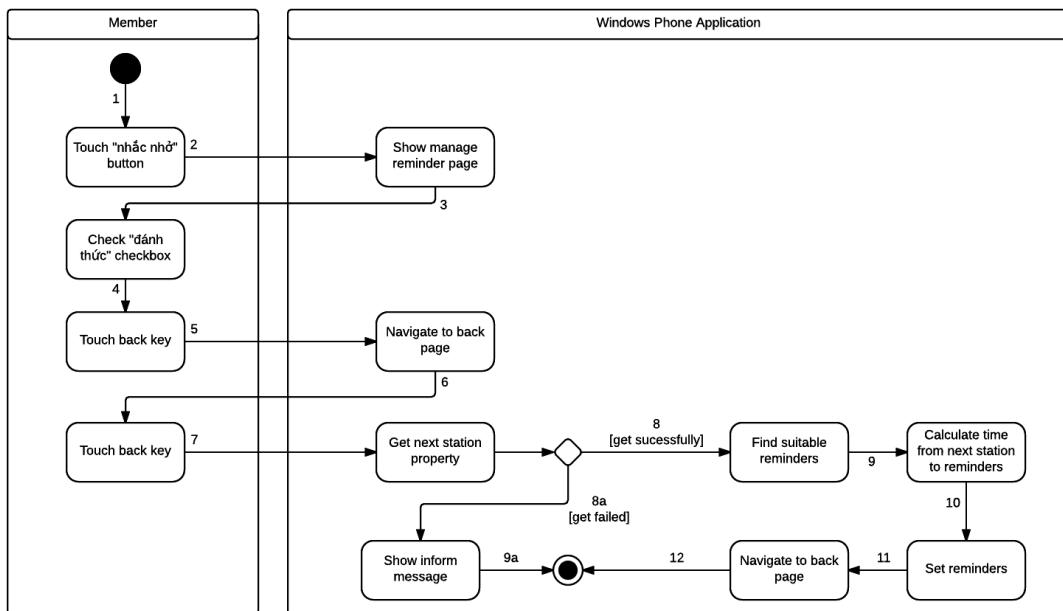


Figure 45 <Member> Set reminder by average time

Step	Description
1	Start
2	Navigate to reminder page
3	Check checkbox
4	Touch back key of device
5	Navigate back
6	Touch back key of device
7	Get next station property
8	Show reminders
8a	Show message, finish
9	Calculate time
9a	Finish
10	Set reminders
11	Navigate
12	Finish

4.5.11. <Member> Search walking route

Summary: This diagram show how member search walking route.

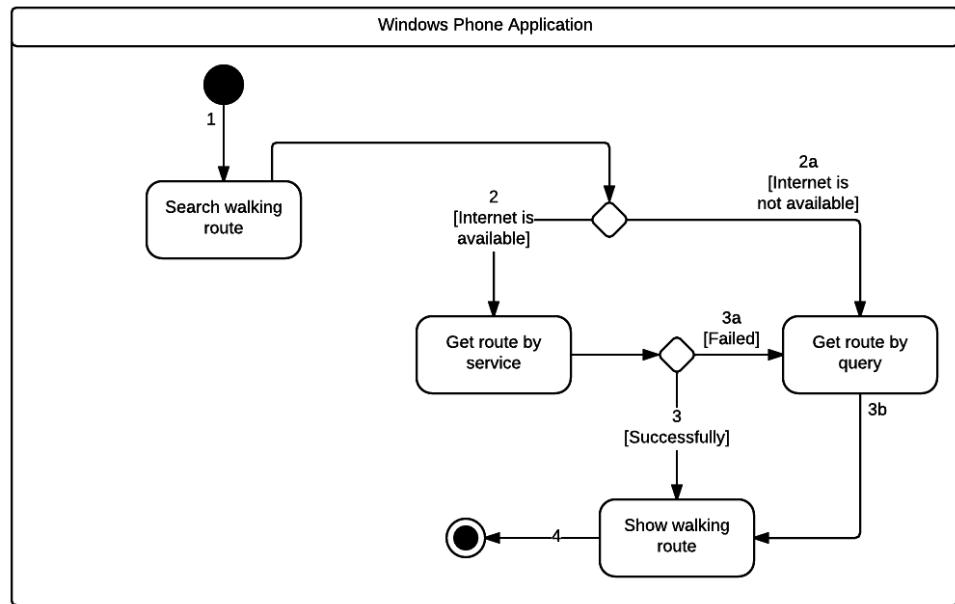


Figure 46 <Member> Search walking route

Step	Description
1	Start
2	Get route by service
2a	Get route by query
3	Show route
3a	Get route by query
3b	Show route
4	Finish

4.5.12. <Member> Save bus path history

Summary: This diagram shows how member save bus path history.

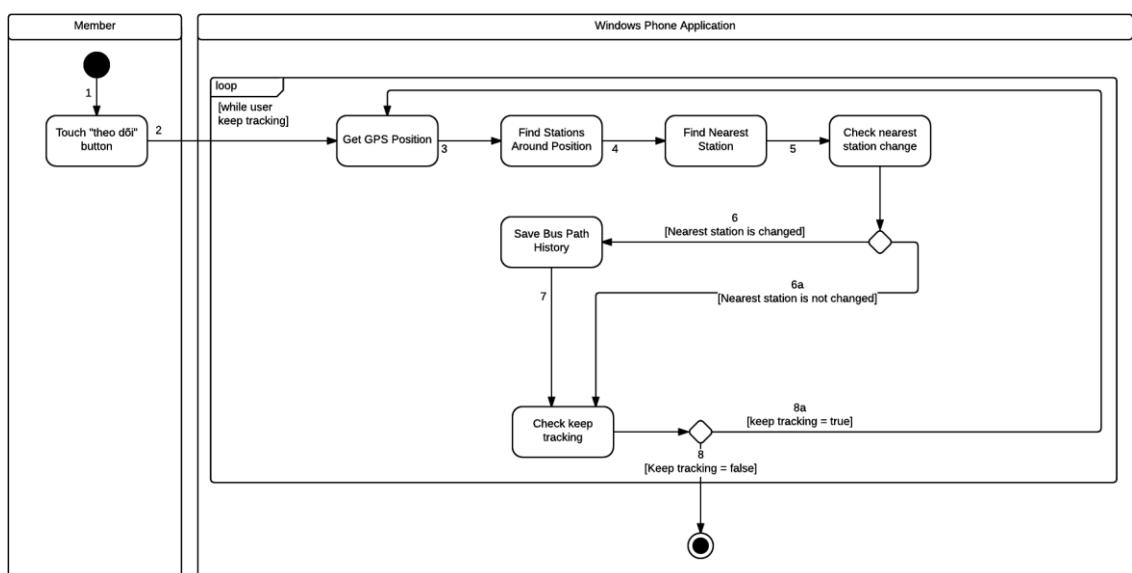


Figure 47 <Member> Save bus path history

Step	Description
1	Start
2	Tracking, get GPS location
3	Find Station Around
4	Find Nearest Station
5	Check nearest station changed
6	Save bus path history
6a	Check keep tracking
7	Check keep tracking
8	Finish
8a	Get GPS location

4.5.13. <Member> Upload bus path history

Summary: This diagram shows how member upload bus path history.

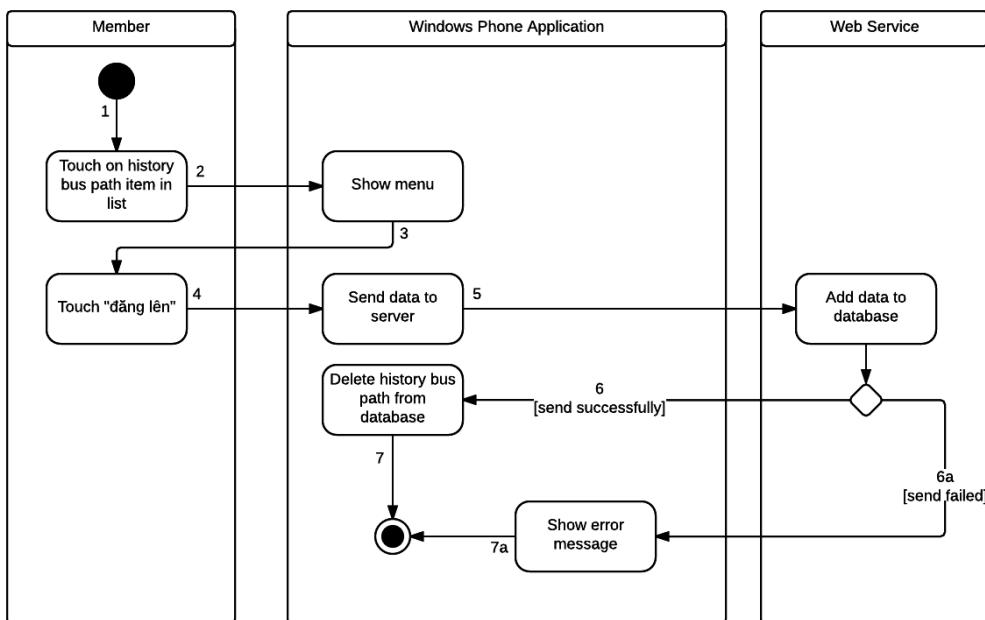


Figure 48 <Member> Upload bus path history

Step	Description
1	Start
2	Show menu
3	Touch on item
4	Send History
5	Send data to web service
6	Delete History, show message
6a	Show message
7	Finish
7a	Finish

4.5.14. <Member> Edit own route

Summary: This diagram show how member edit own route.

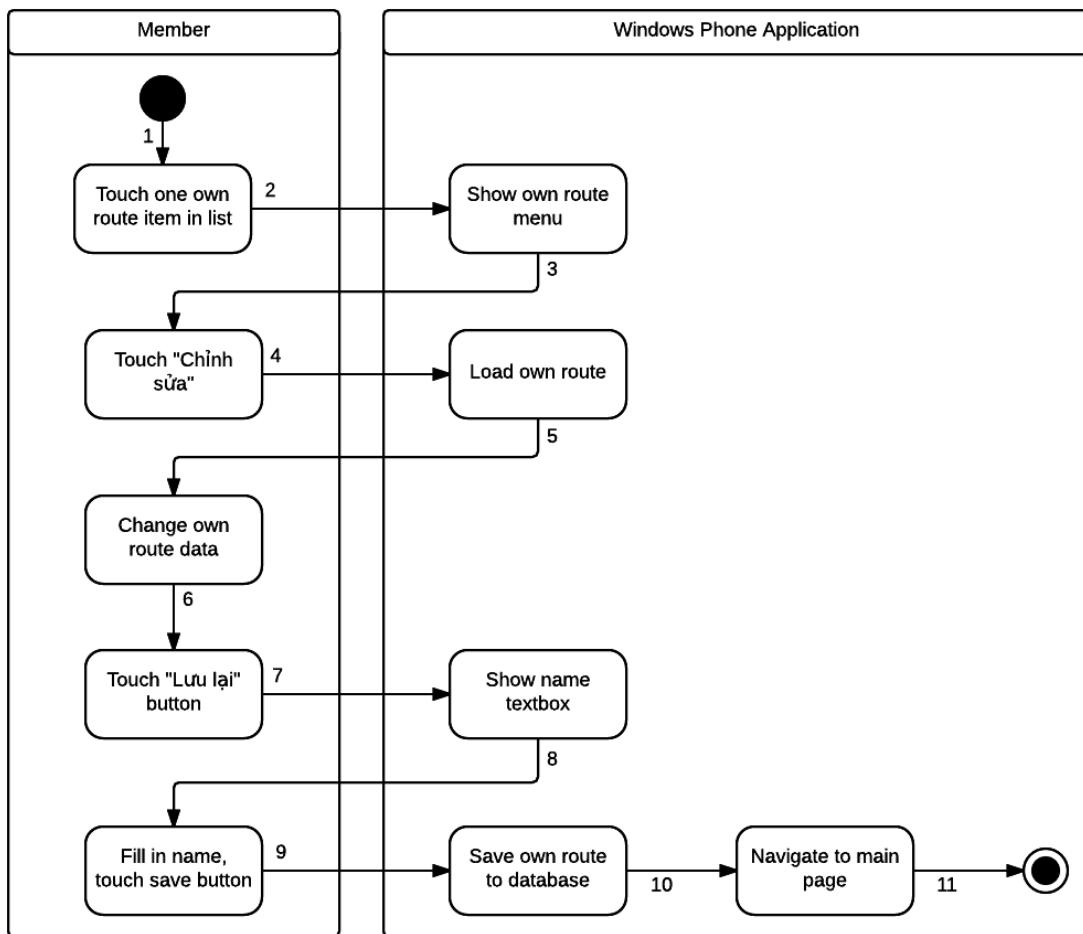


Figure 49 <Member> Edit own route

Step	Description
1	Start
2	Show Own Route Menu
3	Touch on item
4	Navigate, load own route
5	Change own route data
6	Touch button
7	Show textbox
8	Fill in name
9	Save own route
10	Navigate
11	Finish

4.5.15. <Member> Delete own route

Summary: This diagram show how member delete own route.

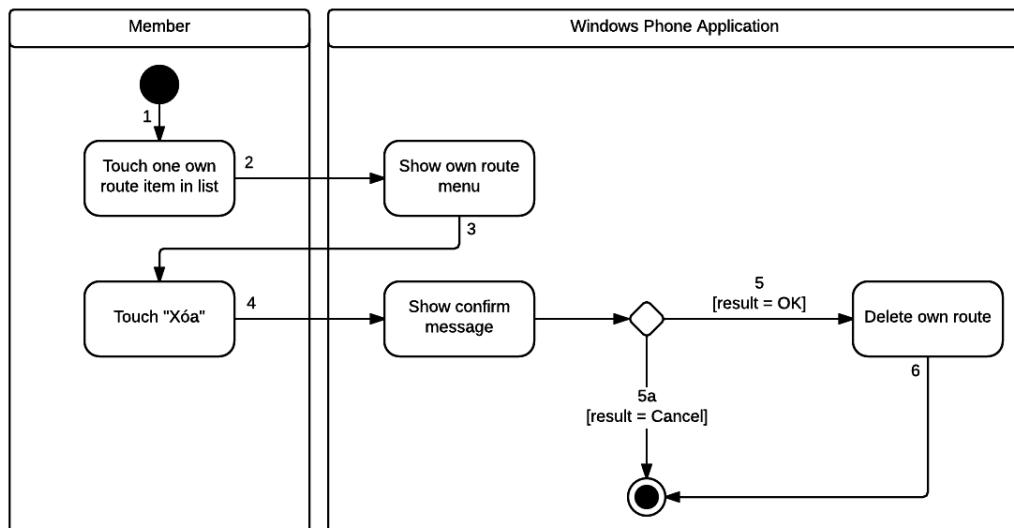


Figure 50 <Member> Delete own route

Step	Description
1	Start
2	Show Own Route Menu
3	Touch on item
4	Show confirm message
5	Delete Own Route
5a	Finish
6	Finish

4.5.16. <Member> Edit reminder

Summary: This diagram shows how member edit reminder.

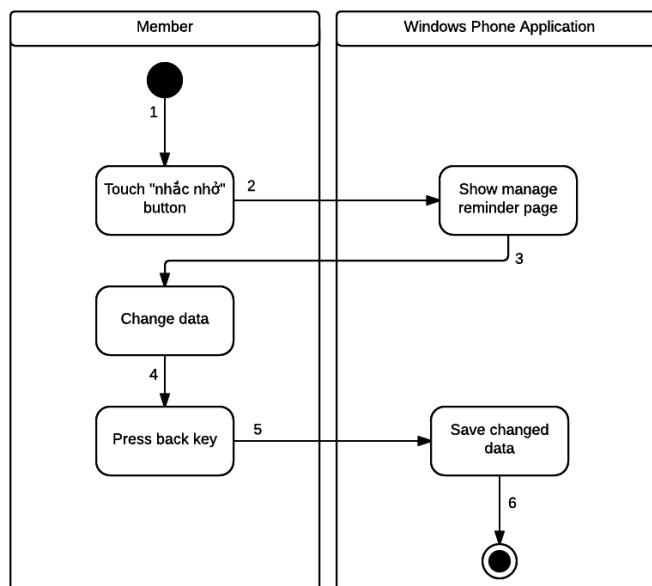


Figure 51 <Member> Edit reminder

Step	Description
1	Start
2	Navigate
3	Change data
4	Press back key on device
5	Save changed data
6	Finish

4.5.17. <Member> Remove reminder

Summary: This diagram show how member remove reminder.

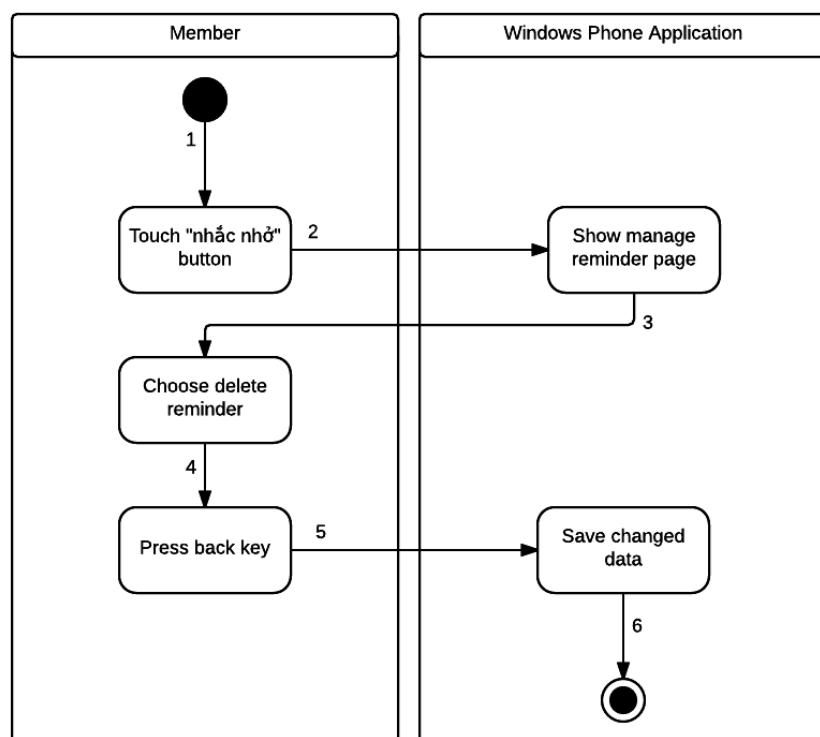


Figure 52 <Member> Remove reminder

Step	Description
1	Start
2	Navigate
3	Touch delete
4	Press back key on device
5	Save changed data
6	Finish

4.5.18. <Member> Feedback station

Summary: This diagram show how member feedback station.

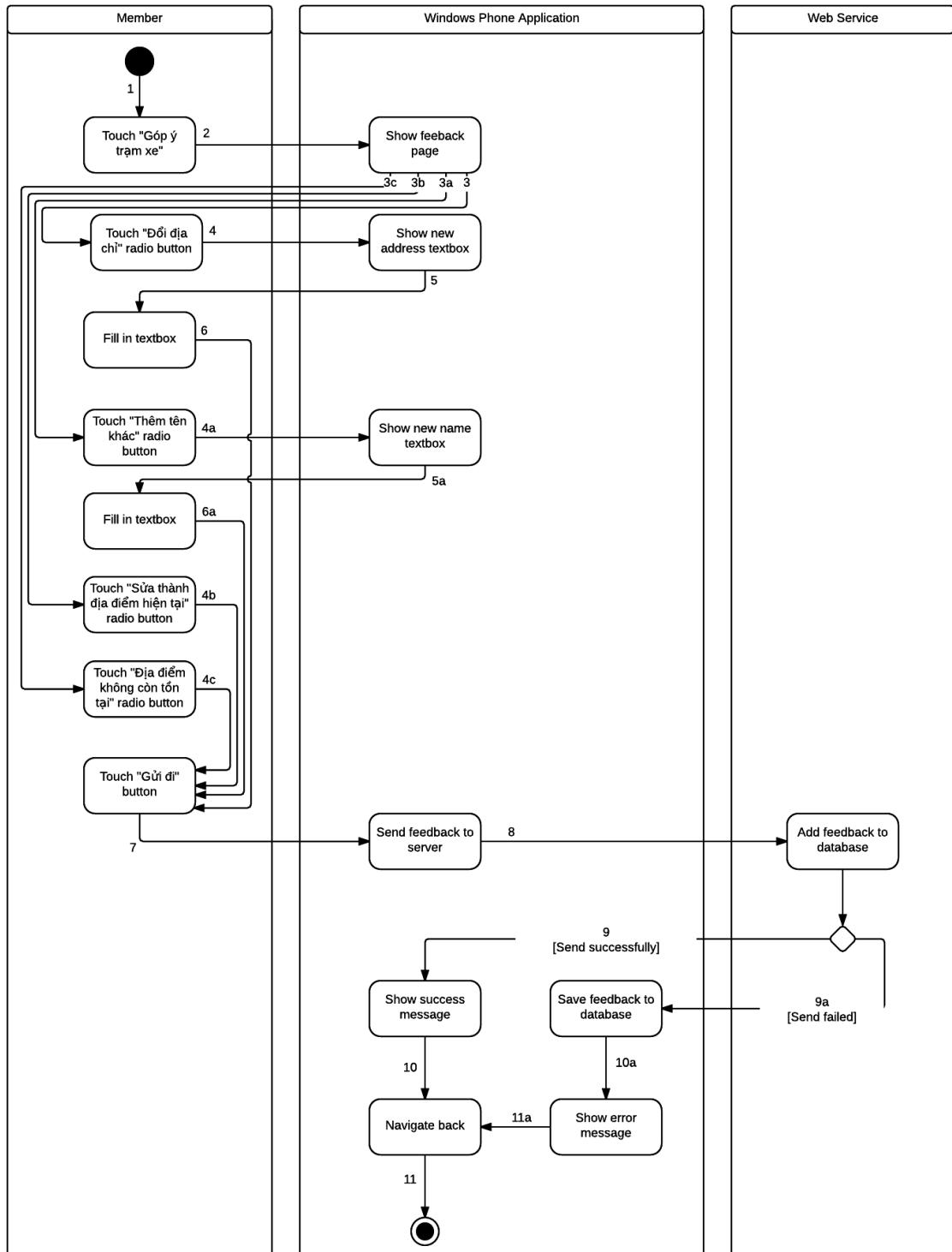


Figure 53 <Member> Feedback station

Step	Description
1	Start
2	Navigate to feedback page
3	Touch radio button
3a	Touch radio button
3b	Touch radio button
3c	Touch radio button

4	Show textbox
4a	Show textbox
4b	Show textbox
4c	Show textbox
5	Fill in textbox
5a	Fill in textbox
6	Touch button
6a	Touch button
7	Send Feedback
8	Send data to web service
9	Show message
9a	Save feedback to database
10	Navigate
10a	Show message
11	Finish
11a	Navigate

4.5.19. <Member> Feedback bus route

Summary: This diagram show how member feedback bus route.

