|  |  |
| --- | --- |
| **Logo_FPT_University_doc** | **MINISTRY OF EDUCATION AND TRAINING** |

**FPT UNIVERSITY**

|  |
| --- |
| Capstone Project Document |
| Bus Number and Route Suggestion and Management |

|  |  |
| --- | --- |
| **Group 1** | |
| **Group Members** | Nguyễn Phan Quang Nhật – Team leader – 60143  Nguyễn Viết Vĩnh – Team member – 00330  Đoàn Xuân Quang – Team member – 60448  Bùi Tường Thi – Team member – 00721 |
| **Supervisor** | Mr. Kiều Trọng Khánh |
| **Ext Supervisor** | N/A |
| **Capstone Project code** | BUSSUG |

Ho Chi Minh City, 04/2013

# Table of Contents

[Table of Contents 2](#_Toc349677212)

[List of Figures 5](#_Toc349677213)

[List of Tables 6](#_Toc349677214)

[A. Introduction 7](#_Toc349677215)

[B. Project Management Plan 10](#_Toc349677216)

[I. Problem Definition 10](#_Toc349677217)

[1. Capstone Project Name 10](#_Toc349677218)

[2. Problem Abstract 10](#_Toc349677219)

[3. Project Overview 11](#_Toc349677220)

[3.1 The Current Systems 11](#_Toc349677221)

[3.1.a Government hotline 11](#_Toc349677222)

[3.1.b Government public website 11](#_Toc349677223)

[3.1.c Google maps service 11](#_Toc349677224)

[3.2 The Proposed System 11](#_Toc349677225)

[3.3 Boundaries of the System 12](#_Toc349677226)

[3.4 Development Environment 12](#_Toc349677227)

[3.4.a Hardware Requirements 12](#_Toc349677228)

[3.4.b Software Requirements 12](#_Toc349677229)

[II. Project Organization 13](#_Toc349677230)

[1. Software Process Model 13](#_Toc349677231)

[2. Roles and Responsibilities 13](#_Toc349677232)

[3. Tools and Techniques 14](#_Toc349677233)

[III. Project Management Plan 15](#_Toc349677234)

[1. Project Iterations 15](#_Toc349677235)

[2. Iteration Detail 17](#_Toc349677236)

[2.1 Phase 1: Preliminary Investigation or Analysis 17](#_Toc349677237)

[2.2 Phase 2: Data Synchronizer 17](#_Toc349677238)

[2.3 Phase 3: Bus Routes & Bus Stops Management 18](#_Toc349677239)

[2.4 Phase 4: Direction Suggestion 18](#_Toc349677240)

[2.5 Phase 5: Time Schedule 19](#_Toc349677241)

[2.6 Phase 6:Authentication and Authorization 19](#_Toc349677242)

[2.7 Phase 7: Android Client 20](#_Toc349677243)

[3. Meeting minutes 20](#_Toc349677244)

[IV. Coding Convention 20](#_Toc349677245)

[C. Software Requirement Specification 20](#_Toc349677246)

[I. User Requirement Specification 20](#_Toc349677247)

[1. Public users requirements 20](#_Toc349677248)

[2. Administrator requirements 20](#_Toc349677249)

[3. Staff requirements 20](#_Toc349677250)

[II. System Requirement Specification 21](#_Toc349677251)

[1. System Features Overview 21](#_Toc349677252)

[2. Public User 22](#_Toc349677253)

[2.1 Login 22](#_Toc349677254)

[2.2 Request Directions 23](#_Toc349677255)

[2.2.a Shortest itinerary in term of time 23](#_Toc349677256)

[2.2.b Shortest itinerary in term of distance 25](#_Toc349677257)

[2.2.c Itinerary with least number of buses 27](#_Toc349677258)

[3. Logged Users 30](#_Toc349677259)

[3.1 Logout 30](#_Toc349677260)

[3.2 Change Own Account Information 30](#_Toc349677261)

[4. Admin 32](#_Toc349677262)

[4.1 View List of Accounts 32](#_Toc349677263)

[4.2 Add an Account 33](#_Toc349677264)

[4.3 Edit an Account Information 35](#_Toc349677265)

[5. Staff 38](#_Toc349677266)

[5.1 Compare Local and Government Data 38](#_Toc349677267)

[5.1.a Compare Local & Government Data 38](#_Toc349677268)

[5.1.b Update System Data 39](#_Toc349677269)

[5.2 Manage Bus Stops 41](#_Toc349677270)