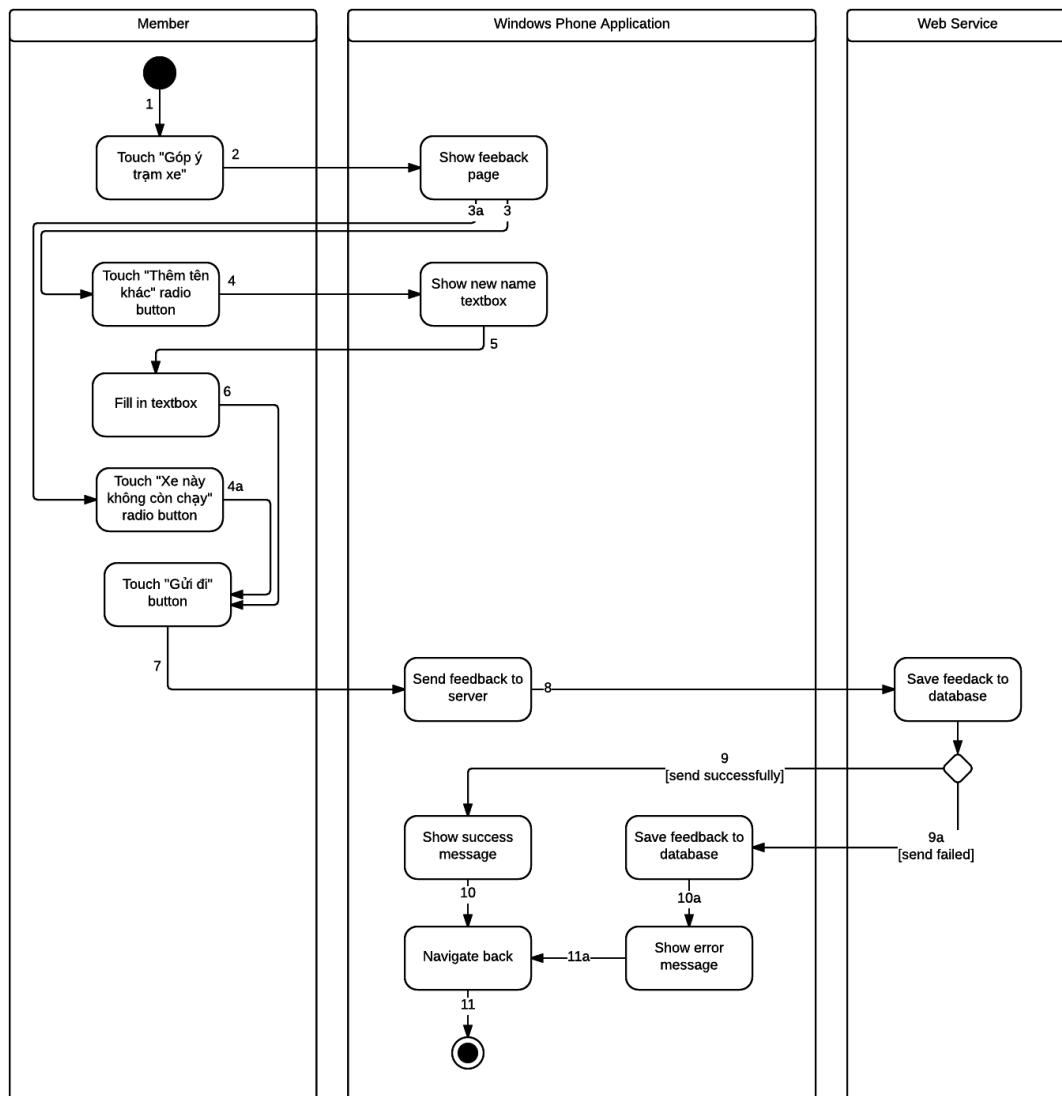


Figure 54 <Member> Feedback bus route

Step	Description
1	Start
2	Navigate to feedback page
3	Touch radio button
3a	Touch radio button
4	Show textbox
4a	Touch button
5	Fill in textbox
6	Touch button
7	Send Feedback
8	Send data to server
9	Show message
9a	Save feedback to database
10	Navigate back
10a	Show message
11	Finish

11a

Navigate back

4.5.20. <Member> Feedback location

Summary: This diagram shows how member feedback location.

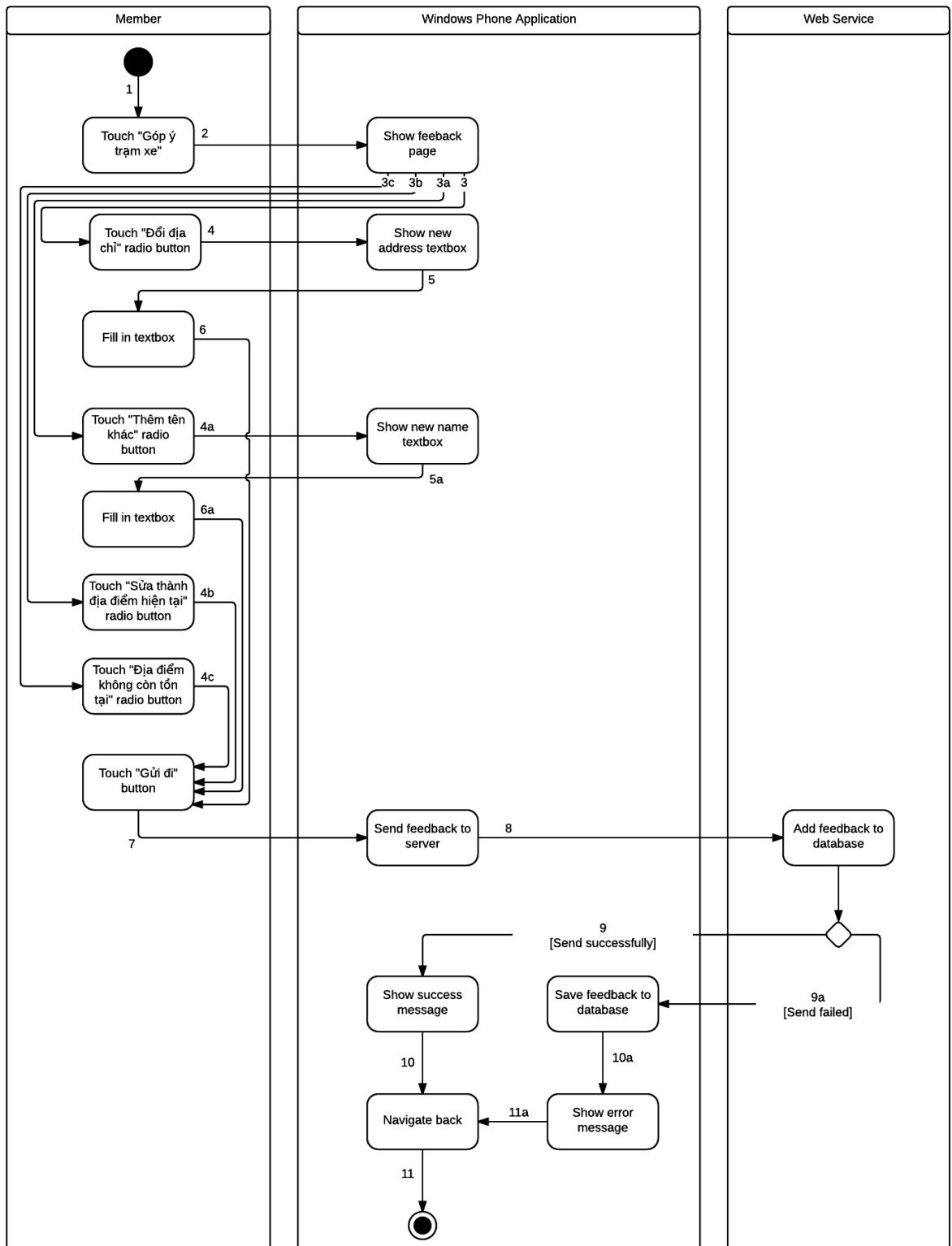


Figure 55 <Member> Feedback location

Step	Description
1	Start
2	Navigate to feedback page
3	Touch radio button
3a	Touch radio button
3b	Touch radio button
3c	Touch radio button
4	Show textbox
4a	Show textbox
4b	Show textbox
4c	Show textbox
5	Fill in textbox
5a	Fill in textbox
6	Touch button
6a	Touch button
7	Send Feedback
8	Send data to web service
9	Show message
9a	Save feedback to database
10	Navigate
10a	Show message
11	Finish
11a	Navigate

4.5.21. <Staff> Edit bus route

Summary: This diagram show how staff edit bus station

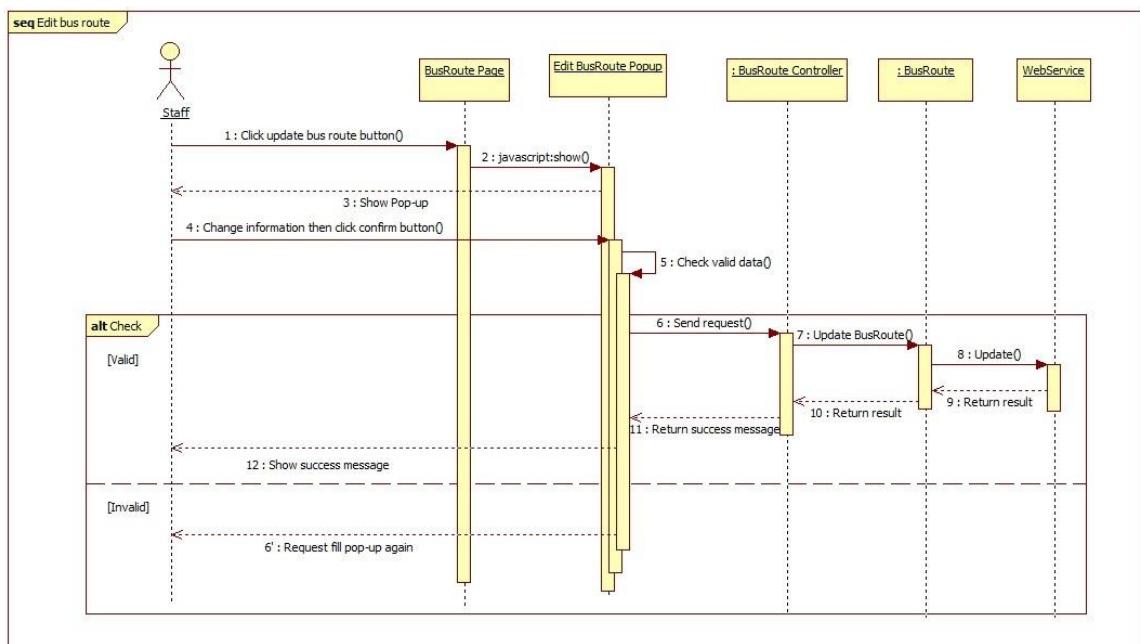


Figure 56 <Staff> Edit bus route

4.5.22. <Staff> Edit bus station

Summary: This diagram show how staff update bus station

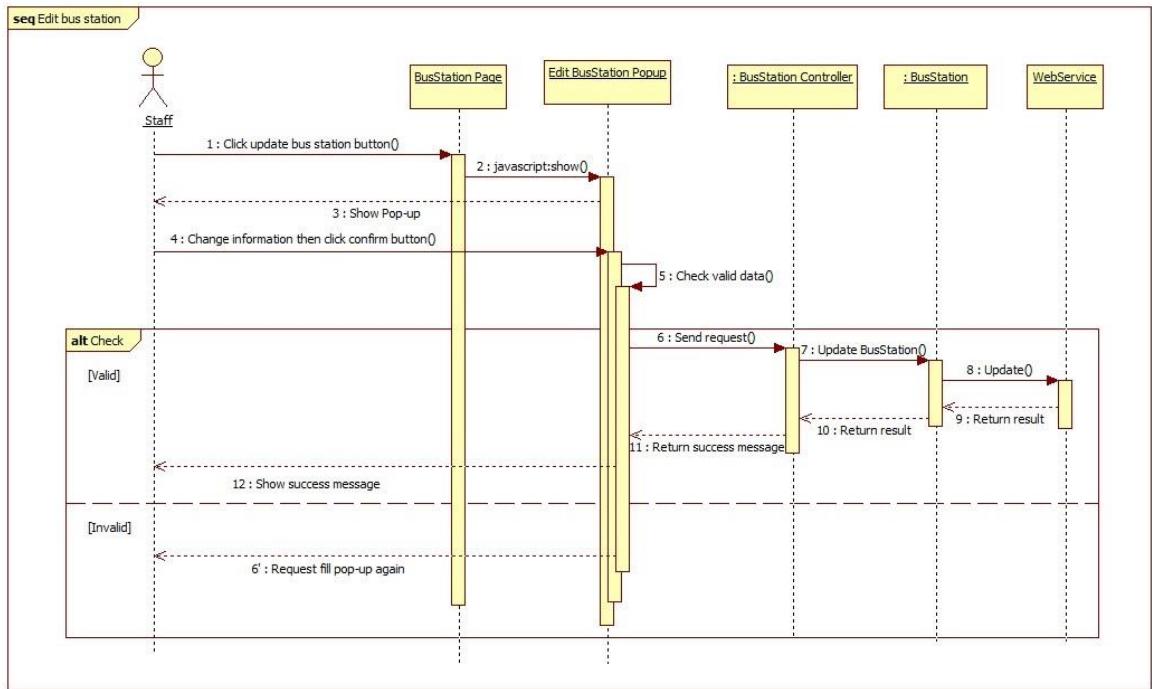


Figure 57 <Staff> Edit bus station

4.5.23. <Staff> Edit bus path

Summary: This diagram show how staff update bus path

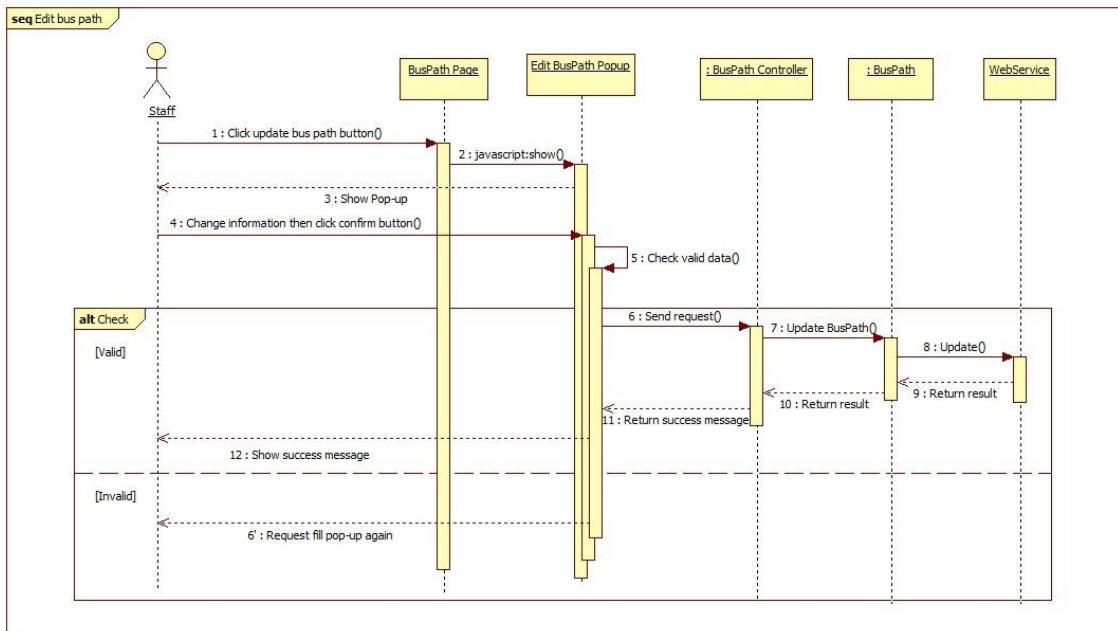


Figure 58 <Staff> Edit bus path

4.5.24. <Staff> Add location

Summary: This diagram show how staff add location

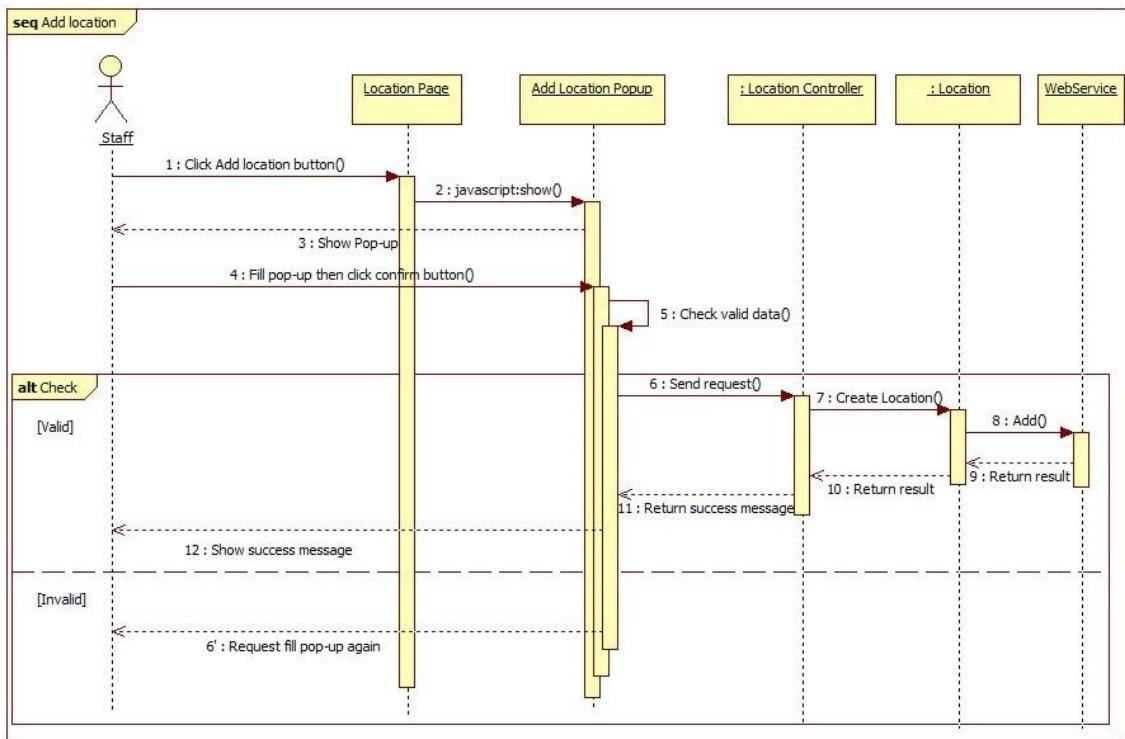


Figure 59 <Staff> Add location

5. User Interface Design

5.1. Mobile Interface Design

5.1.1. Main page



Figure 60 Main page

Fields

No	Field Name	Description	Read only	Mandatory	Control Type	Data Type	Length
1	SearchText	Fill in to search bus route.	No	Yes	Text box	String	N/A
7	RouteList	List of route.	No	Yes	List	ListItem	N/A

Buttons/Hyperlinks

No	Function	Description	Validation	Outcome
2	Clear Search Text	Clear SearchText textbox.	N/A	Navigate to home page.
3	Multiselect	Enable / Disable multiselect mode.	N/A	Show register page.
4	Search Route	Open search route page.	N/A	Navigate to search route page
5	Open mutiroute	Open chosen routes in view route page.	N/A	Navigate to view route page.
6	Create route	Open create route page	N/A	Navigate to create route page.

5.1.1.1 Choose bus route menu



Figure 61 Choose bus route menu

Buttons/Hyperlinks

No	Function	Description	Validation	Outcome
1	View Route	View route with chosen bus number and turn "Lượt đi".	N/A	Navigate to view route page.
2	View Route	View route with chosen bus number and turn "Lượt về".	N/A	Navigate to view route page.

3	View Route	View route with chosen bus number and turn “Unknown”	N/A	Navigate to view route page.
---	------------	--	-----	------------------------------

5.1.1.2 Choose own route menu



Figure 62 Choose own route menu

Buttons/Hyperlinks

No	Function	Description	Validation	Outcome
1	View Route	View chosen own route.	N/A	Navigate to view route page.
2	Edit Route	Load chosen own route on create route page to edit.	N/A	Navigate to create route page.
3	Delete route	Delete chosen own route.	N/A	Delete own route from database.

5.1.1.3 Settings page



Figure 63 Settings page

Fields

No	Field Name	Description	Read only	Mandatory	Control Type	Data Type	Length
1	SaveHistory	Decide whether system save history while tracking or not.	No	Yes	Check box	Boolean	N/A
2	NotifyAllStation	Decide whether system notify all station when app is running background or reminded station only.	No	Yes	Radio button	Boolean	N/A
3	NotifyReminderStation	Decide whether system notify only reminded station when app is running background or all.	No	Yes	Radio button	Boolean	N/A

Buttons/Hyperlinks

No	Function	Description	Validation	Outcome
4	UpdateData	Update data in system.	N/A	Data is updated.
5	Update Information	Update member's account information.	N/A	Member's account information is updated.
6	Log out	Log out.	N/A	Navigate to login page.

5.1.2. View route page

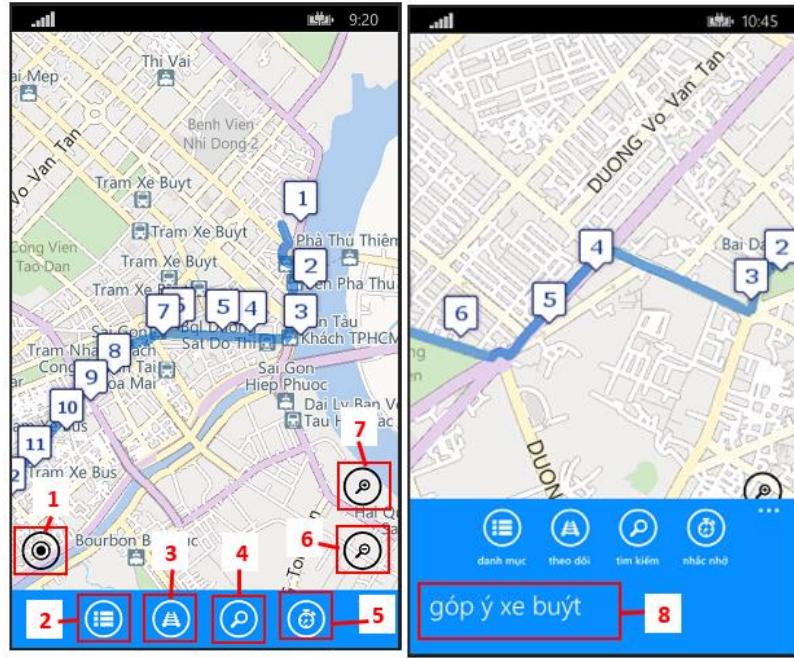


Figure 64 View route page

Buttons/Hyperlinks

No	Function	Description	Validation	Outcome
1	Get Current Location	Get current location of device.	N/A	Map is focus at current location.
2	Menu	Open menu.	N/A	Menu is opened.
3	Tracking	Start tracking / Stop tracking.	N/A	Start/Stop tracking.
4	Search	Show search textbox.	N/A	Search textbox is shown.
5	Manage Reminder	Open edit reminder page.	N/A	Navigate to reminder page.
6	Zoom out	Zoom map out.	N/A	Map is zoomed out.
7	Zoom in	Zoom map in	N/A	Map is zoomed in.
8	Feedback	Open feedback page for bus route.	N/A	Navigate to feedback page.

5.1.2.1 Add reminder popup



Figure 65 Add reminder popup

Fields

No	Field Name	Description	Read only	Mandatory	Control Type	Data Type	Length
1	Number Of Station Before	Set number of stations before when reminded location comes that reminder will be alerted.	No	Yes	Drop down list	String	N/A
2	Style	Style of reminder.	No	Yes	Drop down list	String	N/A

Buttons/Hyperlinks

No	Function	Description	Validation	Outcome
3	Add Reminder	Add reminder to current route.	N/A	Close popup.
4	Go back	Close popup.	N/A	Close popup.

5.1.2.2 Edit reminder page



Figure 66 Edit reminder page

Fields

No	Field Name	Description	Read only	Mandatory	Control Type	Data Type	Length
1	Wake up	Enable wake up function.	No	Yes	Check box	Boolean	N/A
3	1 station	Set “number of station before” attribute of reminder to 1.	No	Yes	Radio button	Boolean	N/A
4	2 station	Set “number of station before” attribute of reminder to 2.	No	Yes	Radio button	Boolean	N/A
5	3 station	Set “number of station before” attribute of reminder to 3.	No	Yes	Radio button	Boolean	N/A

Buttons/Hyperlinks

No	Function	Description	Validation	Outcome
2	Enable	Enable/Disable reminder.	N/A	Reminder is enable

	Reminder			/ disable.
6	Close details	Close details of reminder.	N/A	Close details of reminder.
7	Open details	Open details of reminder.	N/A	Open details of reminder.

5.1.3. Create route page



Figure 67 Create route page

No	Function	Description	Validation	Outcome
1	Get Current Location	Get current location of device.	N/A	Map is focus at current location.
2	Menu	Open menu.	N/A	Menu is opened.
3	Auto Create Route	Auto create route from current data.	N/A	Route is created automatically.
4	Search	Show search textbox.	N/A	Search textbox is shown.
5	Save route	Open save route popup.	N/A	Open save route popup.
6	Zoom out	Zoom map out.	N/A	Map is zoomed out.
7	Zoom in	Zoom map in	N/A	Map is zoomed in.

5.1.3.1 Location menu

- Location menu of create route page appears when tapping on a location marker of map in create route page.



Figure 68 Location menu

Buttons/Hyperlinks

No	Function	Description	Validation	Outcome
1	Mark start location	Mark location as start location of creating route.	N/A	Location is marked as start location.
2	Mark stop location	Mark location as stop location of creating route.	N/A	Location is marked as stop location.

5.1.3.2 Add route menu

- Add route menu appears when tapping on an unadded bus route in route list.



Figure 69 Add route menu

Buttons/Hyperlinks

No	Function	Description	Validation	Outcome
1	Add Bus Route	Add chosen bus route to creating route.	N/A	Bus route is added.

5.1.3.3 Edit route menu

- Edit route menu appears when tapping on an added bus route in route list.

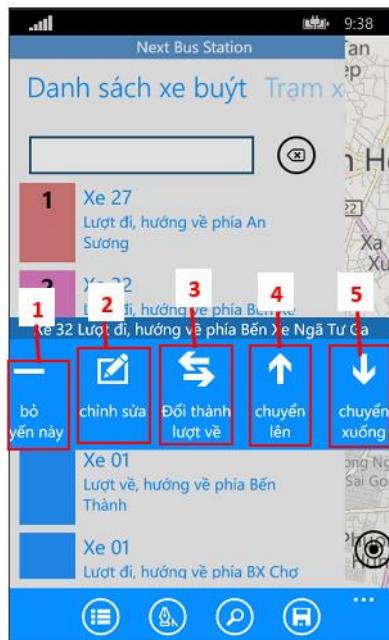


Figure 70 Edit route menu

Buttons/Hyperlinks

No	Function	Description	Validation	Outcome
1	Remove Bus Route	Remove chosen bus route from creating route.	N/A	Bus route is removed.
2	Edit Bus Route	Edit chosen bus route (adjust route manually).	N/A	Bus route is enable to edit.
3	Change Bus Route Turn	Change bus route's turn.	N/A	Bus route's turn is changed.
4	Move Bus Route Up	Move bus route up (to change index of bus route in creating route).	N/A	Bus route is moved up.
5	Move Bus Route Down	Move bus route down (to change index of bus route in creating route).	N/A	Bus route is moved down.

5.1.3.4 Save route popup



Figure 71 Save route popup

Fields

No	Field Name	Description	Read only	Mandatory	Control Type	Data Type	Length
1	Route Name	Name of route.	No	Yes	Text box	String	N/A
3	Start Location Name	Name of start location.	No	Yes	Text box	String	N/A
5	Stop Location Name	Name of stop location.	No	Yes	Text box	String	N/A

Buttons/Hyperlinks

No	Function	Description	Validation	Outcome
2	Save Route	Save creating route to database.	N/A	Navigate to main page.
4	Clear Start Location Name	Clear start location name.	N/A	Start location name is cleared.
6	Clear Stop Location Name	Clear stop location name.	N/A	Stop location name is cleared.

5.1.4. Search route page



Figure 72 Search route page

Fields

No	Field Name	Description	Read only	Mandatory	Control Type	Data Type	Length
1	SearchText	Fill in to search location	No	Yes	Text box	String	N/A

Buttons/Hyperlinks

No	Function	Description	Validation	Outcome
2	ClearText	Clear search text	N/A	Search text is cleared.
3	Get Current Location	Get current location of device.	N/A	Map is focus at current location.
4	Menu	Open menu.	N/A	Menu is opened.
5	Start search route	Search route.	N/A	Search route.
6	Save route	Open save route popup.	N/A	Open save route popup.
7	Zoom out	Zoom map out.	N/A	Map is zoomed out.
8	Zoom in	Zoom map in	N/A	Map is zoomed in.

5.1.4.1 Location menu

- Location menu of search route page appears when tapping on a location marker of map in search route page.



Figure 73 Location menu

Buttons/Hyperlinks

No	Function	Description	Validation	Outcome
1	Mark start location	Mark location as start location of creating route.	N/A	Location is marked as start location.
2	Mark stop location	Mark location as stop location of creating route.	N/A	Location is marked as stop location.

5.1.4.2 Search settings

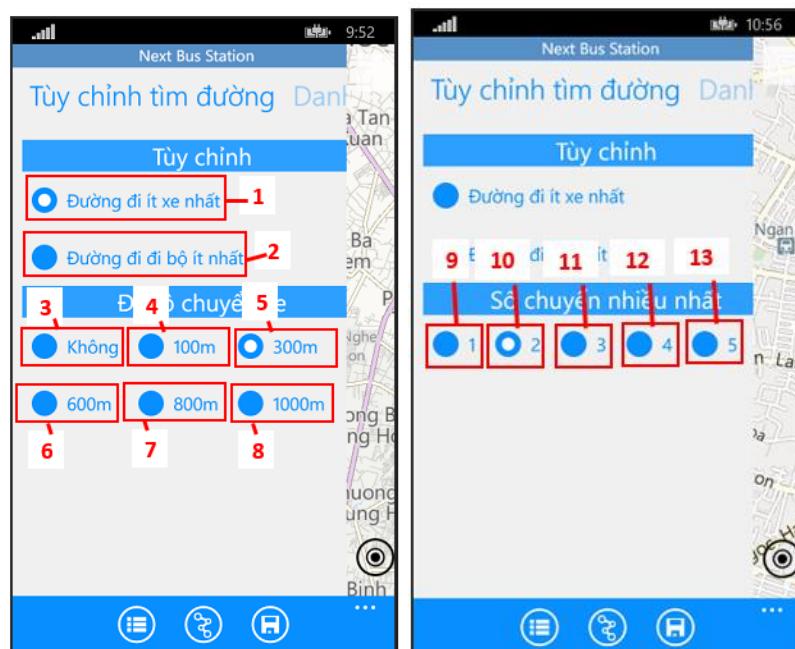


Figure 74 Search settings

Fields

No	Field Name	Description	Read only	Mandatory	Control Type	Data Type	Length
1	Search Least Bus	Enable search in least bus mode	No	Yes	Radio button	Boolean	N/A
2	Search Least Walking	Enable search in least walking mode	No	Yes	Radio button	Boolean	N/A
3	MaxWalking0	Maximum walking distance is 0 meter.	No	Yes	Radio button	Boolean	N/A
4	MaxWalking100	Maximum walking distance is 100 meters.	No	Yes	Radio button	Boolean	N/A
5	MaxWalking300	Maximum walking distance is 300 meters.	No	Yes	Radio button	Boolean	N/A
6	MaxWalking600	Maximum walking distance is 600 meters.	No	Yes	Radio button	Boolean	N/A
7	MaxWalking800	Maximum walking distance is 800 meters.	No	Yes	Radio button	Boolean	N/A
8	MaxWalking1000	Maximum walking distance is 1000 meters.	No	Yes	Radio button	Boolean	N/A
9	MaxBus1	Maximum number of bus is 1.	No	Yes	Radio button	Boolean	N/A
10	MaxBus2	Maximum number of bus is 2.	No	Yes	Radio button	Boolean	N/A
11	MaxBus3	Maximum number of bus is 3.	No	Yes	Radio button	Boolean	N/A
12	MaxBus4	Maximum number of bus is 4.	No	Yes	Radio button	Boolean	N/A

13	MaxBus5	Maximum number of bus is 5.	No	Yes	Radio button	Boolean	N/A
----	---------	-----------------------------	----	-----	--------------	---------	-----

5.1.4.3 Save route

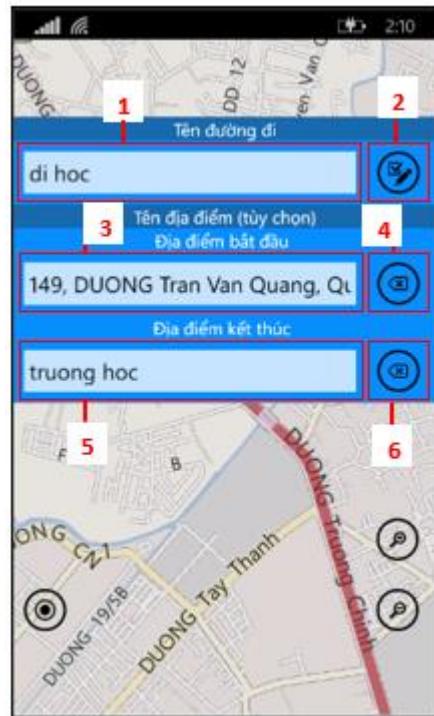


Figure 75 Save route

Fields

No	Field Name	Description	Read only	Mandatory	Control Type	Data Type	Length
1	Route Name	Name of route.	No	Yes	Text box	String	N/A
3	Start Location Name	Name of start location.	No	Yes	Text box	String	N/A
5	Stop Location Name	Name of stop location.	No	Yes	Text box	String	N/A

Buttons/Hyperlinks

No	Function	Description	Validation	Outcome
2	Save Route	Save creating route to database.	N/A	Navigate to main page.
4	Clear Start Location Name	Clear start location name.	N/A	Start location name is cleared.
6	Clear Stop Location Name	Clear stop location name.	N/A	Stop location name is cleared.

5.1.5. Feedback bus route page



Figure 76 Feedback bus route page

Fields

No	Field Name	Description	Read only	Mandatory	Control Type	Data Type	Length
1	ChangeName	Choose type of feedback is "ChangeName"	No	Yes	Radio button	Boolean	N/A
2	ChangeNameText	Text of new name.	No	Yes	Text box	String	N/A
3	NotExist	Choose type of feedback is "NotExist"	No	Yes	Radio button	Boolean	N/A

Buttons/Hyperlinks

No	Function	Description	Validation	Outcome
4	SendFeedback	Send feedback to server.	N/A	Navigate back.
5	GoBack	Go back to previous page.	N/A	Navigate back.

5.1.6. Feedback bus station page

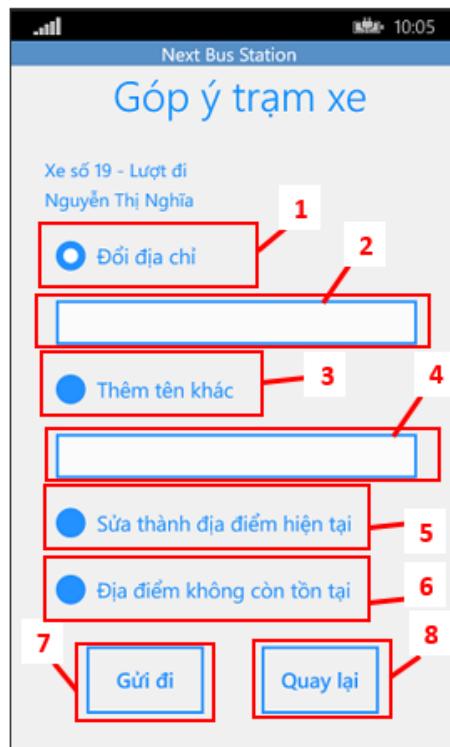


Figure 77 Feedback bus station page

Fields

No	Field Name	Description	Read only	Mandatory	Control Type	Data Type	Length
1	ChangeAddress	Choose type of feedback is “Change Address”	No	Yes	Radio button	Boolean	N/A
2	NewAddressText	Text of new address.	No	Yes	Text box	String	N/A
3	ChangeName	Choose type of feedback is “ChangeName”	No	Yes	Radio button	Boolean	N/A
4	ChangeNameText	Text of new name.	No	Yes	Text box	String	N/A
5	ChangeToCurrentLocation	Choose type of feedback is “ChangeTo Current Location”	No	Yes	Radio button	Boolean	N/A
6	NotExist	Choose type of feedback is “NotExist”	No	Yes	Radio button	Boolean	N/A

Buttons/Hyperlinks

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

7	SendFeedback	Send feedback to server.	N/A	Navigate back.
8	GoBack	Go back to previous page.	N/A	Navigate back.

5.1.7. Feedback location page



Figure 78 Feedback location page

Fields

No	Field Name	Description	Read only	Mandatory	Control Type	Data Type	Length
1	ChangeAddress	Choose type of feedback is "Change Address"	No	Yes	Radio button	Boolean	N/A
2	NewAddressText	Text of new address.	No	Yes	Text box	String	N/A
3	ChangeName	Choose type of feedback is "ChangeName"	No	Yes	Radio button	Boolean	N/A
4	ChangeNameText	Text of new name.	No	Yes	Text box	String	N/A
5	ChangeToCurrentLocation	Choose type of feedback is "ChangeTo Current Location"	No	Yes	Radio button	Boolean	N/A
6	NotExist	Choose type of feedback is "NotExist"	No	Yes	Radio button	Boolean	N/A

Buttons/Hyperlinks

No	Function	Description	Validation	Outcome
7	SendFeedback	Send feedback to server.	N/A	Navigate back.
8	GoBack	Go back to previous page.	N/A	Navigate back.

5.2. Web Interface Design

5.2.1. Manage bus route page

The screenshot shows a web application titled 'NEXT BUS STATION'. The main content area is labeled 'Manage Bus Route'. It features a table with columns: ID Buýt, Tên Xe Buýt, Số Xe Buýt, Lần Cập Nhật Cuối, Ghi Chú, Sửa, and Xoá. The table contains several rows of bus route data. At the top left, there's a navigation bar with links like 'Trang Chủ', 'Thông Tin Xe Buýt', 'Phản Hồi Của Người Dùng', 'Tạo Mới', and dropdown menus for 'Người Dùng', 'Xe Buýt', 'Trạm', 'Tuyến', and 'Địa Điểm'. A search bar is also present. Red numbers 1 through 13 are overlaid on various elements: 1 points to the 'Thông Tin Xe Buýt' link; 2 points to the 'Phản Hồi Của Người Dùng' link; 3 points to the 'Tạo Mới' button; 4 points to the dropdown menu for 'Xe Buýt'; 5 points to the search bar; 6 points to the 'Sửa' button in the first row of the table; 7 points to the 'Xoá' button in the same row; 8 points to the 'Trang Chủ' link; 9 points to the 'Phản Hồi Của Người Dùng' link; 10 points to the 'Xe Buýt' dropdown menu; 11 points to the 'Trạm' dropdown menu; 12 points to the 'Tuyến' dropdown menu; 13 points to the 'Địa Điểm' dropdown menu.

Figure 79 Manage Bus Route page

Field

No	Field Name	Description	Read Only	Mandatory	Control Type	Data Type	Length
1	Bus Route Information	Tab include Bus Route's information	No	N/A	Tab	String	N/A
2	User Report	Tab include users' report	No	N/A	Tab	String	N/A
4	Display row numbers	Select the number of row displayed for the table	No	N/A	Dropdownlist	String	N/A
5	Search	Search information	No	N/A	Textbox	String	N/A

Buttons/Hyperlinks

No	Function	Description	Validation	Outcome
3	Create Bus Route	Click to create new Bus Route	N/A	Show pop-up to add Bus Route infor value
6	Edit Bus Route	Click to edit Bus Route	N/A	Show pop-up to edit Bus Route infor value
7	Remove Bus	Click to remove Bus Route	N/A	Show confirm pop-

	Route			up
8	Main Page	Click to link to Main Page	N/A	N/A
9	Account Page	Click to link to Account Page	N/A	N/A
10	Bus Route Page	Click to link to Bus Route Page	N/A	N/A
11	Bus Station Page	Click to link to Bus Station Page	N/A	N/A
12	Bus Path Page	Click to link to Bus Path Page	N/A	N/A
13	Location Page	Click to link to Location Page	N/A	N/A

5.2.2. Edit bus path popup

The screenshot shows a modal dialog titled "Chinh Sửa Thông Tin Tuyến". It contains four dropdown menus: "Trạm Đầu" (Stop 1) with value "P-8999" (labeled 1), "Trạm Cuối" (Stop 2) with value "P-8994" (labeled 2), "Số Xe Buýt" (Bus Number) with value "P-01-True" (labeled 4), and "Thứ tự tuyến" (Order) with value "1" (labeled 3). At the bottom are two buttons: "Tiếp" (Next) (labeled 5) and "Hủy" (Cancel) (labeled 6).

Figure 80 Edit bus path popup

Field

No	Field Name	Description	Read Only	Mandatory	Control Type	Data Type	Length
1	Station From ID	Select station from ID	No	N/A	Dropdownlist	String	N/A
2	Station To ID	Select station to ID	No	N/A	Dropdownlist	String	N/A
3	Bus Path Number	Fill number of Bus Path	No	N/A	Textbox	String	N/A
4	BusRoute ID	Select BusRoute ID of Bus Path	No	N/A	Dropdownlist	String	N/A

Buttons/Hyperlinks

No	Function	Description	Validation	Outcome
5	Next	Click to move to map pop-up	N/A	Show pop-up include map
6	Close	Click to close pop-up	N/A	Pop-up closed
7	Close	Click to close pop-up	N/A	Pop-up closed

6. Database Design

6.1. Web Service Logical Diagram

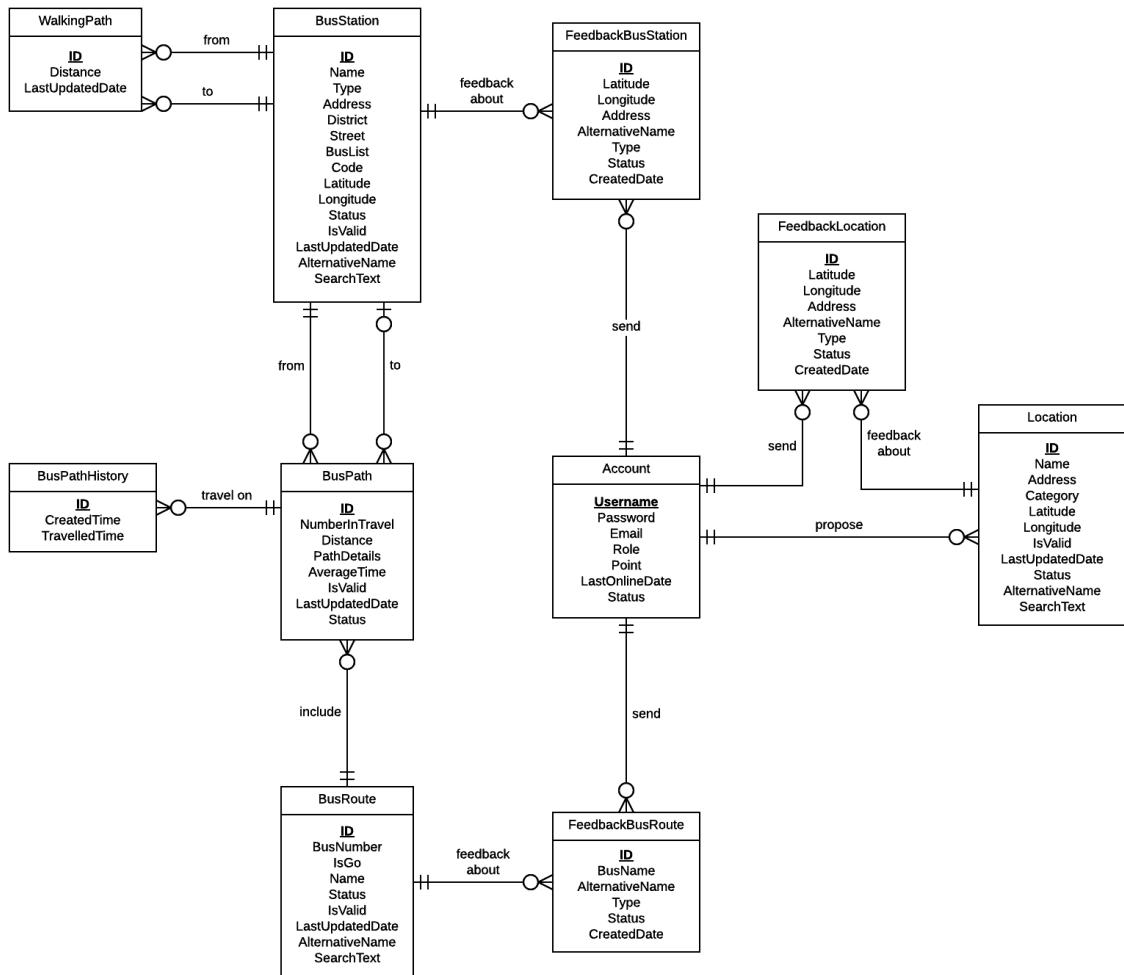


Figure 81 Web Service Logical Diagram

6.2. Web Service Data Dictionary

Entity Data dictionary: describe content of all entities	
Entity Name	Description
BusStation	Describe all bus stations in the system.
BusPath	Describe all bus paths in the system.
BusRoute	Describe all bus routes in the system.
WalkingPath	Describe all walking distances between two nearby bus stations in the system.
BusPathHistory	Describe all actual time need to travel in a bus path in the system.
Account	Describe all accounts in the system.
FeedbackBusStation	Describe all feedbacks for bus stations in the system.
FeedbackBusRoute	Describe all feedbacks for bus routes in the system.
FeedbackLocation	Describe all feedbacks for locations in the system.
Location	Describe all locations in the system.