[5.2.a View list of Bus Stops 41](#_Toc349677271)

[5.2.b Add a new Bus Stop 42](#_Toc349677272)

[5.2.c Edit a Bus Stop 44](#_Toc349677273)

[5.2.d Delete a Bus Stop 45](#_Toc349677274)

[5.3 Manage Bus Routes 46](#_Toc349677275)

[5.3.a View list of Bus Routes 46](#_Toc349677276)

[5.3.b Add a new Bus Route 47](#_Toc349677277)

[5.3.c Edit a Bus Route 49](#_Toc349677278)

[5.3.d Delete a Bus Route 50](#_Toc349677279)

[5.4 Manage Bus Schedules 51](#_Toc349677280)

[5.4.a View list of Schedule for a Bus Route 52](#_Toc349677281)

[5.4.b Add a new Schedule for a Bus Route 53](#_Toc349677282)

[5.4.c Edit a Schedule of a Bus Route 54](#_Toc349677283)

[5.4.d Delete a Schedule of a Bus Route 55](#_Toc349677284)

[6. Software System Attributes 56](#_Toc349677285)

[6.1 Usability 56](#_Toc349677286)

[6.1.a Graphic User Interface 56](#_Toc349677287)

[6.1.b Usability for public user 57](#_Toc349677288)

[6.1.c Usability for admin and staff 57](#_Toc349677289)

[6.1.d Installation 57](#_Toc349677290)

[6.2 Reliability 57](#_Toc349677291)

[6.3 Availability 57](#_Toc349677292)

[6.4 Security 57](#_Toc349677293)

[6.5 Maintainability 57](#_Toc349677294)

[6.6 Portability 57](#_Toc349677295)

[6.7 Performance 57](#_Toc349677296)

[III. Main Flows 57](#_Toc349677297)

[1. Public User 57](#_Toc349677298)

[2. Staff 58](#_Toc349677299)

[IV. Entity Relationship - Conceptual 60](#_Toc349677300)

[D. Software Design Description (SDD) 61](#_Toc349677301)

[I. Design Overview 61](#_Toc349677302)

[II. System Architectural Design 61](#_Toc349677303)

[1. Choice of System Architecture 61](#_Toc349677304)

[2. Discussion of Alternative Designs 61](#_Toc349677305)

[3. Description of System Interface 61](#_Toc349677306)

[III. Component Diagram 61](#_Toc349677307)

[IV. Detailed Description of Components 61](#_Toc349677308)

[1. CRC Cards (Class-Responsibility-Collaborators) 61](#_Toc349677309)

[*2.* Component-*n* 61](#_Toc349677310)

[2.1 Class Diagram 61](#_Toc349677311)

[2.2 Class Diagram Explanation 61](#_Toc349677312)

[2.3 Algorithms of important methods in each class specified in pseudo code or by Flow-Chart 62](#_Toc349677313)

[V. Sequence Diagram 62](#_Toc349677314)

[*VI.* User Interface Design 62](#_Toc349677315)

[1. Description of the User Interface 62](#_Toc349677316)

[1.1 Screen Images 62](#_Toc349677317)

[1.2 Objects and Actions 63](#_Toc349677318)

[VII. Database Design or Data Structures 63](#_Toc349677319)

# List of Figures

[Figure 1: Agile Development 13](#_Toc349680250)

[Figure 2: System Features Overview 21](#_Toc349680251)

[Figure 3: Public User Features 22](#_Toc349680252)

[Figure 4: Logged Users Features 30](#_Toc349680253)

[Figure 5: Admin Features 32](#_Toc349680254)

[Figure 6: Synchronize Data 38](#_Toc349680255)

[Figure 7: Manage Bus Stops 41](#_Toc349680256)

[Figure 8: Manage Bus Routes 46](#_Toc349680257)

[Figure 9: Manage Bus Schedules 51](#_Toc349680258)

[Figure 10: Public User Main Flow 58](#_Toc349680259)

[Figure 11: Staff Main Flow 59](#_Toc349680260)

[Figure 12: Entity Relationship 60](#_Toc349680261)

[Figure 13: Public user page 62](#_Toc349680262)

[Figure 14: Synchronize bus routes page and modal for synchronizing process 63](#_Toc349680263)

[Figure 15: Routes management page 64](#_Toc349680264)

[Figure 16: Bus stop management page and modal for editing bus stop 65](#_Toc349680265)

# List of Tables

[Table 1: Software Introduction 10](#_Toc353280619)