Table 10 Web Entity Data Dictionary

Entity name	Attributes	Description	Domain	Null
BusStation	ID {PK}	Unique identifier of each bus station.	varchar(12)	No
	Name	Name of bus station.	nvarchar(80)	No
	Type	Type of bus station.	nvarchar(20)	No
	Address	Address of bus station.	nvarchar(120)	No
	District	District of bus station.	nvarchar(20)	No
	Street	Street of bus station.	nvarchar(80)	No
	BusList	List of buses which are available at bus station.	varchar(200)	No
	Code	Code of bus station.	varchar(20)	No
	Latitude	Latitude of bus station.	float	No
	Longitude	Longitude of bus station.	float	No
	Status	Status of bus station.	varchar(20)	No
	IsValid	Validity of bus station.	bit	No
BusPath	LastUpdatedDate	Latest date and time bus station has updated.	datetime	No
	AlternativeName	Alternative name of bus station.	nvarchar(MAX)	No
	SearchText	Text which is used for searching bus station.	nvarchar(MAX)	No

		metres.		
	PathDetails	List of coordinates which are used to define path.	varchar(MAX)	No
	AverageTime	Average time which the bus travel on bus path in reality, in second.	int	No
	IsValid	Validity of bus path.	bit	No
	LastUpdatedDate	Latest date and time bus path has updated.	datetime	No
	Status	Status of bus path.	varchar(20)	No
BusRoute	ID {PK}	Unique identifier of each bus route.	varchar(20)	No
	BusNumber	Number of bus route.	varchar(10)	No
	IsGo	Turn of bus route.	bit	No
	Name	Name of bus route.	nvarchar(100)	No
	Status	Status of bus route.	varchar(20)	No
	IsValid	Validity of bus route.	bit	No
	LastUpdatedDate	Latest date and time bus route has updated.	datetime	No
	AlternativeName	Alternative name of bus route.	nvarchar(MAX)	Yes
	SearchText	Text which is used for searching bus route.	nvarchar(MAX)	Yes
WalkingPath	ID {PK}	Unique identifier of each bus station	int	No

		distance.		
	Distance	Walking distance between 2 stations.	float	No
	LastUpdatedDate	Latest date and time bus station distance has updated.	datetime	No
BusPathHistory	ID {PK}	Unique identifier of each history.	int	No
	CreatedTime	Travelled time.	datetime	No
	TravelledTime	Time which consumes to travel on bus path, in second.	int	No
Account	Username {PK}	Username for logging to account.	varchar(20)	No
	Password	Password for logging to account.	varchar(50)	No
	Email	Email of account	Varchar(50)	No
	Role	Role which account is assigned.	varchar(50)	No
	Point	Point of account.	int	No
	LastOnlineDate	Latest date and time account has connected to server.	datetime	No
	Status	Status of account.	varchar(10)	No
FeedbackBusStation	ID {PK}	Unique identifier of each feedback bus station.	int	No
	Latitude	New latitude of bus station in feedback.	float	Yes

	Longitude	New longitude of bus station in feedback.	float	Yes
	Address	New address of bus station in feedback.	nvarchar(120)	Yes
	AlternativeName	New alternative name of bus station in feedback.	nvarchar(80)	Yes
	Type	Type of bus station	varchar(30)	No
	Status	Status of feedback bus station	varchar(15)	No
	CreatedDate	Date and time when feedback is created.	datetime	No
FeedbackBusRoute	ID {PK}	Unique identifier of each feedback bus route.	int	No
	BusName	New name of bus route in feedback.	nvarchar(100)	Yes
	AlternativeName	New alternative name of bus route in feedback.	nvarchar(100)	Yes
	Type	Type of bus route	varchar(30)	No
	Status	Status of feedback bus route	varchar(15)	No
	CreatedDate	Date and time when feedback is created.	datetime	No
FeedbackLocation	ID {PK}	Unique identifier of each feedback location.	int	No

	Longitude	New longitude of location in feedback.	float	Yes
	Latitude	New latitude of location in feedback.	float	Yes
	Address	New address of location in feedback.	nvarchar(120)	Yes
	AlternativeName	New alternative name of location in feedback.	nvarchar(100)	Yes
	Type	Type of location	varchar(30)	No
	Status	Status of feedback location	varchar(15)	No
	CreatedDate	Date and time when feedback is created.	datetime	No
Location	ID {PK}	Unique identifier of each location.	int	No
	Name	Name of location.	nvarchar(100)	No
	Address	Address of location.	nvarchar(120)	No
	Category	Category of location.	nvarchar(20)	No
	Latitude	Latitude of location.	float	No
	Longitude	Longitude of location	float	No
	IsValid	Validity of location.	bit	No
	LastUpdatedDate	Latest date and time location has updated.	datetime	No
	Status	Status of location.	varchar(20)	No
	AlternativeName	Alternative name of location.	nvarchar(MAX)	Yes

	SearchText	Text which is used for searching location.	nvarchar(MAX)	No
--	------------	--	---------------	----

Table 11 Web Detail Data Dictionary

Business integrity constrain:

- FeedbackBusStation's content is based on type: if type is "ChangeName", Name attribute is required. If type is "ChangeToCurrentLocation", Latitude and Longitude attribute is required. If type is "ChangeToAddress", Address attribute is required.
- FeedbackLocation's content is based on type: if type is "ChangeName", Name attribute is required. If type is "ChangeToCurrentLocation", Latitude and Longitude attribute is required. If type is "ChangeToAddress", Address attribute is required.

6.3. Mobile Application Logical Diagram

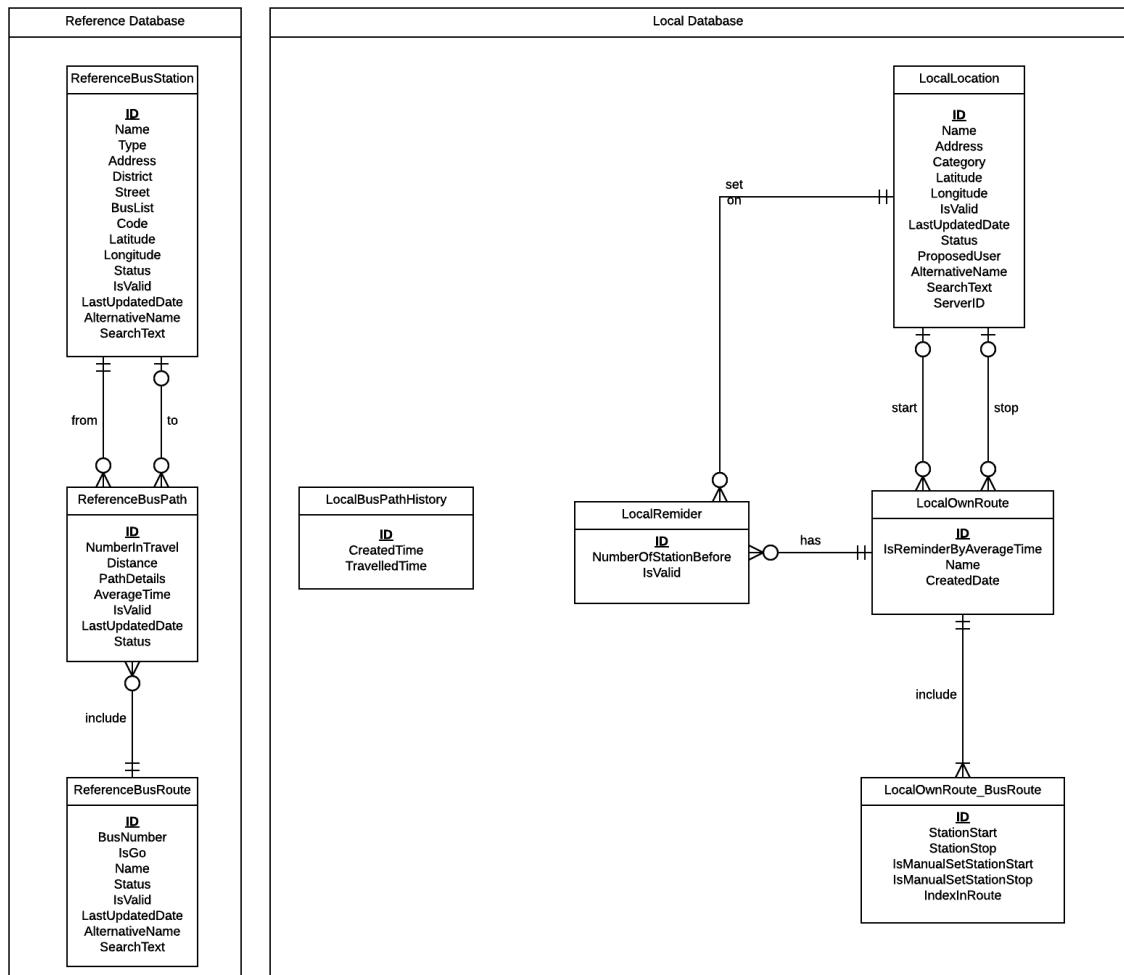


Figure 82 Mobile Application Logical Diagram

6.4. Mobile Application Data Dictionary

Entity Data dictionary: describe content of all entities

Entity Name	Description
ReferenceBusStation	Describe all bus stations in the system.
ReferenceBusPath	Describe all bus paths in the system.
ReferenceBusRoute	Describe all bus routes in the system.
LocalBusPathHistory	Describe all actual time need to travel in a bus path in the system, which member has saved while travelling.
LocalReminder	Describe all reminders of member in the system.
LocalLocation	Describe all locations of member in the system.
LocalOwnRoute	Describe all own route of member in the system.
LocalOwnRoute_BusRoute	Describe all bus routes included in own routes in the system.

Table 12 Mobile Application Entity Data Dictionary

Entity name	Attributes	Description	Domain	Null
ReferenceBusStation	ID {PK}	Unique identifier of each bus station.	varchar(12)	No
	Name	Name of bus station.	nvarchar(80)	No
	Type	Type of bus station.	nvarchar(20)	No
	Address	Address of bus station.	nvarchar(120)	No
	District	District of bus station.	nvarchar(20)	No
	Street	Street of bus station.	nvarchar(80)	No
	BusList	List of buses which are available at bus station.	varchar(200)	No
	Code	Code of bus station.	varchar(20)	No
	Latitude	Latitude of bus station.	float	No
	Longitude	Longitude of bus station.	float	No
	Status	Status of bus station.	varchar(20)	No
	IsValid	Validity of bus station.	bit	No
	LastUpdatedDate	Latest date and time bus station has updated.	datetime	No
	AlternativeName	Alternative name of bus station.	nvarchar(MAX)	No

	SearchText	Text which is used for searching bus station.	nvarchar(MAX)	No
ReferenceBusPath	ID {PK}	Unique identifier of each bus path.	varchar(15)	No
	NumberInTravel	The order of bus path in bus route.	int	No
	Distance	Distance of bus path in metres.	float	No
	PathDetails	List of coordinates which are used to define path.	varchar(MAX)	No
	AverageTime	Average time which the bus travel on bus path in reality, in second.	int	No
	IsValid	Validity of bus path.	bit	No
	LastUpdatedDate	Latest date and time bus path has updated.	datetime	No
	Status	Status of bus path.	varchar(20)	No
ReferenceBusRoute	ID {PK}	Unique identifier of each bus route.	varchar(20)	No
	BusNumber	Number of bus route.	varchar(10)	No
	IsGo	Turn of bus route.	bit	No
	Name	Name of bus route.	nvarchar(100)	No
	Status	Status of bus route.	varchar(20)	No
	IsValid	Validity of bus route.	bit	No
	LastUpdatedDate	Latest date and time bus	datetime	No

		route has updated.		
	AlternativeName	Alternative name of bus route.	nvarchar(MAX)	Yes
	SearchText	Text which is used for searching bus route.	nvarchar(MAX)	Yes
LocalBusPathHistory	ID {PK}	Unique identifier of each travel history	int	No
	CreatedTime	Travelled time.	datetime	No
	TravelledTime	Time which consumes to travel on bus path, in second.	int	No
LocalReminder	ID {PK}	Unique identifier of each reminder.	int	No
	NumberOfStationBefore	Number of station which reminder is alerted before the expected station comes.	int	No
	IsValid	Reminder's validity.	bit	No
LocalLocation	ID {PK}	Unique identifier of each location.	int	No
	Name	Name of location.	nvarchar(100)	No
	Address	Address of location.	nvarchar(120)	No
	Category	Category of location.	nvarchar(20)	No
	Latitude	Latitude of location.	float	No
	Longitude	Longitude of location	float	No

	IsValid	Validity of location.	bit	No
	LastUpdatedDate	Latest date and time location has updated.	datetime	No
	Status	Status of location.	varchar(20)	No
	ProposedUser	User who propose the location	varchar(20)	Yes
	AlternativeName	Alternative name of location.	nvarchar(MAX)	Yes
	SearchText	Text which is used for searching location.	nvarchar(MAX)	No
	ServerID	ID of location in server.	int	Yes
LocalOwnRoute	ID {PK}	Unique identifier of each own route	int	No
	IsReminderByAverageTime	Check if bus route has "reminder by average time" enabled or not.	bit	No
	Name	Name of own route	varchar(100)	No
	CreatedDate	Created date of own route	datetime	No
LocalOwnRoute_Bus Route	ID {PK}	Unique identifier of each bus route in each own route.	int	No
	StationStart	Index of station where bus route starts.	int	No
	StationStop	Index of station where bus route stops.	int	No

	IsManualSetStation Start	Check if start station is set manually or automatically.	bit	No
	IsManualSetStation Stop	Check if stop station is set manually or automatically.	bit	No
	IndexInRoute	Index of bus route in own route.	int	No
	LocalOwnRouteID	Own route.	int	No
	BusRouteID	Bus route.	varchar(20)	No

Table 13 Mobile Detail Data Dictionary

Business integrity constrain:

- ReferenceBusPath, ReferenceBusStation, ReferenceBusRoute's data is get from web database. All changes of data in web database will be update later in mobile application database.
- Relationship between LocalDatabase and ReferenceDatabase:

Attribute of LocalDatabase Entity	Respective attribute of ReferenceDatabase Entity
BusPathID of LocalBusPathHistory	ID of ReferenceBusPath
ID of LocalBusRoute	ID of ReferenceBusRoute
StationID of LocalReminder	ID of ReferenceBusStation
BusRouteID of LocalReminder	ID of ReferenceBusRoute
BusRouteID of LocalOwnRoute_BusRoute	ID of ReferenceBusRoute

Table 14 Relationship between LocalDatabase and ReferenceDatabase

- LocalReminder is set on either own route or bus route.
- LocalReminder is set on either station or location.

7.Algorithms

7.1. Find Next Station

7.1.1. Definition

- Next station is the station the bus is currently heading to.

7.1.2. Define Problem

- When user determined current location, by GPS or chosen location, system has to calculate which is the station of current route that bus is heading to.
- Current location of user is taken in form of coordinates (latitude and longitude).

- Data of bus route which used to calculate is path details attribute of bus paths in bus route. PathDetails attribute is a string which express consecutive pairs of coordinates which create the path between two stations.

7.1.3. Solution

- To solve this problem, we follow these steps:
 - Step 1: find stations around current location (to minimize number of bus path we have to check).
 - If not station is found, then return next station cannot be found.
 - If station(s) is found, create min distance value with initial values equals to a very large number (for example 100000).
 - Step 2: for each found station:
 - Get corresponding bus path of current route which has start station is checking station, then get PathDetails attribute of that bus path.
 - In database, PathDetails attribute is stored in form of a string which has format: “<longitude1><comma><latitude1><space><longitude2><comma><latitude2><space>...”.
 - For example: “106.6106,10.72851 106.608751,10.72582 ”
 - Change PathDetails attribute to list of coordinates by splitting the string and converting the values.
 - Step 3: which each list of coordinates, calculate distance from current location to the line segment which is created from each two consecutive coordinates taken in path details. To calculate the distance from one point onto a line, we are following these steps:
 - Step 3.1: Calculate projection of the point onto the line. Formula to calculate projection point $P(x_p, y_p)$ of one point $A(x_0, y_0)$ onto a line which made from $B(x_1, y_1)$ and $C(x_2, y_2)$:
$$U = \frac{(x_0 - x_1)(x_2 - x_1) + (y_0 - y_1)(y_2 - y_1)}{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

$$x_p = x_1 + U(x_2 - x_1)$$

$$y_p = y_1 + U(y_2 - y_1)$$
 - Step 3.2: Check if the projection is on the line segment. If no, the distance is not valid and won't be used in further steps. If yes, we calculate the distance from the point to the found projection. Formula to check if projection $P(x_p, y_p)$ is on the line segment which made from $B(x_1, y_1)$ and $C(x_2, y_2)$:
$$\text{result} = (x_p \geq \min(x_1, x_2) \text{ and } x_p \leq \max(x_1, x_2)) \text{ and } (y_p \geq \min(y_1, y_2) \text{ and } y_p \leq \max(y_1, y_2))$$

Formula to calculate distance between 2 points $A(x_1, y_1)$ and $B(x_2, y_2)$.

$$distance = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$$

- Step 3.3: If calculated distance is valid, compare it with min distance.
 - If smaller, update the min distance, clear old marked bus path and mark the checking bus path.
 - If equals, mark the checking bus path.
- Step 4: Check the marked bus path(s):
 - If there are no marked bus path: Next station is not found.
 - If there are more than one marked bus path, we check:
 - If all those bus path has the same bus route ID attribute, the first calculated bus path is taken as the result bus path, next station is that bus path's StationTo attribute.
 - If there are more than one bus route ID attribute, system will show popup for user to choose the bus they are currently travelling. The bus path with chosen bus route ID will always be taken as the result bus path later in this case, next station is that bus path's StationTo attribute.
 - If there are only one marked bus path, next station is that bus path's StationTo attribute.

7.1.4. Flow chart

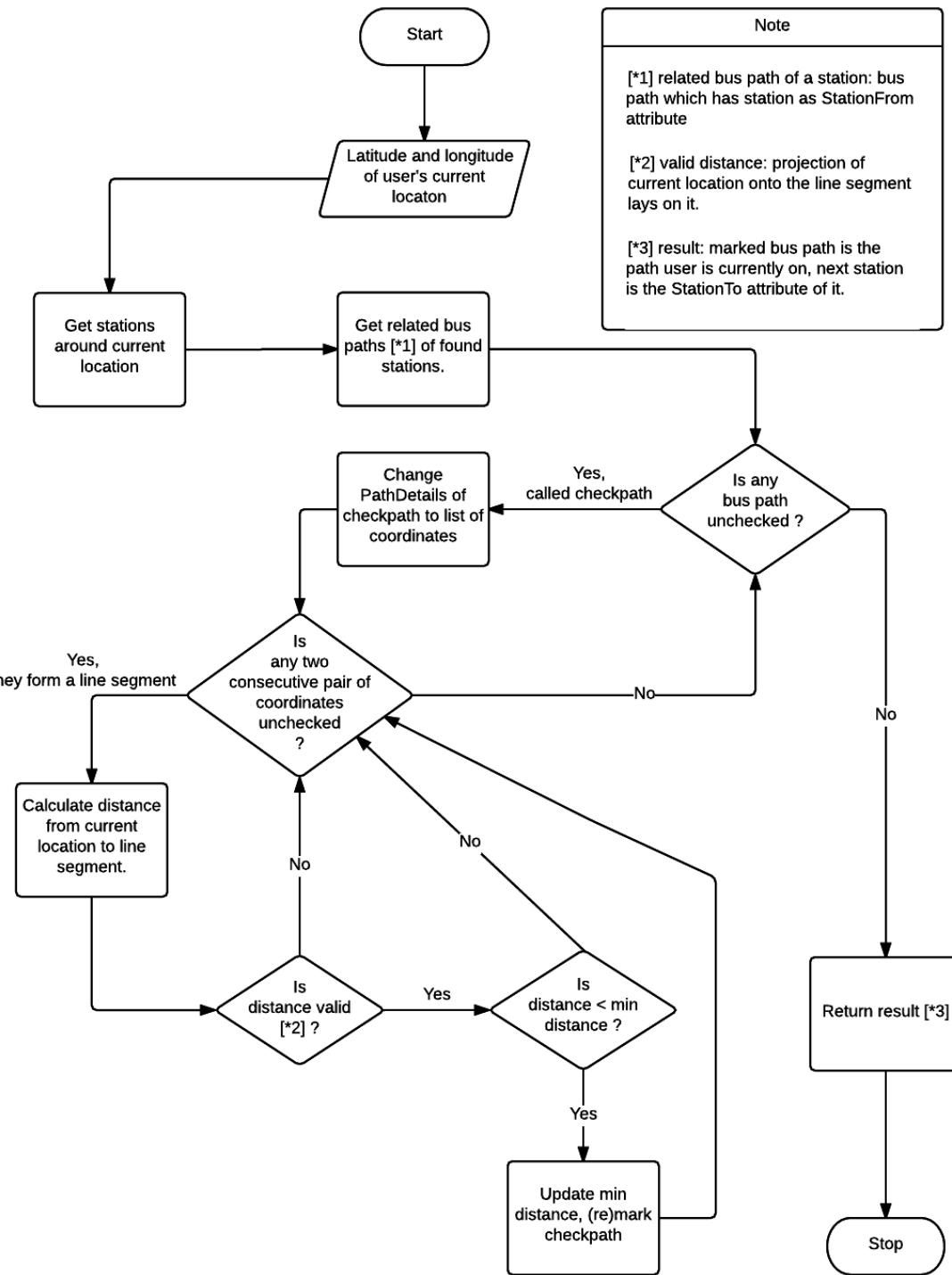


Figure 83 Find next station algorithm

7.2. Search Bus Rote

7.2.1. Definition

- User want to search route between 2 locations.

7.2.2. Define Problem

- User gives 2 locations, start and stop location. System search for suitable bus routes, based on user's search settings, for user to travel from start location to stop location.

7.2.3. Attribute Definition

- Length list: list which stored paths between two bus stations in system.
 - Each element of length list represent one bus station.
 - Each element of length list contains a list of paths of bus stations which has a path to go from represent station to, called LengthElement.
 - Each LengthElement has following attribute: BusRouteID (represent the bus which is used to travel), DistanceWalking (distance if walking), DistanceOnBus (distance if travel by bus) , StationToID (ID of station to).
 - If path is travelled by bus, Distance attribute will equals to 0.
 - If path is travelled by walking, DistanceOnBus will equals to 0, BusRouteID will equals to “-” (none).
- ResultList list: list which stored shortest routes from start location to each station.
 - Each element of result list represent one bus station.
 - Each element of result list contains lists which represent the attributes shortest route:
 - BusRoute: list of string, represent current travelling bus route ID.
 - BusStation: list of string, represent current travelling bus station.
 - NumberOfBus: list of int, represent total number of bus used to travel.
 - DistanceOnBus: list of double, represent total distance travelled by bus.
 - DistanceWalking: list of double, total distance travelled by walking.
 - IsCheck: list of boolean, to check if this shortest route is used in process or not.
 - Accumulate bus station: list of string, represent all station which has been travelled in the route, is used to show result.
 - Accumulate bus route: list of string, represent all bus route which has been travelled by in the route, is used to show result.
 - Note: each above element is a separate list. One shortest route's attributes is represent by elements at a certain index of each list. We use this way to easily take all accumulate bus routes of one station for checking later.
- Weight array: array of double which stored weight of shortest routes.

- Each element of weight array represent one bus station, with the meaning: weight of route travel from start location to represent station.
- In least bus mode: each element is represent number of bus routes which is used to travel from start location to represent station.
- In least walking mode: each element is represent total walking distance from start location to represent station.
- ListStation: list of station which is taken from reference database and is ordered by id.
- Check array: array of boolean which is used to check if a station should be checked (true) or not (false).

7.2.4. Solution

- Search route algorithm is based on Dijkstra's algorithm.
- To solve this problem, we follow these steps:
 - Validate: If there is no bus stations around start location or stop location: show error message and stop.
 - Step 1 - Search Preparation: Get Length array stored in device, if not exist, get it from server and store it in device for later use. Create ResultList. Create Weight array with all infinite value. Create Check array with all true values. Get list stations.
 - Step 2 - Add start and stop location to lists:
 - With start location:
 - Length list: add one element, called Length[start]. With each stations around the start location: add one LengthElement to Length[start] which represent the walking path from start location to the station.
 - Weight array: add one element, use as the initial value later.
 - List stations: add one element, use as start point in later searched route, called ListStation[start].
 - Stop location:
 - Length list: add one element, called Length[stop]. With each stations around the stop location: add one LengthElement to that station's elements in length list which represent that walking path from that station to stop location.
 - Weight array: add one element, use as the result weight value.
 - List stations: add one element, use stop point in later searched route, called ListStation[stop].
 - Step 3 - Search route
 - Step 3.1 - Search Initial: set Weight[start] = 0, set initial attributes of start location (treated as a station in list stations at this time) in ResultList.
 - In normal mode (search between two locations):
 - BusRoute: Add "0" string (no bus route at start location)
 - BusStation: Add "0" string (no bus station at start location)

- NumberOfBus: Add value 0 (no travelled bus at start location).
 - DistanceOnBus: Add value 0 (has not been travelled by bus).
 - DistanceWalking: Add value 0 (has not walked).
 - IsCheck: Add true value (has not been checked).
 - AccumulateBusStation: Add “0” string.
 - AccumulateBusRoute: Add “0” string.
- In on-bus mode (search between current location while travelling on bus to another location):
 - BusRoute: Add current bus route’s ID.
 - BusStation: Add current next station’s ID.
 - NumberOfBus: Add value 1.
 - DistanceOnBus: Add value 0 (previous travelled distance before searching is not count).
 - DistanceWalking: Add value 0 (previous walking distance before searching is not count).
 - IsCheck: Add true value (has not been checked).
 - AccumulateBusStation: Add current bus route’s ID.
 - AccumulateBusRoute: Add current next station’s ID.
- Step 3.2

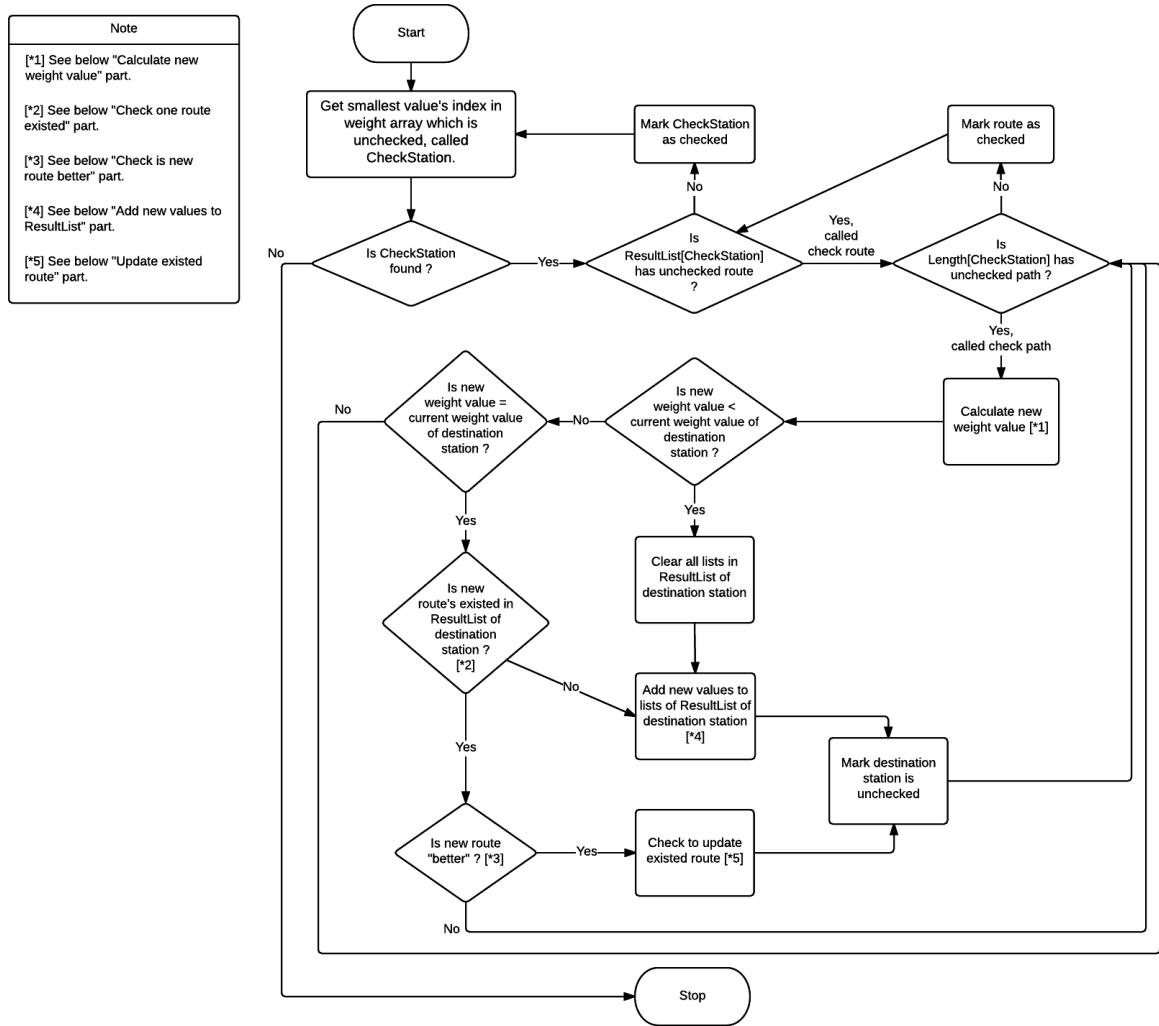


Figure 84 Search route algorithm

- [*1] Calculate new weight value
 - In least bus mode:

BusRoute value of check route	BusRouteID value of check path	New weight value
Walking	Walking	= weight value of CheckStation
Walking	(Any) Bus route	= weight value of CheckStation + 1
Bus route	Walking	= weight value of CheckStation
Bus route	Bus route with same ID	= weight value of CheckStation
Bus route	Bus route with different ID	= weight value of CheckStation + 1

- In least walking mode: new weight value = weight value of CheckStation + DistanceWalking value of check path.
 - [*2] Check one route existed
 - Get the short form of route by using AccumulateBusRoute value of check route + BusRouteID of check path ("-" is walking):

AccumulateBusRoute value of check route	BusRouteID value of check path	Short form
“- - 19 19 19 - 32 32 - -”	“55”	“- 19 - 32 - 55”
“19 19 - - 32 - - 55”	55	“19 - 32 - 55”

- Then compare with short forms of every routes in ResultList of station to (StationToID value in check path) to check whether it is existed or not.
- [*3] Check is new route better
 - If new route's NumberOfRoute value < existed route's NumberOfRoute value, it is better.
 - If new route's NumberOfRoute value = existed route's NumberOfRoute value and new route's DistanceOnBus value < existed route's DistanceOnBus value, it is better.
 - The other cases count as not better.
- [*4] Add new values to ResultList
 - BusRoute: Add check path's BusRouteID value.
 - BusStation: Add check path's StationToID value.
 - NumberOfBus: Add calculate value as new weight in least bus mode.
 - DistanceOnBus: Add DistanceOnBus value of check route + DistanceOnBus value of check path.
 - DistanceWalking: Add DistanceWalking value of check route + DistanceWalking value of check path.
 - IsCheck: Add true value.
 - AccumulateBusStation: AccumulateBusStation value of check route + StationToID of check path (separated by a space).
 - AccumulateBusRoute: AccumulateBusRoute value of check route + BusRouteID of check path (separated by a space).
- [*5] Update existed route
 - Change all values of existed route to values of new route (values is calculated as new values in “Add new values to ResultList” action)
- Step 4 - Remove all added data of step 2 from Length list to ensure accuracy for later search times.
- Step 5 - Select Result: Because of restricted in walking distance (in least bus mode) or number of bus (in least walking mode), there may not be any absolute suitable result in ResultList[stop] but only relatively results in ResultList[stations around stop location]. So we take all routes in ResultList of all stations around stop location and select some good ones as result of search function. Taking decision is based on:
 - If route 1 and route 2 has different short form, take both.

- If route 1 and route 2 has the same short form, take which has DistanceWalking value smaller. If DistanceWalking value is equals, take which has DistanceOnBus value smaller.

7.2.5. Search Time Testing

- Average time of search time (step 3 in solution): below 1 minute.
- All search results below are done in Windows Phone 8.1 emulator and in online mode (walking distance is taken from service). Search time may be different for each times with same locations, but results are the same.
- Least bus mode - Walking distance: 300m

Start location	Stop location	Number of selected results	Search time (Step 3 in solution only)	Total time
Đầm Sen	Suối Tiên	10	1.88 seconds	5.08 seconds
Nguyễn Văn Linh street	Quang Trung Software city	7	2.70 seconds	5.38 seconds
23/9 park	Quang Trung Software city	2	0.82 seconds	5.43 seconds

- Least walking mode - Number of bus: 2

Start location	Stop location	Number of selected results	Search time (Step 3 in solution only)	Total time
Đầm Sen	Suối Tiên	20	7.50 seconds	10.11 seconds
Nguyễn Văn Linh street	Quang Trung Software city	2	2.40 seconds	4.97 seconds
23/9 park	Quang Trung Software city	3	5.69 seconds	10 seconds

7.2.6. Reference

http://en.wikipedia.org/wiki/Dijkstra%27s_algorithm

7.3. Automatically Create Route

7.3.1. Define Problem

- Normally, people won't use search route function of one system unless they have faith in that system. When users want to know bus routes to travel between 2 locations, they often ask the others.
- Users provide 2 locations and a list of bus routes (bus number only, no turn information), create route from those information.

7.3.2. Attribute Definition

- Route: means own route, is LocalOwnRoute class in local database of mobile application, includes a list of RouteElement, start location (optional) and/or stop location (optional).
- RouteElement: means each bus route in route, is LocalOwnRoute_BusRoute class in local database of mobile

application, includes a BusRouteID value, station start value and station stop value.

7.3.3. Solution

- To solve this problem, we follow these steps:
- Step 1 - Find correct turn of each bus route to form a route from start location to stop location.

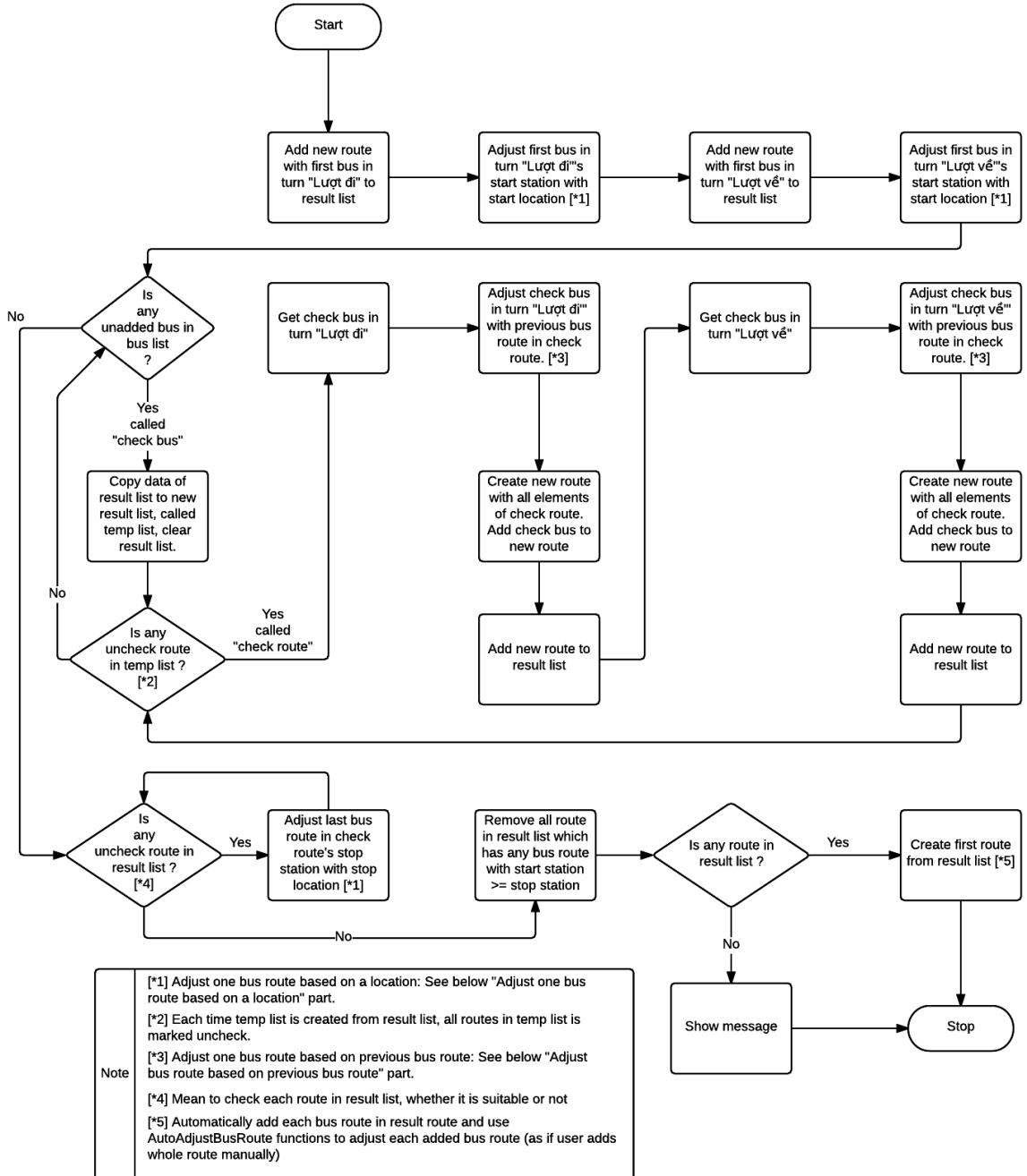


Figure 85 Create route algorithm - Select turns

- [*1] Adjust one bus route based on a location
 - If location is start location, find which station in bus route have the smallest distance to start location. Mark that station

- as start station of bus route. If not found, start station is first station of bus route.
- If location is stop location, find which station in bus route have the smallest distance to stop location. Mark that station as stop station of bus route. If not found, stop station is last station of bus route.
 - To increase speed, distance in this part is calculated as distance between 2 points (instead of distance of walking path).
 - [*3] Adjust bus route based on previous bus route
 - Find which station of checking bus route and which station of previous bus route has the smallest distance to each other. Marked station of checking bus route is start station of it. Marked station of previous bus route is stop station of it.
 - To increase speed, distance in this part is calculated as distance between 2 points (instead of distance of walking path).
 - Step 2 - Create first result route (if any).
 - Auto adjust route function when one bus route is added (as well as edited / removed / change index, or start location is changed, or stop location is changed)

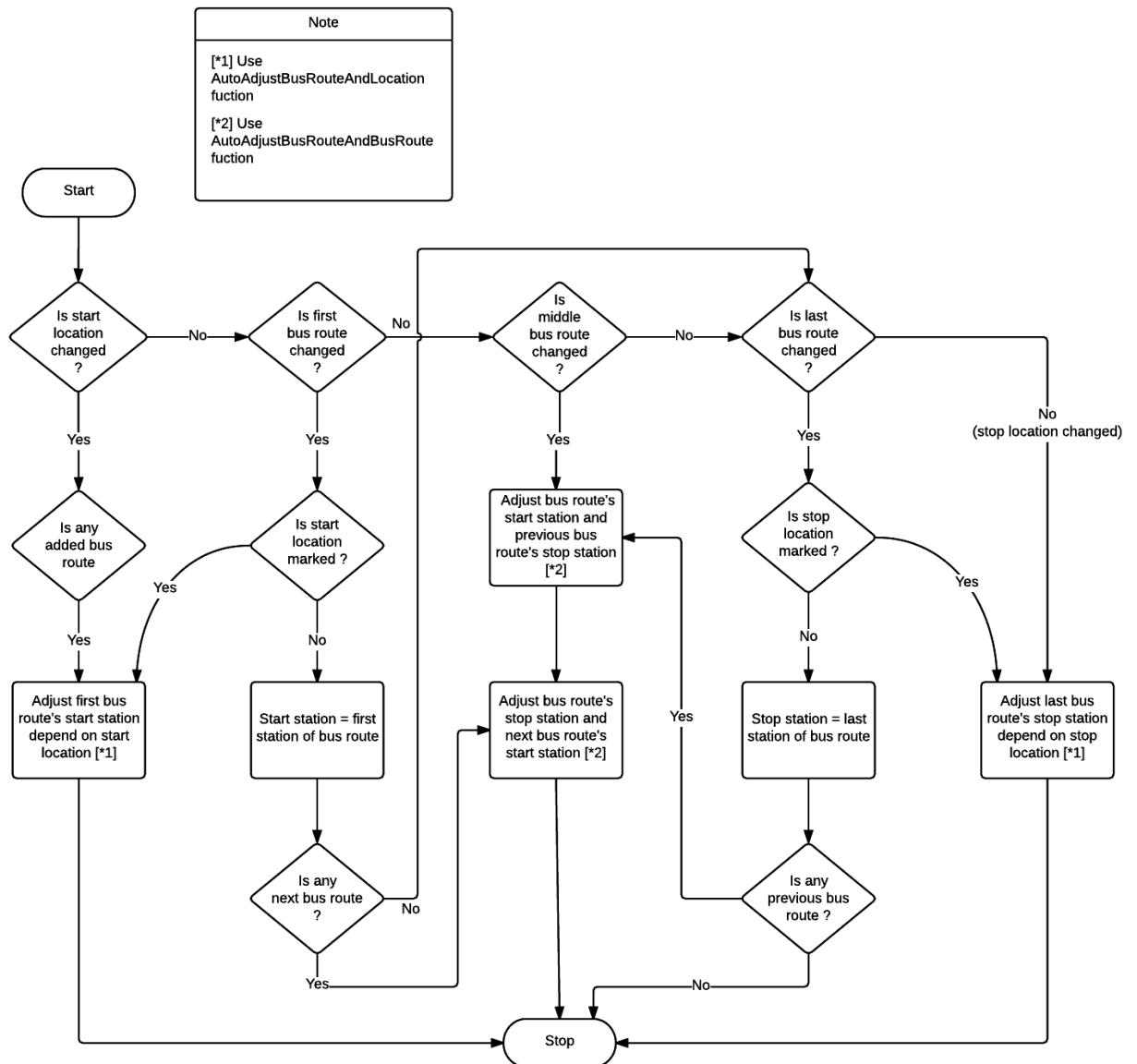


Figure 86 Create route algorithm - Adjust route

- [*1] AutoAdjustBusRouteAndLocation function: use to set suitable start station of bus route and start location or suitable stop station of bus route and stop location. In this part, distance is calculated as distance of walking path between 2 locations to ensure accuracy.

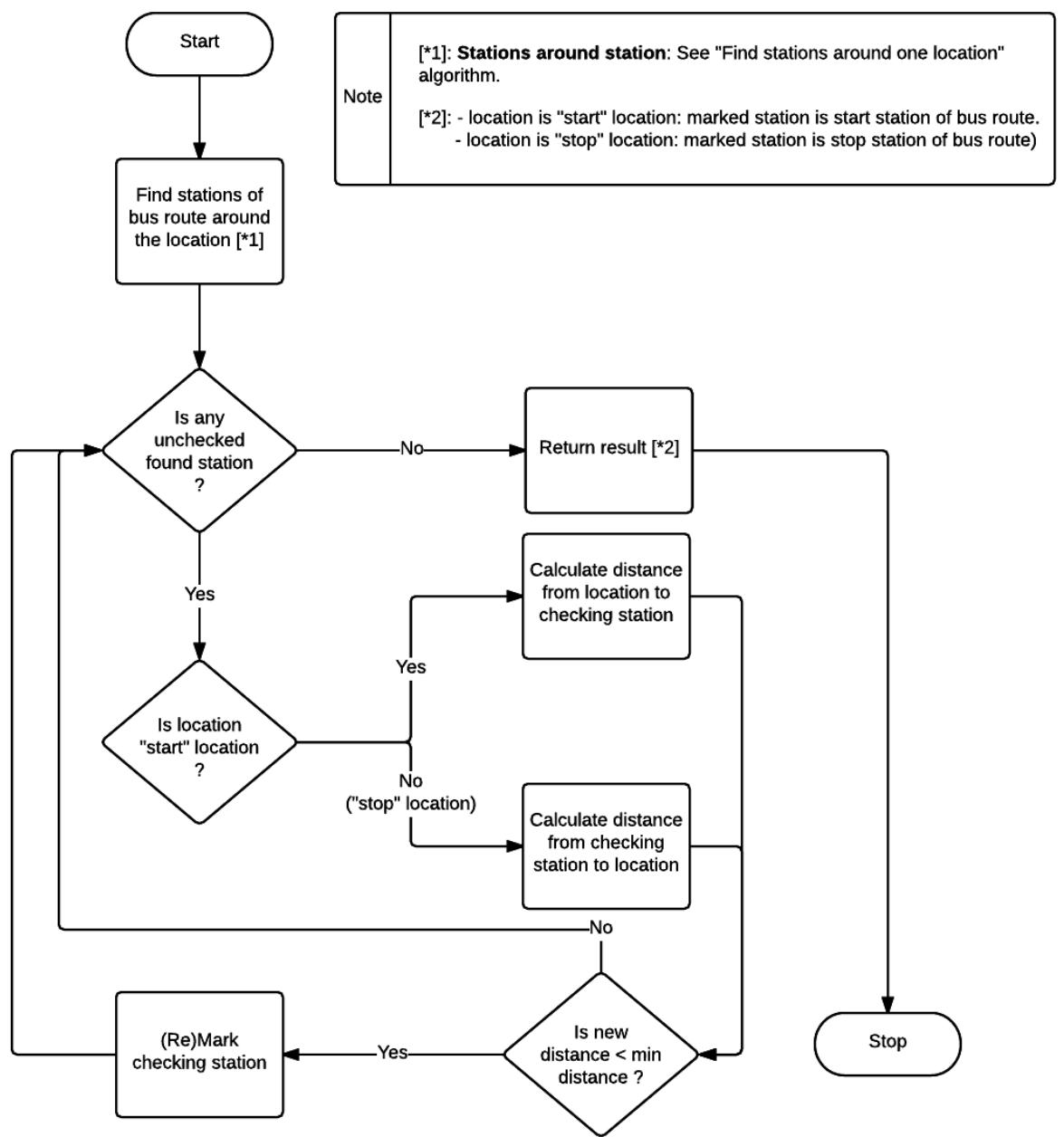


Figure 87 Create route algorithm - Adjust bus route and location

- [*2] AutoAdjustBusRouteAndBusRoute function: use to set stop station of bus route 1 and start station of bus route 2 (in order).