[Table 2: Use Case PU001 - Login 23](#_Toc353280620)

[Table 3: Use Case PU002 - Request Shortest Itinerary in term of time 25](#_Toc353280621)

[Table 4: Use Case PU003 - Request Shortest Itinerary in term of distance 27](#_Toc353280622)

[Table 5: Use Case PU004 - Request Itinerary with least number of buses 29](#_Toc353280623)

[Table 6: Use Case LU002 - Change Own Account Information 32](#_Toc353280624)

[Table 7: Use Case AD001 - View List of Accounts 33](#_Toc353280625)

[Table 8: Use Case AD002 - Add an Account 35](#_Toc353280626)

[Table 9: Use Case AD003 - Edit an Account Information 37](#_Toc353280627)

[Table 10: Use Case ST001 - Compare Local & Government Data 39](#_Toc353280628)

[Table 11: Use Case ST002 - Update System Data 40](#_Toc353280629)

[Table 12: Use Case ST003 - View list of Bus Stops 42](#_Toc353280630)

[Table 13: Use Case ST004 - Add a new Bus Stop 44](#_Toc353280631)

[Table 14: Use Case ST005 - Edit a Bus Stop 45](#_Toc353280632)

[Table 15: Use Case ST006 - Delete a Bus Stop 46](#_Toc353280633)

[Table 16: Use Case ST007 - View list of Bus Routes 47](#_Toc353280634)

[Table 17: Use Case ST008 - Add a new Bus Route 49](#_Toc353280635)

[Table 18: Use Case ST009 - Edit a Bus Route 50](#_Toc353280636)

[Table 19: Use Case ST010 - Delete a Bus Route 51](#_Toc353280637)

[Table 20: Use Case ST011 - View list of Schedule for a Bus Route 52](#_Toc353280638)

[Table 21: Use Case ST012 - Add a new Schedule for a Bus Route 54](#_Toc353280639)

[Table 22: Use Case ST013 - Edit a Schedule for a Bus Route 55](#_Toc353280640)

[Table 23: Use Case ST014 - Delete a Schedule of a Bus Route 56](#_Toc353280641)