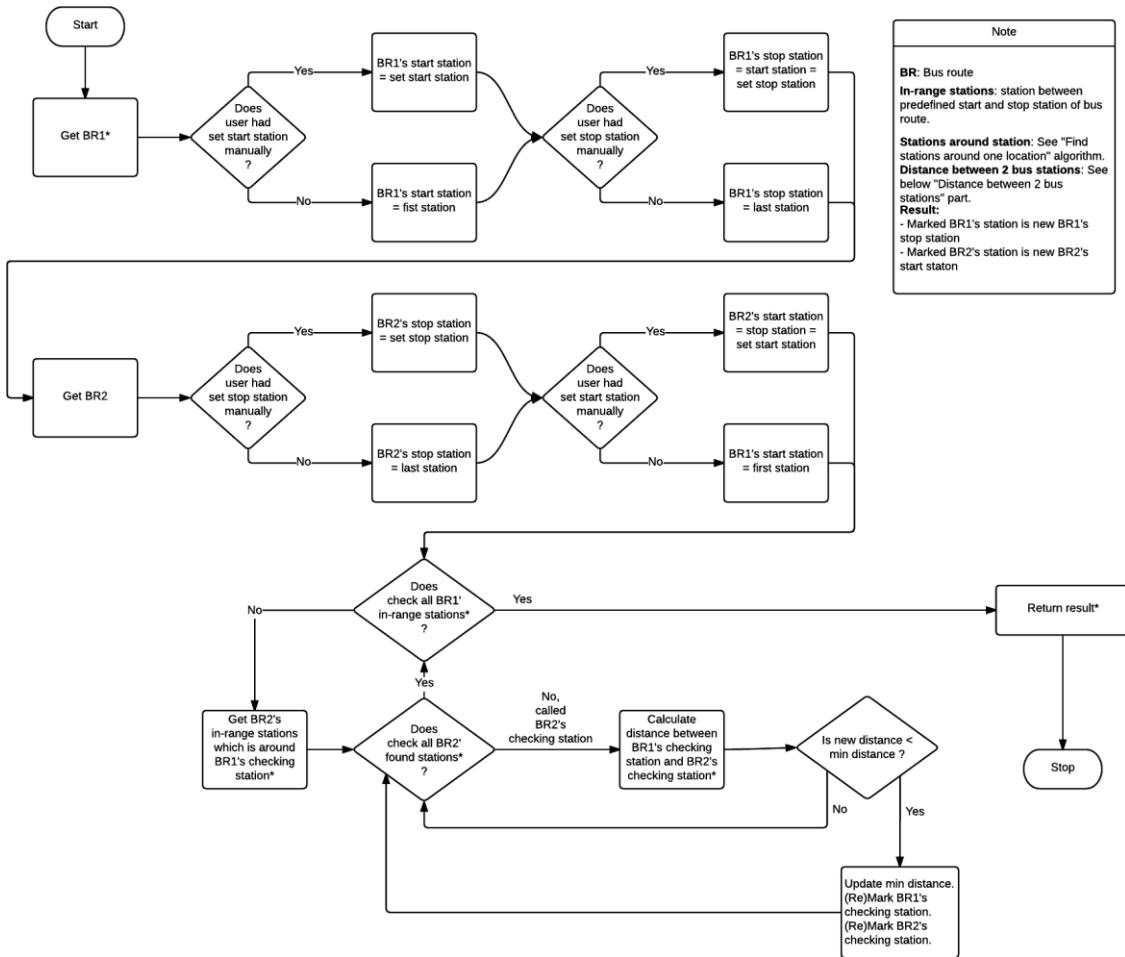


Figure 88 Create route algorithm - Adjust two bus routes

- Distance between 2 bus stations: in this part, distance is calculated as distance of walking path between 2 locations to ensure accuracy.

7.4. Find stations around one location

7.4.1. Define Problem

- Given latitude and longitude of one location, find stations around that location.

7.4.2. Solution

- By default, one station is counted as around one location when latitude of that station is between latitude of location ± 0.005 and longitude of that station is between longitude of location ± 0.005 .
- 0.005 is a number which give best number of stations around in functions which need to find stations around one location during testing. This number may be varied in each function use it, based on what need to be prioritized. Sometimes we need only a few stations around one location, 0.0025 can be used instead. Return too many stations around one location may slowdown later processing.

7.5. Parse and Update data

7.5.1. Define Problem

- Buses' data is gotten from <http://buyttphcm.com.vn/> as initial data for system.
- While using application, member may send feedback about stations for buses if data is not correct. Staff may correct the data based on feedbacks or by checking in real life.
- Parsed data from <http://buyttphcm.com.vn/> later will be stored temporarily in system. Staff should decide to accept or reject changes data later.
- Summary, there are 3 main parts of parsing and updating data:
 - Parse data from <http://buyttphcm.com.vn/>, check if parsed data is different from current data.
 - Approve parsed data of one bus route.
 - Update one bus route in mobile application.

7.5.2. Solution

7.5.2.1 Parse data from <http://buyttphcm.com.vn/>

- By using Fiddler when choosing one bus route in <http://buyttphcm.com.vn/> we can get a link which is used to transfer one bus route data:
<http://mapbus.ebms.vn/ajax.aspx?action=listRouteStations&rid=1&isgo=true>
 - By changing "rid" and "isgo" parameter in that link, we can get each bus route data in <http://buyttphcm.com.vn/>.
- By using Fiddler when choosing one station in <http://buyttphcm.com.vn/>, we can get a link which is used to transfer bus station data. Parsed data are: district, street, bus list. Those data are then added to corresponding bus station data.
- Summary, parsed data includes:

Bus Route	Bus Path	Bus Station
- Bus number - Turn - Bus name	- Path details - Order of bus path in route - Distance of bus path	- ID - Name - Longitude - Latitude - Address - Code - District - Street - Bus list

- Each data is then compared with current data in database as:

Data	Changes	Add parsed data to database as "Change"	Add parsed data to database as "Add"
Bus Station	Name, Type, Address, District, Street, BusList, Code, Latitude, Longitude Corresponding bus station	X	X

	("P-“ + parsed station's id) does not existed in database.		
Bus Path	Information of start station or stop station is changed.	X	
	StationFromID or StationToID is changed.	X	
	Distance is changed.	X	
	Corresponding bus path (bus path which has the same attribute: BusNumber, IsGo, NumberInTravel) does not existed in database.		X
Bus Route	Information of any bus path is changed.	X	
	Bus name, number of bus path is changed.	X	
	Corresponding bus route (bus route which has the same attribute: BusNumber, IsGo) does not existed in database.		X

- Parsed data are added to database in parsed tables. Staff chooses to approve later to applied parsed data to current data in database.

7.5.2.2 Approve parsed data of one bus route

- When staff approve one parsed bus route, all data in bus route, include all related bus path and bus station, is approved.
- Process to approve one parsed bus station:
 - If parsed bus station is “Change”
 - Get corresponding current bus station in database.
 - Update current bus station’s data based on parsed bus station’s data.
 - Delete parsed bus station from database.
 - If parsed bus station is “Add”
 - Add new bus station with parsed bus station’s data.
 - Delete parsed bus station from database.
- Process to approve one parsed bus path:
 - Approve related start parsed bus station (if any).
 - Approve related stop parsed bus station (if any).
 - If parsed bus path is “Change”
 - Get corresponding current bus path in database.

- Update current bus path's data based on parsed bus station's data.
 - Delete parsed bus path from database.
- If parsed bus path is "Add"
 - Add new bus path with parsed bus path's data.
 - Delete parsed bus path from database.
- Process to approve one parsed bus route:
- If parsed bus route is "Change"
 - Approve all related parsed bus path (if any).
 - Get corresponding current bus route in database.
 - Update current bus route's data based on parsed bus route's data.
 - Delete parsed bus route from database.
- If parsed bus route is "Add"
 - Add new bus route with parsed bus route's data.
 - Approve all related parsed bus path (if any).
 - Delete parsed bus route from database.

7.5.2.3 Update one bus route in mobile application

- Member can only update a whole bus route (not bus station nor bus path separately). When member chooses to update one bus route, application will download all bus route data in server (include related bus path and bus station), called server bus route, and compare with corresponding bus route data in reference database, called local bus route, to make changes.
- Process to update one bus route
- If last update date of server bus route is larger than local bus route
 - Update all related bus path.
 - Update local bus route's data based on server bus route's data.
- If no local bus route which is corresponding to server bus route (has the same ID attribute)
 - Add new local bus route with server bus route's data.
 - Update all related server bus path.
- Process to update one bus path
- Update start station.
- Update stop station
- If last update date of server bus path is larger than local bus path
 - Update local bus path's data based on server bus path's data.
- If no local bus path which is corresponding to server bus path (has the same NumberInTravel attribute)
 - Add new local bus path with server bus path's data.
- Process to update one bus station
- If last update date of server bus station is larger than local bus station

- Update local bus station's data based on server bus station's data.
- If no local bus station which is corresponding to server bus station (has the same ID attribute)
 - Add new local bus station with server bus station's data.

E. System Implementation & Test

1. Introduction

1.1. Overview

This section provides in details all necessary information about implementation information and testing procedure of TMST includes test plan, test cases, test result and risks estimations.

1.2. Test Approach

- Goal: To test the whole system based on the core workflow.
- Method: Black-box Testing.

2. Database Relationship Diagram

2.1. Web Service Physical Diagram

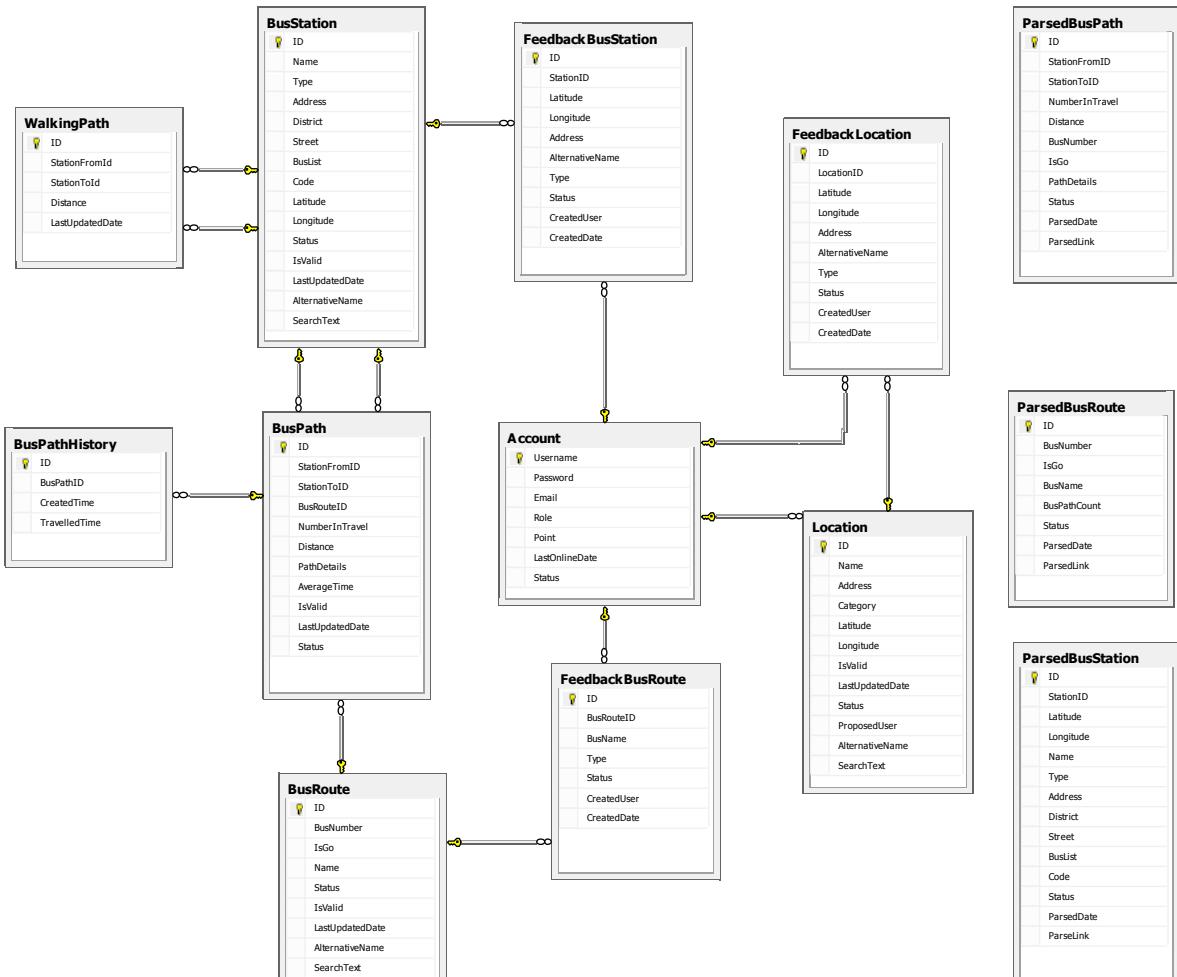


Figure 89 Web Service Physical Diagram

2.2. Web Service Data Dictionary

Entity Data dictionary: describe content of all entities	
Entity Name	Description
BusStation	Describe all bus stations in the system.
BusPath	Describe all bus paths in the system.
BusRoute	Describe all bus routes in the system.
WalkingPath	Describe all walking distances between two nearby bus stations in the system.
BusPathHistory	Describe all actual time need to travel in a bus path in the system.
Account	Describe all accounts in the system.
FeedbackBusStation	Describe all feedbacks for bus stations in the system.
FeedbackBusRoute	Describe all feedbacks for bus routes in the system.
FeedbackLocation	Describe all feedbacks for locations in the system.
Location	Describe all locations in the system.
ParsedBusStation	Describe all parsed bus station which has changes compare to respective bus station in the system.
ParsedBusPath	Describe all parsed bus path which has changes compare to respective bus path in the system.
ParsedBusRoute	Describe all parsed bus route which has changes compare to respective bus route in the system.

Table 15 Web Data Dictionary

Entity name	Attributes	Description	Domain	Null
BusStation	ID {PK}	Unique identifier of each bus station.	varchar(12)	No
	Name	Name of bus station.	nvarchar(80)	No
	Type	Type of bus station.	nvarchar(20)	No
	Address	Address of bus station.	nvarchar(120)	No
	District	District of bus station.	nvarchar(20)	No
	Street	Street of bus station.	nvarchar(80)	No
	BusList	List of buses which are available at bus station.	varchar(200)	No
	Code	Code of bus station.	varchar(20)	No
	Latitude	Latitude of bus station.	float	No
	Longitude	Longitude of bus station.	float	No

	Status	Status of bus station.	varchar(20)	No
	IsValid	Validity of bus station.	bit	No
	LastUpdatedDate	Latest date and time bus station has updated.	datetime	No
	AlternativeName	Alternative name of bus station.	nvarchar(MAX)	No
	SearchText	Text which is used for searching bus station.	nvarchar(MAX)	No
BusPath	ID {PK}	Unique identifier of each bus path.	varchar(15)	No
	StationFromID {FK}	ID of bus station which bus path starts.	varchar(12)	No
	StationToID {FK}	ID of bus station which bus path stops.	varchar(12)	Yes
	NumberInTravel	The order of bus path in bus route.	int	No
	Distance	Distance of bus path in metres.	float	No
	BusRouteID {FK}	ID of bus route which bus path belongs to.	varchar(20)	No
	PathDetails	List of coordinates which are used to define path.	nvarchar(MAX)	No
	AverageTime	Average time which the bus travel on bus path in reality, in second.	int	No
	IsValid	Validity of	bit	No

		bus path.		
	LastUpdatedDate	Latest date and time bus path has updated.	datetime	No
	Status	Status of bus path.	varchar(20)	No
BusRoute	ID {PK}	Unique identifier of each bus route.	varchar(20)	No
	BusNumber	Number of bus route.	varchar(10)	No
	IsGo	Turn of bus route.	bit	No
	Name	Name of bus route.	nvarchar(100)	No
	Status	Status of bus route.	varchar(20)	No
	IsValid	Validity of bus route.	bit	No
	LastUpdatedDate	Latest date and time bus route has updated.	datetime	No
	AlternativeName	Alternative name of bus route.	nvarchar(MAX)	Yes
	SearchText	Text which is used for searching bus route.	nvarchar(MAX)	Yes
WalkingPath	ID {PK}	Unique identifier of each bus station distance.	int	No
	StationFromID {FK}	ID of bus station which starts.	varchar(12)	No
	StationToID {FK}	ID of bus station which stops.	varchar(12)	No
	Distance	Walking distance between 2 stations.	float	No
	LastUpdatedDate	Latest date and time bus	datetime	No

		station distance has updated.		
BusPathHistory	ID {PK}	Unique identifier of each history.	int	No
	BusPathID {FK}	Travelled bus path.	varchar(15)	No
	CreatedTime	Travelled time.	datetime	No
	TravelledTime	Time which consumes to travel on bus path, in second.	int	No
Account	Username {PK}	Username for logging to account.	varchar(20)	No
	Password	Password for logging to account.	varchar(50)	No
	Email	Email of account	Varchar(50)	No
	Role	Role which account is assigned.	varchar(50)	No
	Point	Point of account.	int	No
	LastOnlineDate	Latest date and time account has connected to server.	datetime	No
	Status	Status of account.	varchar(10)	No
FeedbackBusStation	ID {PK}	Unique identifier of each feedback bus station.	int	No
	StationID {FK}	ID of bus station	varchar(12)	No
	Latitude	New latitude of bus station in feedback.	float	Yes
	Longitude	New longitude of bus station	float	Yes

		in feedback.		
	Address	New address of bus station in feedback.	nvarchar(120)	Yes
	AlternativeName	New alternative name of bus station in feedback.	nvarchar(80)	Yes
	Type	Type of bus station	varchar(30)	No
	Status	Status of feedback bus station	varchar(15)	No
	CreatedUser {FK}	User who is created feedback bus station	Varchar(20)	No
	CreatedDate	Date and time when feedback is created.	datetime	No
FeedbackBusRoute	ID {PK}	Unique identifier of each feedback bus route.	int	No
	BusRouteID {FK}	ID of bus route	varchar(20)	No
	BusName	New name of bus route in feedback.	nvarchar(100)	Yes
	AlternativeName	New alternative name of bus route in feedback.	nvarchar(100)	Yes
	Type	Type of bus route	varchar(30)	No
	Status	Status of feedback bus route	varchar(15)	No
	CreatedUser {FK}	User who is created feedback bus route	Varchar(20)	No
	CreatedDate	Date and time when	datetime	No

		feedback is created.		
FeedbackLocation	ID {PK}	Unique identifier of each feedback location.	int	No
	LocationID {FK}	ID of location	int	No
	Longitude	New longitude of location in feedback.	float	Yes
	Latitude	New latitude of location in feedback.	float	Yes
	Address	New address of location in feedback.	nvarchar(120)	Yes
	AlternativeName	New alternative name of location in feedback.	nvarchar(100)	Yes
	Type	Type of location	varchar(30)	No
	Status	Status of feedback location	varchar(15)	No
	CreatedUser {FK}	User who is created feedback location	varchar(20)	No
	CreatedDate	Date and time when feedback is created.	datetime	No
Location	ID {PK}	Unique identifier of each location.	int	No
	Name	Name of location.	nvarchar(100)	No
	Address	Address of location.	nvarchar(120)	No
	Category	Category of location.	nvarchar(20)	No
	Latitude	Latitude of location.	float	No

	Longitude	Longitude of location	float	No
	IsValid	Validity of location.	bit	No
	LastUpdatedDate	Latest date and time location has updated.	datetime	No
	Status	Status of location.	varchar(20)	No
	ProposedUser {FK}	User who propose the location	varchar(20)	Yes
	AlternativeName	Alternative name of location.	nvarchar(MAX)	Yes
	SearchText	Text which is used for searching location.	nvarchar(MAX)	No
ParsedBusStation	ID {PK}	Unique identifier of each parsed bus station.	int	No
	StationID	Parsed ID of parsed bus station	nvarchar(10)	No
	Latitude	Parsed latitude of parsed bus station	float	No
	Longitude	Parsed longitude of parsed bus station	float	No
	Name	Parsed name of parsed bus station	nvarchar(80)	Yes
	Type	Parsed type of parsed bus station	nvarchar(20)	Yes
	Address	Parsed address of parsed bus station	nvarchar(120)	Yes
	District	Parsed district of parsed bus station	nvarchar(20)	Yes

	Street	Parsed street of parsed bus station	nvarchar(80)	Yes
	Buslist	Parsed list which is available at station.	varchar(200)	Yes
	Code	Parsed code of parsed bus station	varchar(20)	Yes
	Status	Status of parsed bus station	varchar(20)	No
	ParsedDate	Latest date and time bus path has parsed.	datetime	No
	ParsedLink	Link which is parsed bus path.	varchar(100)	No
ParsedBusPath	ID {PK}	Unique identifier of each parsed bus path.	int	No
	StationFromID	Parsed ID of bus station which parsed bus path begins.	varchar(12)	No
	StationToID	Parsed ID of bus station which parsed bus path ends.	varchar(12)	No
	NumberInTravel	The parsed order of parsed bus path in bus route.	int	No
	Distance	Parsed Distance of parsed bus path in metres.	float	No
	BusNumber	Parsed number of bus route.	varchar(50)	No
	IsGo	Parsed turn of bus route.	bit	No

	PathDetails	Parsed list of coordinate which is used to define path.	varchar(MAX)	Yes
	Status	Status of parsed bus path.	varchar(20)	No
	ParsedDate	Latest date and time bus path has parsed.	datetime	No
	ParsedLink	Link which is parsed bus path.	varchar(100)	No
ParsedBusRoute	ID {PK}	Unique identifier of each parsed bus route.	int	No
	BusNumber	Parsed number of parsed bus route.	varchar(10)	No
	IsGo	Parsed turn of bus route.	bit	No
	BusName	Parsed name of parsed bus route.	nvarchar(100)	Yes
	BusPathCount	Number of parsed bus path which is belonged to parsed bus route.	int	Yes
	Status	Status of parsed bus route.	varchar(20)	No
	ParsedDate	Latest date and time bus route has parsed.	datetime	No
	ParsedLink	Link which is parsed bus route.	varchar(100)	No

Table 16 Web Attribute Data Dictionary

2.3. Mobile Application Physical Diagram

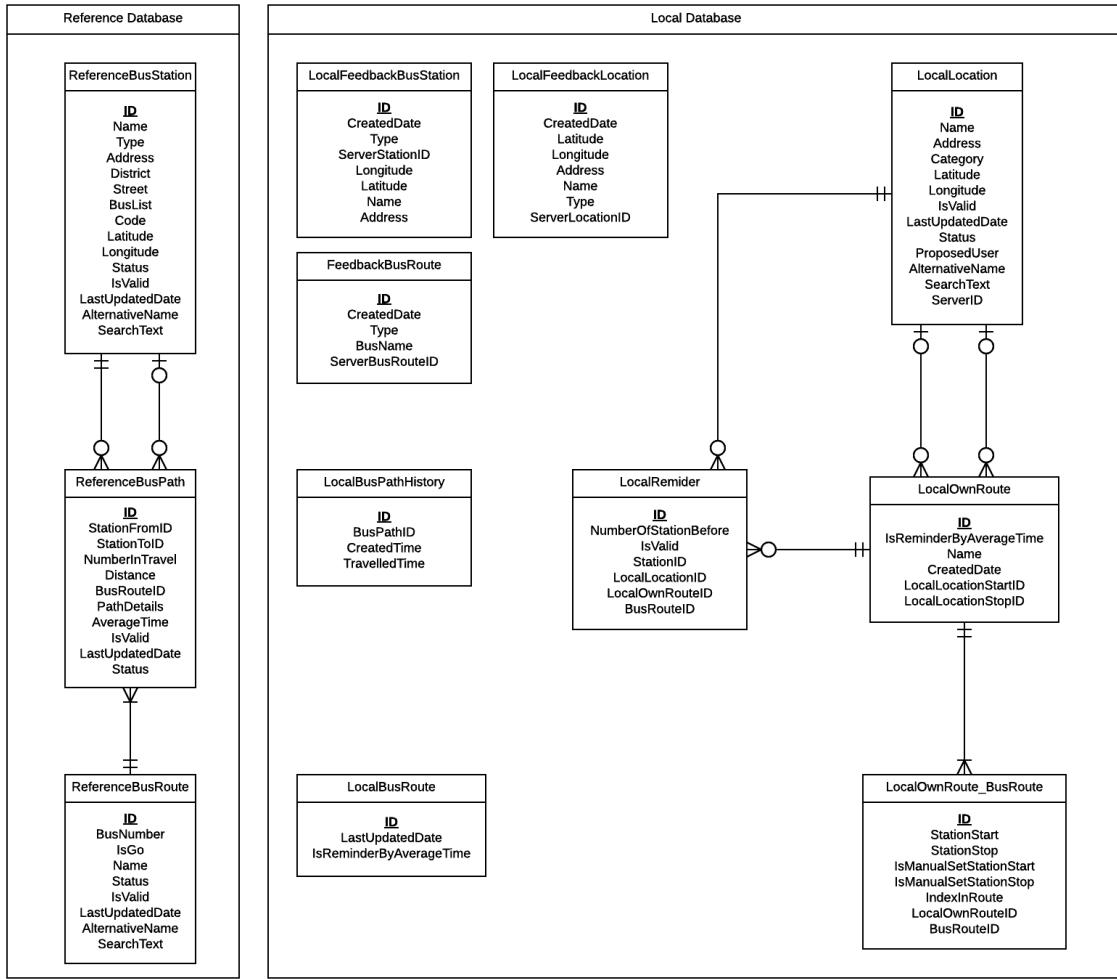


Figure 90 Mobile Application Physical Database

2.4. Mobile Application Data Dictionary

Entity Data dictionary: describe content of all entities	
Entity Name	Description
ReferenceBusStation	Describe all bus stations in the system.
ReferenceBusPath	Describe all bus paths in the system.
ReferenceBusRoute	Describe all bus routes in the system.
LocalFeedbackBusStation	Describe all unsent feedbacks for bus stations of member in the system.
LocalFeedbackBusRoute	Describe all unsent feedbacks for bus routes of member in the system.
LocalFeedbackLocation	Describe all unsent feedbacks for locations of member in the system.
LocalBusPathHistory	Describe all actual time need to travel in a bus path in the system, which member has saved while travelling.
LocalReminder	Describe all reminders of member in the system.
LocalLocation	Describe all locations of member in the system.
LocalOwnRoute	Describe all own route of member in the system.
LocalOwnRoute_BusRoute	Describe all bus routes included in own routes in the

	system.
LocalBusRoute	Describe all saved bus route of member in the system.