# Introduction

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Project Title:** | | Bus Number and Route Suggestion and Management | | | | | | |
| **Start Date:** | | Jan 7th, 2013 | | | **Finished Date:** | | Apr 22nd, 2013 | |
| **Overview of Existing Scenario and Project’s Objectives:** | | * Ho Chi Minh City (HCHM) is the largest city and one of the most important economic hubs of Vietnam. The key means of HCMC transportation system are bus and private vehicles, including 3.2 % by the former and 90 % by the later (with motorcycles accounts for 78.8 %, cars 4.8 %, and bicycles 5 %) (Hong Tan Van, 2009)[[1]](#endnote-1). * The bus system in HCMC consists of bus routes. Each bus route has an outward trip and a return trip. The outward trip and the return trip of one bus route might share the same, some or use totally different bus stops. Moreover, different bus routes might also intersect each other at joint bus stops. Each bus route is comprised many buses. Each bus leaves the station with a predetermined time-table. The interval period a bus takes to travel from one station or bus stop to another is around 5 to 10 minutes. HCMC buses are cheap and plentiful, serving more than 110 routes around greater HCMC. * According to the Public Passenger Transport Management and Control Center of HCMC, HCMC bus system currently includes 110 routes. In 2012, there are approximately 370 million arrivals (compared to 318 million in 2011) and more than 392,740 phone call queries asking about bus routes information, increased by 74,400 calls (http://www.tinmoi.vn/nam-2012-tphcm-co-370-trieu-luot-nguoi-di-xe-buyt-011153226.html, n.d.)[[2]](#endnote-2). * In respect to the above statistics, it is obvious that the demand of bus travelling as well as bus routes’ information has become an urgent matter. As a result, we've made attempt to create a bus routes searching system. Here are our initial results in aiming to construct a bus routes searching system in Ho Chi Minh City. | | | | | | |
| **Overview of Existing Solutions:** | | * Government hotline: The government hotline system is friendly and easy to use but it is not designed to serve a large number of users simultaneously. In addition, it is costly and slower in comparison with other services. Many find it difficult to remember hotline number as well as the directions given by operators. * Government public website: The government provides access to public bus information at http://www.buyttphcm.com.vn/. Up-to-date information (i.e. ticket prices, ticket counters, bus routes, etc) can be found here. The website also provides mean for user to get directions, but this service has some shortcomings:   + It is under-performed.   + It is not reliable for it cannot supply any direction for many requests.   + It does not support searching by time of departure or time of arrival.   + It does not support multi-destination search. * Google maps service: Those who use Google maps service at https://maps.google.com/ can find bus direction in HCHM via their engine’s transit service search. This website has a user-friendly and easy-to-use interface and is designed to serve a very large number of users. The service is not satisfiable, however:   + The data is not up-to-date leads to the suggested directions become rather unreliable.   + It does not support searching by multiple criteria (e.g. shortest by distance, shortest by time, least number of bus used …) * Community project [www.donxebuyt.com](http://www.donxebuyt.com): A project to help people finding bus routes. However:   + Its interface is not user-friendly.   + It is very slow (it often takes about 10 seconds for each request).   + The data is out-of-dated. | | | | | | |
| **Proposed Solutions:** | | * A system needs to be established which takes care of bus routes data and bus time schedule. These data should be maintained by staff to keep it up-to-date. It should be possible to synchronize automatically with the data from government website. * The system should be able to forecast the time buses reach a bus stop. This can be done by driver experience or via mathematical method or a combination of both. * The system should also be able to provide a mean by which users can search and get bus directions. Search queries can be carried out by many criteria: shortest time, shortest distance or least number of buses used… Users can also inquiry for directions of multi-destination trips. The direction given by the system should be aware of user’s expectation of departure time or arrival time. The system, when possible, should give more than one direction to benefit user’s self-preference. * The system should be hosted on the web for ease of access. It is possible to expose service as an Android application for mobile devices. * It is possible to implement a component to report buses’ locations for purpose of tracking and data collecting to serve bus time forecasting. | | | | | | |
| **Approach:** | | * Research more about user’s real experience. * Make attempts to create a product that overcome all shortcomings of existing similar services. | | | | | | |
| **References:** | | * <http://www.buyttphcm.com.vn> * <https://maps.google.com> * <http://www.donxebuyt.com/> | | | | | | |
|  | | | | | | | | |
| **Functions:** | | | | | | | | |
| **Group Functions** | | **Summary** | | | | | | |
| **Management Functions** | | * Authentication and Authorization: User must be authenticated and authorized to use management functions. These functions are aimed to be used by management staff. * Bus Route Management: Allow staff to view, add, modify and delete bus route when needed. * Bus Time Schedule: Allow staff to schedule for buses’ departure by day of week or by specific date of year. * Time forecast: Estimate time a bus reaching a bus stop. * Collecting data from government website: Collect and synchronize part of data from government website automatically. | | | | | | |
| **Public Service** | | * Bus Direction Suggestion based on:   + Departure Time (only for fastest)   + Optimize time or distance or cost or number of bus used   + Maximum walking transit   + Maximum bus used (except optimizing number of bus used)   + Minimum transit time (in case of optimizing time) * Should also provide alternative directions * Able to handle one or more destinations | | | | | | |
|  | | | | | | | | |
| **Roles and Responsibility:** | | | | | | | | |
| **No** | **Full name** | | | **Role** | | **Position** | | **Contact** |
| 1 | Kiều Trọng Khánh | | | Project Manager | | Instructor | | KhanhKT@fpt.edu.vn |
| 2 | Nguyễn Phan Quang Nhật | | | Developer | | Team leader | | NhatNPQ60143@fpt.edu.vn |
| 3 | Nguyễn Viết Vĩnh | | | Developer | | Team member | | VinhNV00330@fpt.edu.vn |
| 4 | Đoàn Xuân Quang | | | Developer | | Team member | | QuangDX60448@fpt.edu.vn |
| 5 | Bùi Tường Thi | | | Developer | | Team member | | ThiBT00721@fpt.edu.vn |
|  | | | | | | | | |
| **Project Authorization:** | | | | | | | | |
| **Approved by** | | | | **Role** | | | | **Date** |
| Kiều Trọng Khánh | | | | Project Manager | | | |  |
|  | | | | | | | | |
| **Comments:** | | |  | | | | | |

Table 1: Software Introduction

# Project Management Plan

## Problem Definition

### Capstone Project Name

* Project full name: Bus Number and Route Suggestion and Management
* Project code: BUSSUG

### Problem Abstract

Ho Chi Minh City (HCHM) is the largest city and one of the most important economic hubs of Vietnam. The key means of HCMC transportation system are bus and private vehicles, including 3.2 % by the former and 90 % by the later (with motorcycles accounts for 78.8 %, cars 4.8 %, and bicycles 5 %) (Hong Tan Van, 2009)1.