Table 17 Mobile Application Data Dictionary

Entity name	Attributes	Description	Domain	Null
ReferenceBusStation	ID {PK}	Unique identifier of each bus station.	varchar(12)	No
	Name	Name of bus station.	nvarchar(80)	No
	Type	Type of bus station.	nvarchar(20)	No
	Address	Address of bus station.	nvarchar(120)	No
	District	District of bus station.	nvarchar(20)	No
	Street	Street of bus station.	nvarchar(80)	No
	BusList	List of buses which are available at bus station.	varchar(200)	No
	Code	Code of bus station.	varchar(20)	No
	Latitude	Latitude of bus station.	float	No
	Longitude	Longitude of bus station.	float	No
	Status	Status of bus station.	varchar(20)	No
	IsValid	Validity of bus station.	bit	No
ReferenceBusPath	LastUpdatedDate	Latest date and time bus station has updated.	datetime	No
	AlternativeName	Alternative name of bus station.	nvarchar(MAX)	No
	SearchText	Text which is used for searching bus station.	nvarchar(MAX)	No
ReferenceBusPath	ID {PK}	Unique identifier of each bus path.	varchar(15)	No
	StationFromID {FK}	ID of bus	varchar(12)	No

		station which bus path starts.		
	StationToID {FK}	ID of bus station which bus path stops.	varchar(12)	Yes
	NumberInTravel	The order of bus path in bus route.	int	No
	Distance	Distance of bus path in metres.	float	No
	BusRouteID {FK}	ID of bus route which bus path belongs to.	varchar(20)	No
	PathDetails	List of coordinates which are used to define path.	varchar(MAX)	No
	AverageTime	Average time which the bus travel on bus path in reality, in second.	int	No
	IsValid	Validity of bus path.	bit	No
	LastUpdatedDate	Latest date and time bus path has updated.	datetime	No
	Status	Status of bus path.	varchar(20)	No
ReferenceBusRoute	ID {PK}	Unique identifier of each bus route.	varchar(20)	No
	BusNumber	Number of bus route.	varchar(10)	No
	IsGo	Turn of bus route.	bit	No
	Name	Name of bus route.	nvarchar(100)	No
	Status	Status of bus route.	varchar(20)	No
	IsValid	Validity of	bit	No

		bus route.		
	LastUpdatedDate	Latest date and time bus route has updated.	datetime	No
	AlternativeName	Alternative name of bus route.	nvarchar(MAX)	Yes
	SearchText	Text which is used for searching bus route.	nvarchar(MAX)	Yes
LocalFeedbackBusStation	ID {PK}	Unique identifier of each saved feedback bus station	int	No
	CreatedDate	Date of feedback bus station	datetime	No
	Type	Type of feedback bus station	varchar(30)	No
	ServerStationID	ID of bus station in server.	int	No
	Longitude	New longitude of bus station in feedback.	float	Yes
	Latitude	New latitude of bus station in feedback.	float	Yes
	Name	New name of bus station in feedback.	nvarchar(80)	Yes
	Address	New address of bus station in feedback.	nvarchar(120)	Yes
LocalFeedbackLocation	ID {PK}	Unique identifier of each saved feedback location.	int	No
	CreatedDate	Date of feedback location	datetime	No

	Latitude	New latitude of location in feedback.	float	Yes
	Longitude	New longitude of location in feedback.	float	Yes
	Address	New address of location in feedback.	nvarchar(120)	Yes
	Name	New name of location in feedback.	nvarchar(100)	Yes
	Type	Type of feedback location	varchar(30)	No
	ServerLocationID	ID of location in server.	int	No
LocalFeedbackBusRoute	ID {PK}	Unique identifier of each saved feedback bus route	int	No
	CreatedDate	Date of feedback bus route	datetime	No
	Type	Type of feedback bus route	varchar(30)	No
	BusName	New name of bus route in feedback	nvarchar(100)	No
	ServerBusRouteID	ID of bus route in server.	varchar(20)	No
LocalBusPathHistory	ID {PK}	Unique identifier of each travel history	int	No
	BusPathID	Travelled bus path.	varchar(15)	No
	CreatedTime	Travelled time.	datetime	No
	TravelledTime	Time which consumes to travel on bus path, in second.	int	No

LocalBusRoute	ID {PK}	Unique identifier of each bus route.	varchar(20)	No
	LastUpdatedDate	Last date local bus route is updated.	datetime	No
	IsReminderByAverageTime	Check if bus route has "reminder by average time" enabled or not.	bit	No
LocalReminder	ID {PK}	Unique identifier of each reminder.	int	No
	NumberOfStationBefore	Number of station which reminder is alerted before the expected station comes.	int	No
	IsValid	Reminder's validity.	bit	No
	StationID	Reminded bus station.	varchar(12)	Yes
	LocalLocationID {FK}	Reminded bus station.	int	Yes
	LocalOwnRouteID {FK}	Reminded own route.	int	Yes
	BusRouteID	Reminded bus route.	varchar(20)	Yes
LocalLocation	ID {PK}	Unique identifier of each location.	int	No
	Name	Name of location.	nvarchar(100)	No
	Address	Address of location.	nvarchar(120)	No
	Category	Category of location.	nvarchar(20)	No
	Latitude	Latitude of	float	No

		location.		
	Longitude	Longitude of location	float	No
	IsValid	Validity of location.	bit	No
	LastUpdatedDate	Latest date and time location has updated.	datetime	No
	Status	Status of location.	varchar(20)	No
	ProposedUser	User who propose the location	varchar(20)	Yes
	AlternativeName	Alternative name of location.	nvarchar(MAX)	Yes
	SearchText	Text which is used for searching location.	nvarchar(MAX)	No
	ServerID	ID of location in server.	int	Yes
LocalOwnRoute	ID {PK}	Unique identifier of each own route	int	No
	IsReminderByAverageTime	Check if bus route has “reminder by average time” enabled or not.	bit	No
	Name	Name of own route	varchar(100)	No
	CreatedDate	Created date of own route	datetime	No
	LocalLocationStartID {FK}	Location where own route starts.	int	Yes
	LocalLocationStopID {FK}	Location where own route stops.	int	Yes
LocalOwnRoute_Bus Route	ID {PK}	Unique identifier of each bus route in each	int	No

		own route.		
	StationStart	Index of station where bus route starts.	int	No
	StationStop	Index of station where bus route stops.	int	No
	IsManualSetStation Start	Check if start station is set manually or automatically.	bit	No
	IsManualSetStation Stop	Check if stop station is set manually or automatically.	bit	No
	IndexInRoute	Index of bus route in own route.	int	No
	LocalOwnRouteID	Own route.	int	No
	BusRouteID	Bus route.	varchar(20)	No

Table 18 Mobile Application Attribute Data Dictionary

3. Performance Measures

3.1. Find next station performance

- In general, find next station function and related functions such as save history and set reminder run fine when GPS is changed per 3 seconds (tested on emulator and real device).

3.2. Search route performance

- In general, the speed of search route algorithm should less than 1 minute.
- The purpose of this project is not about search route so it is accepted with above result.

4. Test Plan

- The purpose of this section is to verify and ensure that TMST meets its design specification and other requirements from user. The following part will describe which features to be tested and which is not.
- Test plan will only focus on mobile application with functions which are frequently used by member: View route flow, Create route flow, Search route flow.

4.1. Features to be tested

- Member: View route, Add reminder, Set reminder, Find next station, Create own route, Search route, Search location.

4.2. Features not to be tested

- Guest: Login, Register.
- Member: Logout, Change password, View route list, Feedback, History, Manage reminder, Update data, Edit own route, Remove own route.
- Staff: Set parse time, Manage bus route, Manage bus path, Manage bus station, Manage location, Manage account.

5. System Testing Test Case

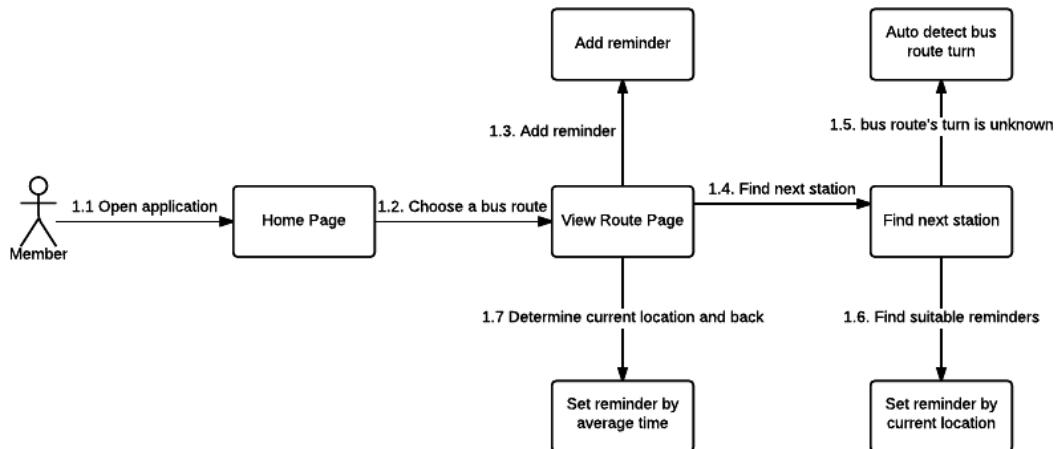


Figure 91 View Route Flow

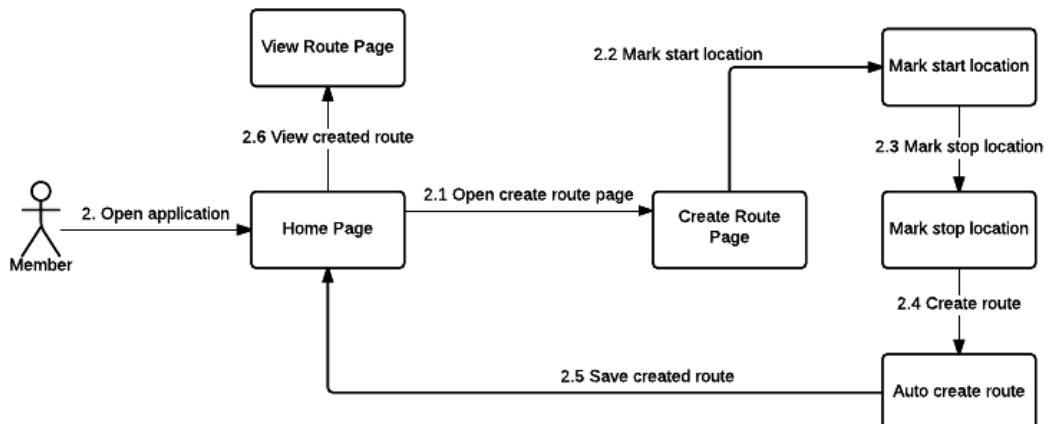


Figure 92 Create Route Flow

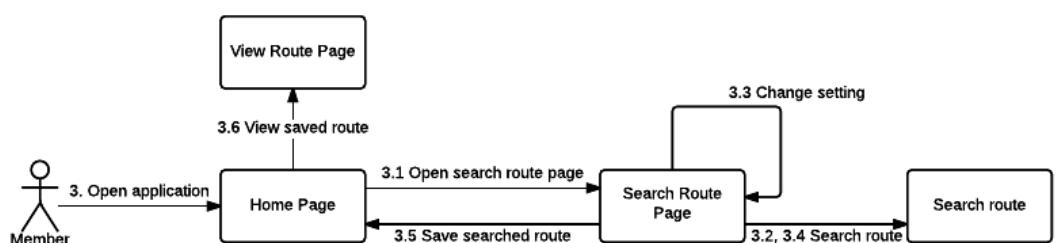


Figure 93 Search Route Flow

5.1. View Route Flow

ID	Test Case Description	Test Case Procedure	Expected Output	Inter-test Case Dependency	Result	Test Date	Note
VRL1	<ul style="list-style-type: none"> - Member opens application, system will check account and policy and data is loaded. - Member has saved buses: 27, 18 and created route: "di hoc". 	1 – Click on application icon.	After step 1, application starts and check account and policy. List of route is loaded by order: "di hoc" -> 18 -> 27 -> other buses.		Passed	28/03/2015	
VRL2	Member opens application, system will check account and policy with expired result.	1 – Click on application icon.	After step 1, application starts and check account and policy. And show message "Thời hạn sử dụng offline đã kết thúc. Xin vui lòng đăng nhập để tiếp tục sử dụng chương trình."		Passed	28/03/2015	
VR1	<ul style="list-style-type: none"> - Member views one bus route. - Member has one set reminder at 	1 – Choose one bus number 27.	After step 1, shows menu with 3 turns includes "Lượt đi - Hướng		Passed	28/03/2015	

	<p>“Trạm hoa kiếng Thành Hưng” station in bus number 27 with turn “Lượt đi”.</p>	<p>2 - Choose turn “Lượt đi - Hướng về phía An Sương”.</p>	<p>về phía An Sương”, “Lượt về - Hướng về phía Bến Thành”, “Tự động chỉnh theo hướng xe đi.</p> <p>After step 2, data of bus route is loaded: bus paths in form of polylines and bus stations in form of markers.</p> <p>“Trạm hoa kiếng Thành Hưng” station's marker is changed to orange color.</p>				
VR2	<ul style="list-style-type: none"> - Member views one own route. - Member has one own route “di hoc” with 3 bus routes 27, 32, 55. 	<p>1 - Choose “di hoc” in route list.</p> <p>2 - Choose “xem đường” option.</p>	<p>After step 1, shows menu with 3 options: “Xem đường”, “Chỉnh sửa”, “Xóa đường”.</p> <p>After step 2, data of own route is loaded: bus paths in form of polylines and bus</p>		Passed	28/03/2015	

			stations in form of markers. Each bus route's color is different. "Khu công nghiệp Tân Bình" station of bus route 27, "Đường Nguyễn Duy Cung" station of bus route 32, and "CVPM Quang Trung" station of bus route 55's markers change to orange color.				
AR1	Member adds reminder at one station.	1 - Hold one station's marker. 2 - Choose "thêm nhắc nhở" option. 3 - Choose "Báo trước" and "Đặt lúc" option and click "Thêm" button.	After step 1, station menu is displayed which include "thêm nhắc nhở" option. After step 2, Add reminder dialog is displayed. After step 3, Data of reminder is added to local database. Station's	VR1	Passed	28/03/2015	

			marker's color is changed to orange.				
AR2	Member adds reminder at one location near route.	1 - Hold one location's marker. 2 - Choose "thêm nhắc nhở" option. 3 - Choose "Báo trước" and "Đặt lúc" option and click "Thêm" button.	After step 1, location menu is displayed which include "thêm nhắc nhở" option. After step 2, add reminder dialog is displayed. After step 3, station which has the shortest walking route to chosen location's marker's color is changed to orange. Data of reminder is added to local database. Data of location is added to local database.	VR1	Passed	28/03/2015	
FNS1	Member find next station by GPS.	1 - Each time device's GPS is changed.	After step 1, current location is displayed at GPS' location data in form of	VR1	Passed	28/03/2015	

			marker. Next station's marker 'style is changed.				
FSRL	Find and set suitable reminders based on location.	1 - Each time next station is found.	After step 1, reminders which are set at next station is registered to device. Show message for each registered reminder.	VR1	Passed	28/03/2015	
FSRA	Find and set suitable reminders based on average time.	1 - Current location is determined. Click back button.	After step 1, reminders which haven't set is set which registered time calculated by route distance from current location to reminder's set station and speed of 30km/8 (8.333m/s).	VR1	Passed	28/03/2015	

5.2. Create Route Flow

ID	Test Case	Test Case	Expected	Inter-test	Result	Test Date	Note
----	-----------	-----------	----------	------------	--------	-----------	------

	Description	Procedure	Output	Case Dependency			
CRABR1	Member add first bus route in create route page.	1 - Member choose bus route “27 - Lượt đi, hướng về phía An Sương” from list. 2 - Choose “thêm tuyến này” option.	After step 1, bus route menu is displayed which include “thêm tuyến này” option. After step 2, bus route data is loaded in map. First station “Ga xe buýt Sài Gòn” is marked as “Trạm lên”, last station “Bến xe An Sương” is marked as “Trạm xuống”. Bus route item in list’s style is changed.	CR	Passed	28/03/2015	
CRABR2	Member add next bus route in create route page.	1 - Member choose bus route “32 - Lượt về, hướng về phía Bến xe Miền Tây” from list.	After step 1, bus route menu is displayed which include “thêm tuyến này” option.	CR CRABR1	Passed	28/03/2015	Use result of CRABR1 as initial data.

		2 - Choose “thêm tuyến này” option.	After step 2, bus route data is loaded in map with different color. Then “Khu công nghiệp Tân Bình” station of previous bus route 27 is chosen as “Trạm xuống” of bus route 27, “Trạm dệt Thành Công” station is chosen as “Trạm lên” of bus route 32. Bus route item in list's style is changed.				
CRABR3	Member add next bus route in create route page, make route becomes invalid.	1 - Member choose bus route “55 - Lượt về, hướng về phía CVPM Quang Trung” from list. 2 - Choose “thêm	After step 1, bus route menu is displayed which include “thêm tuyến này” option. After step 2,	CR CRABR1 CRABR2	Passed	28/03/2015	Use result of CRABR2 as initial data.

		tuyến này" option.	show message "Xe 32 lượt về không thể đổi sang xe 55 lượt về. Xin vui lòng đổi lượt của xe hoặc tùy chỉnh lại trạm bằng tay".				
CRSR	Member save created route.	1 - Click "lưu lại" button. 2 - Input name and click save button.	1 - Show name textbox and save button. 2 - Data is saved to local database. Navigate back to main page with created route is shown at first in route list.	CR	Passed	28/03/2015	Use result of CRABR3 as initial data.

5.3. Search Route Flow

ID	Test Case	Test Case	Expected Output	Inter-	Result	Test Date	Note
----	-----------	-----------	-----------------	--------	--------	-----------	------

	Description	Procedure		test Case Dependency			
SR2	Member search route.	1 - Choose start location at "900 Âu Cơ" and stop location at "Suối Tiên". Click "tìm đường" button.	After step 1, show results: 3 routes - 1 buses: "30 Lượt đi". - 2 buses: "30 Lượt đi" and "12 Lượt đi". - 2 buses: "30 Lượt đi" and "30 Lượt về".		Passed	28/03/2015	- Mode: least bus - Maximum walking : 0m
SR3	Member search route.	1 - Choose start location at "900 Âu Cơ" and stop location at "Suối Tiên". Click "tìm đường" button.	After step 1, show results: 25 routes.		Passed	28/03/2015	- Mode: least walking - Maximum bus: 2.
SRSR	Member save searched route.	1 - Click "lưu lại" button. 2 - Input name and click save button.	1 - Show name textbox and save button. 2 - Data is saved to local database. Navigate back to main page with saved route is shown at first in route list.		Passed	28/03/2015	Use any valid result from any SR test case.

F. Software User's Manual

1. Installation Guide

1.1. Setting up environment at server side

The following software must be installed into the server machine:

1.1.1. Hardware Requirements

Personal computer for developing with the minimum configuration:

- CPU Intel® Core™ 2.10GHz
- 2GB of RAM
- 5GB of hard disk
- Wi-Fi Internet (> 2Mbps)

1.1.2. Software Requirements

- .Net Framework 4.5.
- Web Server: Internet Information System - IIS 7.5 with MVC5 enable.
- Microsoft Windows Server 2008 R2 Enterprise.
- Create and manager: SQL Server 2008 R2.

1.1.3. Mobile Requirement

- Windows Phone 8.0 or later.