The bus system in HCMC consists of bus routes. Each bus route has an outward trip and a return trip. The outward trip and the return trip of one bus route might share the same, some or use totally different bus stops. Moreover, different bus routes might also intersect each other at joint bus stops. Each bus route is comprised many buses. Each bus leaves the station with a predetermined time-table. The interval period a bus takes to travel from one station or bus stop to another is around 5 to 10 minutes. HCMC buses are cheap and plentiful, serving more than 140 routes around greater HCMC.

According to the Public Passenger Transport Management and Control Center of HCMC, HCMC bus system currently includes 150 routes. In 2012, there are approximately 370 million arrivals (compared to 318 million in 2011) and more than 392,740 phone call queries asking about bus routes information, increased by 74,400 calls (http://www.tinmoi.vn/nam-2012-tphcm-co-370-trieu-luot-nguoi-di-xe-buyt-011153226.html, n.d.)2.

In respect to the above statistics, it is obvious that the demand of bus travelling as well as bus routes’ information has become an urgent matter. As a result, we've made attempt to create a bus routes searching system. Here are our initial results in aiming to construct a bus routes searching system in Ho Chi Minh City.

### Project Overview

#### The Current Systems

##### Government hotline

The government hotline system is friendly and easy to use but it is not designed to serve a large number of users simultaneously. In addition, it is costly and slower in comparison with other services. Many find it difficult to remember hotline number as well as the directions given by operators.

##### Government public website

The government provides access to public bus information at http://www.buyttphcm.com.vn/. Up-to-date information (i.e. ticket prices, ticket counters, bus routes, etc.) can be found here. The website also provides mean for user to get directions, but this service has some shortcomings:

* + It is under-performed.
  + It is not reliable for it cannot supply any direction for multiple requests.
  + It does not support searching by time of departure or time of arrival.
  + It does not support multi-destination search.

##### Google maps service

Those who use Google maps service at https://maps.google.com/ can find bus direction in HCHM via their engine’s transit service search. This website has a user-friendly and easy-to-use interface and is designed to serve a very large number of users. The service is not satisfactory, however:

* + The data is not up-to-date leads to the suggested directions become rather unreliable.
  + It does not support searching by multiple criteria (e.g. shortest by distance, cheapest, shortest by time, least number of buses used …)

##### [www.donxebuyt.com](http://www.donxebuyt.com)

People can use the community project at [www.donxebuyt.com](http://www.donxebuyt.com) to find bus directions. However this service has some shortcomings:

* + It is not user-friendly.
  + It is very slow (it often takes about 10 seconds for each request).
  + The data is out-of-dated.

#### The Proposed System

The system needs to be established should have these abilities:

* Takes care of bus routes data and bus time schedule. These data should be maintained by staff to keep it up-to-date. It should be possible to synchronize automatically with the data from government website.
* Forecast the time buses reach a bus stop. This can be done by driver experience or via mathematical method or a combination of both.
* Provide a mean by which users can search and get bus directions. Search queries can be carried out by many criteria: shortest time, shortest distance, least number of buses used or cheapest… Users can also inquiry for directions of multi-destination trips. The direction given by the system should be aware of user’s expectation of departure time. The system, when possible, should give more than one direction to benefit user’s self-preference.
* Be hosted on the web for ease of access. It is possible to expose service as an Android application for mobile devices.

#### Boundaries of the System

* The system is currently used for internal state-supported city bus routes which are managed by Ho Chi Minh City (www.buyttphcm.com.vn, n.d.)[[3]](#endnote-3).
* The system is used to manage these aspects of city bus system:
  + Itinerary of buses
  + Departure time table
* The system is not intended for managing these aspects:
  + Buses
  + Human Resources
  + Finance
* The system can give direction suggestions which include walking and/or using city buses.
* The system supports Vietnamese and English for public users and English for staff.
* The completed product includes
  + The website for public users and staff.
  + Android software to access service via Android smartphone.
  + All the process involved documents.

#### Development Environment

##### Hardware Requirements

* Internet connected computer (ADSL 4Mbs).

##### Software Requirements

* Operating System: Windows 7
* IDE: Visual Studio 2010
* DBMS: SQL Server Express 2008
* Subversion Client: Tortoise SVN, Visual SVN
* Source Control: Assembla Free Private Subversion® Repository[[4]](#endnote-4)
* Tickets System: Assembla Free Tickets Tool for Agile Task Management[[5]](#endnote-5)

## Project Organization

### Software Process Model

Project is developed under agile model.