1.2. Deployment at server

1.2.1. Prepare deployment packages

- Create a folder to contains the deployment packages, for example: D:\NextBusStation-Deploy
- Copy the Web Service deployment package (about 1.75MB) to a folder, for example: D:\NextBusStation-Deploy\WebService
- Copy the Web Application deployment package (about 37.7MB) to a folder, for example: D:\NextBusStation-Deploy\WebApplication

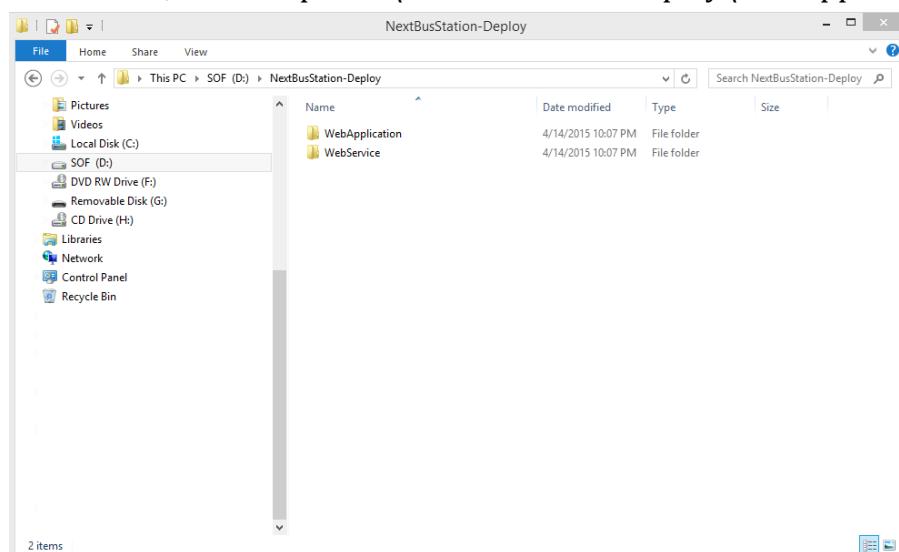


Figure 94 Prepare deployment package folders

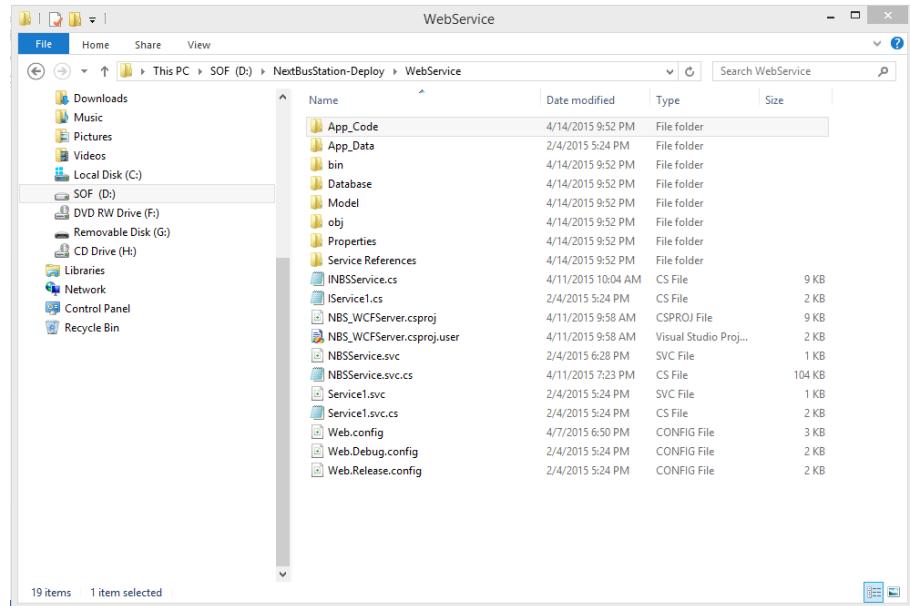


Figure 95 Prepare deployment package for web service

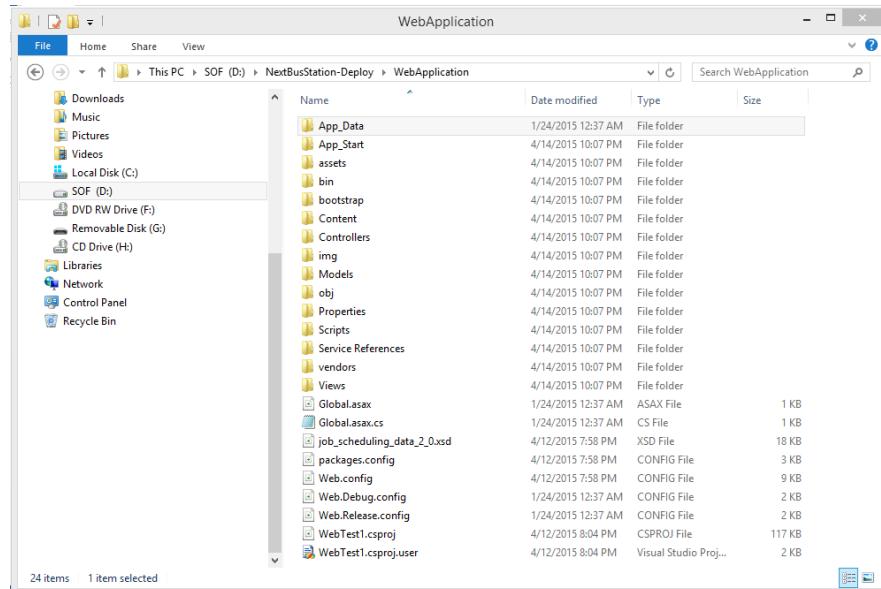


Figure 96 Prepare deployment package for web application

1.2.2. Deploy database

- Open Microsoft SQL Server Management Studio.
- Server name: the host server IPv4 address or SQL instance name of the machine.
- Login: account have DDL and DML authority.
- Password: password of the account.

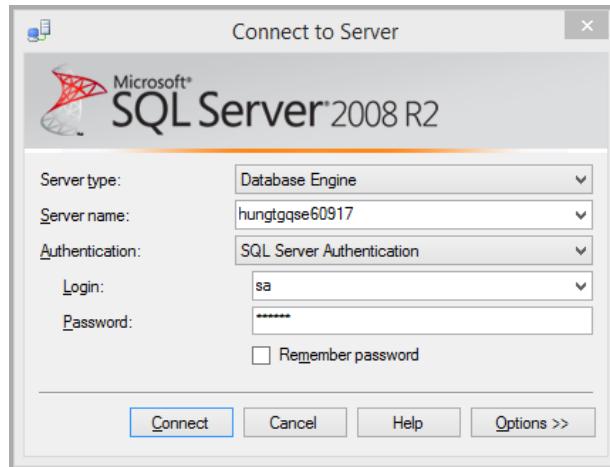


Figure 97 Connect database

- Open NextBusStationDBscript.sql file
- Go to line 5 and 7, set the correct path of SQL Management setup.

```

4: DATABASE [NextBusStation] ON PRIMARY
5: = N'NextBusStation', FILENAME = N'C:\Program Files\Microsoft SQL Server\MSSQL10_50.MSSQLSERVER\MSSQL\DATA\NextBusStation.mdf', SIZE
6: = 1, MAXSIZE = 1024, FILEGROWTH = 10
7: = N'NextBusStation_log', FILENAME = N'C:\Program Files\Microsoft SQL Server\MSSQL10_50.MSSQLSERVER\MSSQL\DATA\NextBusStation_log.ldf'
8:

```

- Execute script file.
- Check the existed of NextBusStation database (13 tables) to complete.

1.2.3. Deploy web service on server

- Open the Internet Information System (IIS) Manager.
- Change the connection string in D:\NextBusStation-Deploy\WebService\Web.config.
- Right click at Sites. Choose "Add Web Site".

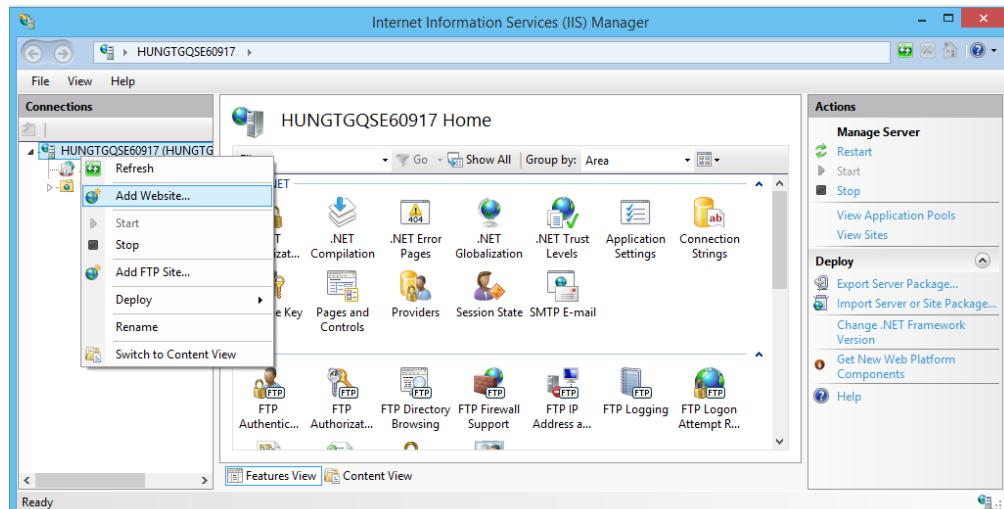


Figure 98 IIS Control Panel

- Enter site name. Choose Application Pool as .NET v4.5
- Select the path as the folder D:\NextBusStation-Deploy\WebService
- Change port if needed.

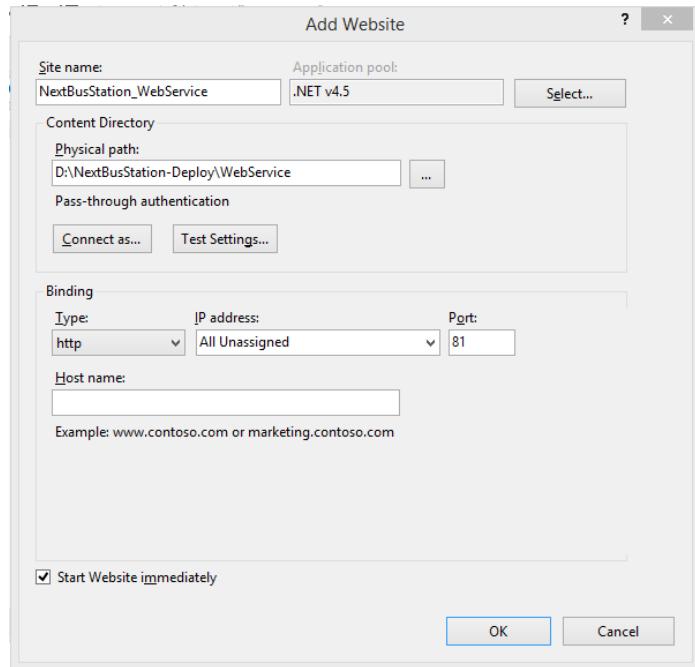


Figure 99 Add web service in IIS

- Press ok to finish the process.
- Test webservice by clicking "Browse *:81 (http)"

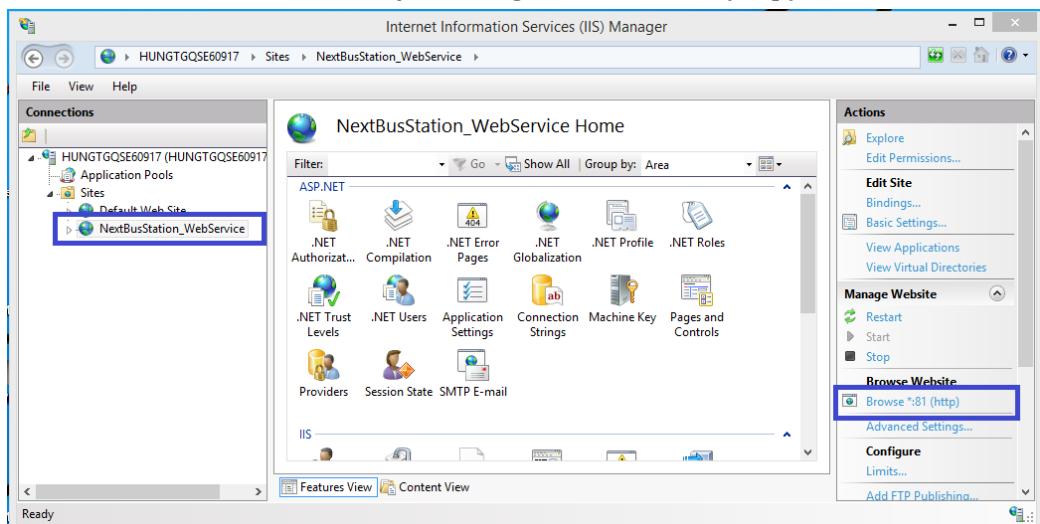


Figure 100 Test deployment

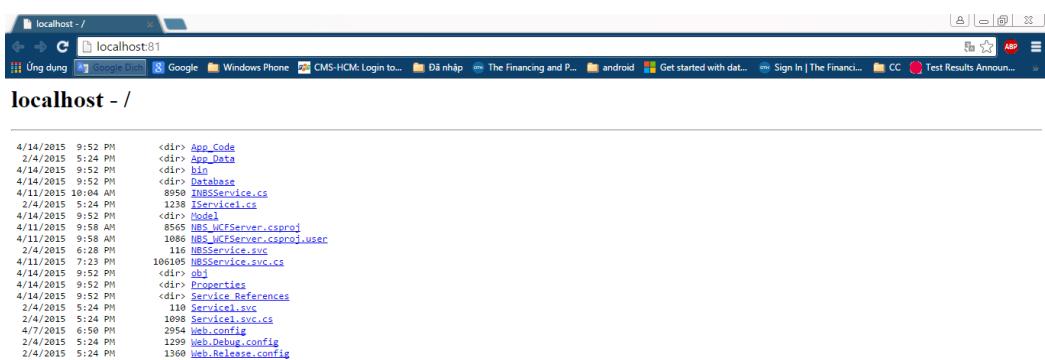


Figure 101 Deployment result

1.2.4. Deploy web application on server

- Open the Internet Information System (IIS) Manager.
- Change the connection string in D:\NextBusStation-Deploy\WebApplication\Web.config.
- Right click at Sites. Choose “Add Web Site”.
- Enter site name. Choose Application Pool as .NET v4.5
- Select the path as the folder D:\NextBusStation-Deploy\WebApplication
- Change port if needed.

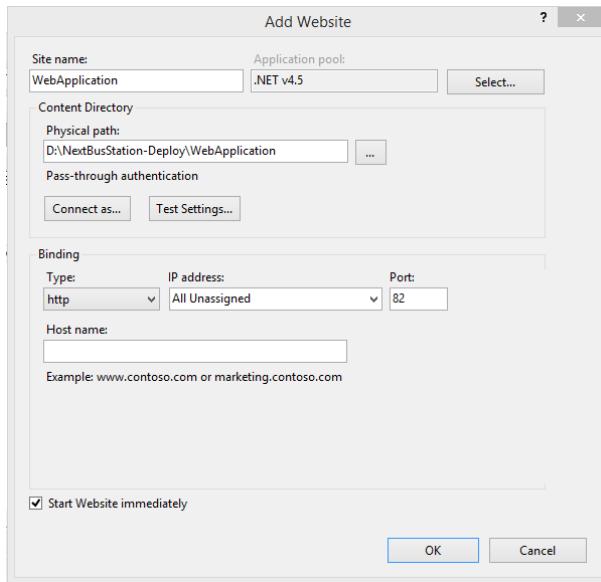


Figure 102 Add Web Site on IIS

- Press ok to finish the process.
- Test webservice by connecting to link
<http://localhost:82/Location/LocationPage>

1.3. Setting up environment at client side

1.3.1. Setting up for computer

- The client devices need to have one the following browsers to access the website:
 - Google Chrome.
 - Firefox.

1.3.2. Setting up for Windows Phone device

- Install application at address <https://www.windowsphone.com/en-us/store/app/next-bus-station/f6137c7c-5aad-4dc4-a14e-43600fa38793> (by downloading .xap file in computer's browser or installing directly in device's browser).

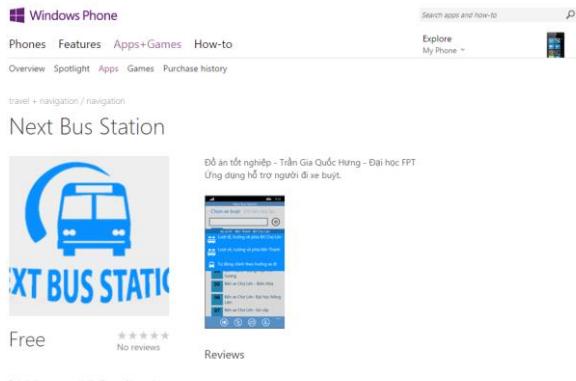


Figure 103 Next Bus Station on Windows Phone store website



Figure 104 Next Bus Station on Windows Phone store mobile

2. User Guide

2.1. Main page



Figure 105 Main page

Step	Description
1	Member touches on search textbox to input information for search.
2	Member touches on button to clear text in search textbox.
3	Member touches on button to enable selecting multiple route mode.
4	Member touches on button to start search route.
5	Member touches on button to select multiple route to view on map.
6	Member touches on button to start create route.

Table 19 Main page

2.1.1. View route page

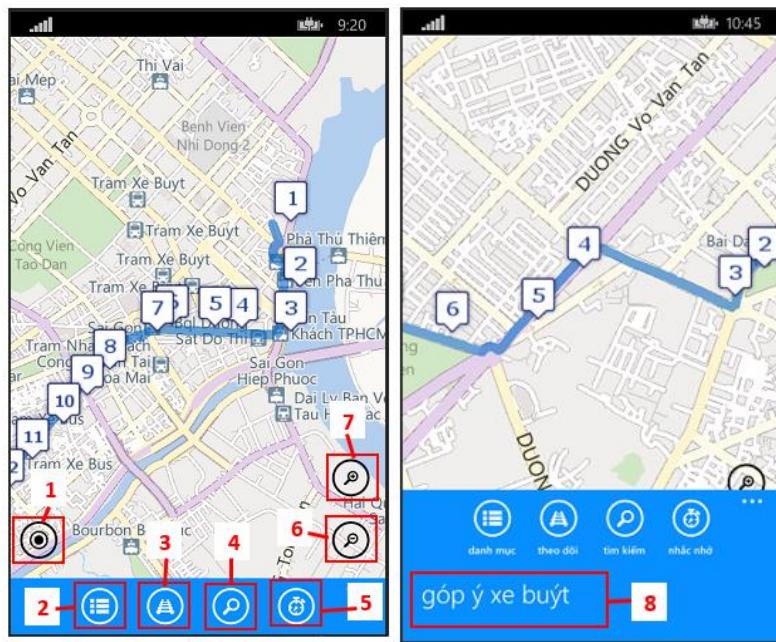


Figure 106 View route page

Step	Description
1	Member touches on button to mark a marker at current location.
2	Member touches on button to show menu.
3	Member touches on button to start tracking.
4	Member touches on button to search information.
5	Member touches on button to open reminder page.
6	Member touches on button to zoom out map.
7	Member touches on button to zoom in map.
8	Member touches on "góp ý xe buýt" link to go to feedback bus route page.

Table 20 View route page

2.2. Create route page



Figure 107 Create route page

Step	Description
1	Member touches on button to mark a marker at current location.
2	Member touches on button to show menu.
3	Member touches on button to create route.
4	Member touches on button to search information.
5	Member touches on button to save bus route.
6	Member touches on button to zoom out map.
7	Member touches on button to zoom in map.

Table 21 Create route page

2.2.1. Edit route menu

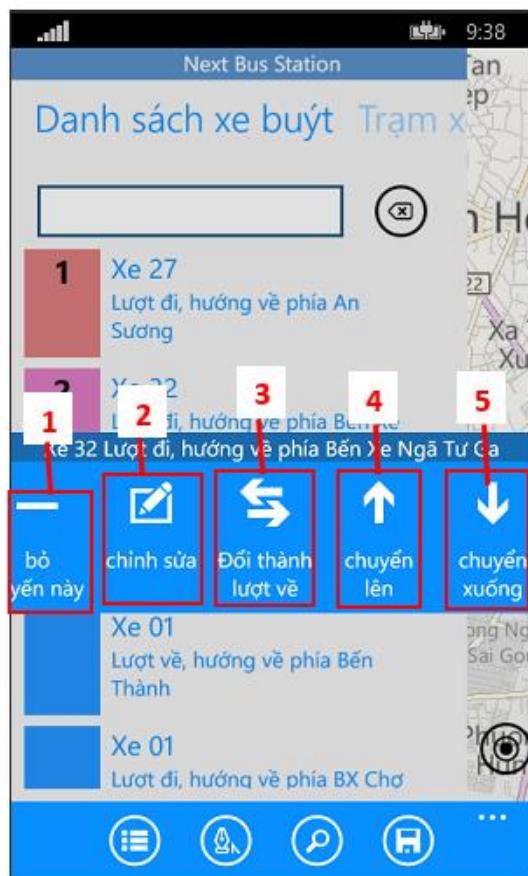


Figure 108 Edit route menu

Step	Description
1	Member touches on “bỏ tuyến này” button to remove chosen bus route from creating route.
2	Member touches on “chỉnh sửa” button to edit chosen bus route from creating route.
3	Member touches on “đổi thành lượt về” button to change turn of chosen bus route from creating route.
4	Member touches on “chuyển lên” button to move up chosen bus route in creating route.
5	Member touches on “chuyển xuống” button to move down chosen bus route in creating route.

Table 22 Edit route menu

2.2.2. Save route

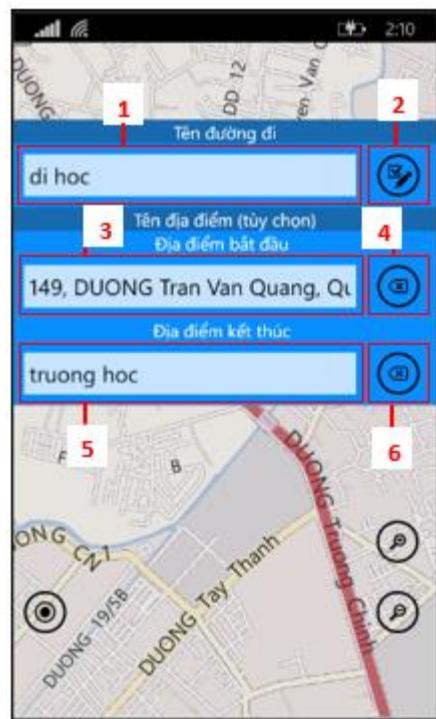


Figure 109 Save created route

Step	Description
1	Member touches on “tên đường đi” textbox to enter name of route.
2	Member touches on button to save route.
3	Member touches on “Địa điểm bắt đầu” textbox to enter name of start location.
4	Member touches on button to clear “Địa điểm bắt đầu” textbox.
5	Member touches on “Địa điểm kết thúc” textbox to enter name of start location.
6	Member touches on button to clear “Địa điểm kết thúc” textbox.

Table 23 Save created route

2.3. Search route page



Figure 110 Search route page

Step	Description
1	Member touches on search textbox to input information for search.
2	Member touches on button to clear text in search textbox.
3	Member touches on button to mark a marker at current location.
4	Member touches on button to show menu.
5	Member touches on button to start search route.
6	Member touches on button to save bus route.
7	Member touches on button to zoom out map.
8	Member touches on button to zoom in map.

Table 24 Search route page

2.3.1. Location menu



Figure 111 Location menu

Step	Description
1	Member touches on “bắt đầu” button to choose start location of creating route.
2	Member touches on “kết thúc” button to choose end location of creating route.

Table 25 Location menu

G. Appendix

1. Microsoft design pattern principles - Microsoft
<https://msdn.microsoft.com/en-us/library/windows/apps/hh781237.aspx>
2. UX/UI Guidelines for Windows Phone 8
<http://blogs.msdn.com/b/africaapps/archive/2014/03/08/ux-guidelines-for-windows-phone-8.aspx>
3. Local database for Windows Phone 8 - Microsoft
<https://msdn.microsoft.com/en-us/library/windows/apps/hh202860%28v=vs.105%29.aspx>
4. Alarms and Reminders for Windows Phone 8 Microsoft
[https://msdn.microsoft.com/en-us/library/windows/apps/hh202946\(v=vs.105\).aspx](https://msdn.microsoft.com/en-us/library/windows/apps/hh202946(v=vs.105).aspx)
5. How to continuously track the phone's location for Windows Phone 8 - Microsoft [https://msdn.microsoft.com/en-us/library/windows/apps/jj247548\(v=vs.105\).aspx](https://msdn.microsoft.com/en-us/library/windows/apps/jj247548(v=vs.105).aspx)
6. Tool for get request - <http://www.telerik.com/fiddler>