Figure 1: Agile Development (www.agilekiwi.com, n.d.)[[6]](#endnote-6)

### Roles and Responsibilities

|  |  |  |  |
| --- | --- | --- | --- |
| No | Full name | Role in Group | Responsibilities |
| 1 | Kieu Trong Khanh | Project Manager | * Specify user requirement * Control the development process * Give out technique and business analysis support |
| 2 | Nguyen Phan Quang Nhat | Team Leader, BA, DEV, Tester | * Managing process * Designing database * Clarifying requirements * Support technique * Prepare documents * Create test plan * Coding * Testing |
| 3 | Nguyen Viet Vinh | BA, DEV, Tester | * Designing database * Clarifying requirements * Support technique * Coding * Creating test cases * Testing |
| 4 | Doan Xuan Quang | BA, DEV, Tester | * Coding * Clarifying requirements * GUI Design * Creating test plan and test cases * Testing |
| 5 | Bui Tuong Thi | BA, DEV, Tester | * Coding * Clarifying requirements * GUI Design * Creating test plan and test cases * Testing |

### Tools and Techniques

* Front-end: html, css, javascript, json, jQuery, AJAX, Google Map API
* Back-end: MVC3, .NET framework 4.0, Entity Framework 4.4, log4net, Google Map API, Task Parallel Library.
* Web server: Microsoft IIS 7
* DBMS: SQL Server Express 2008

## Project Management Plan

### Project Iterations

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Phase/Iteration | Description | Deliverables | Resources Needed | Dependencies and Constraints | | Risks |
| 1. Preliminary Investigation or Analysis. | * Study similar existing systems. * Identify and clarify requirements for the system in general. | * Introduction of proposed system. * Main functions. * Project Iteration Plan | 20 man-days | N/A | | * Project may not be feasible for developing because of strange requirements or lack of technologies and/or data. |
| 2. Data Synchronizer | - Study government website. Find a solution to allow synchronize parts of data from it into our system. | - Data Synchronizer Component | 30 man-days | N/A | | * Data structure of government may be changed time to time. * Mal-form data. * The government website may be down. |
| 3. Bus Routes & Bus Stops Management | * Provide mean for staff to view, add, remove and edit bus stops and bus routes in the system | * Bus Stops Management Component. * Bus Route Management Component. | 40 man-days | N/A | | * Mismatch between system data structure and government public website data structure. |
| 4. Direction Suggestion | * Study and implement algorithm to produce direction suggestion based on system data and input criteria. | * A mock of Time Schedule Component. * Direction Suggestion Component. | 60 man-days | Depend on Bus Routes Management & Bus Stops Management Components | | * Lack of experience or bias between algorithms concepts and business concepts. * Feasibility of algorithm performance. |
| 5. Time Schedule | * Provide mean for staff to view, add, remove and edit departure time table for buses. | * Time Schedule Component | 20 man-days | | Depend on Bus Routes Management & Bus Stops Management Components | Lack of experience or bias when defining bus time schedule. |
| 6. Authentication and Authorization | * Authenticate and set permission for which user to use which function. | * Integrate authentication and authorization into system. | 20 man-days | | N/A |  |
| 7. Android Client for public service | * Implement android application to be used as mobile client. | * Android Client Application | 40 man-days | | Depend on Direction Suggestion Component. | Lack of experience of Android development. |

### Iteration Detail

#### Phase 1: Preliminary Investigation or Analysis

|  |  |  |
| --- | --- | --- |
| Task | Description | Author |
| 1. Identifying and studying existing systems | * Find which systems currently provide similar service, their strengths and weaknesses | NhatNPQ, VinhNV, QuangDX, ThiBT |
| 2. Identifying and clarifying main functions | * Define which main function system should provide. | NhatNPQ, VinhNV, QuangDX, ThiBT |
| 3. Introduction | * Complete Introduction Report | NhatNPQ, VinhNV, QuangDX, ThiBT |
| 4. Project Management Plan | * Making Project Management Plan. | NhatNPQ |
| 5. Website Prototype | * Build a prototype of proposed system. | VinhNV, QuangDX |

#### Phase 2: Data Synchronizer

|  |  |  |
| --- | --- | --- |
| Task | Description | Author |
| 1. Identify Requirement & Planning | * Which feature this component should have and how to implement. | NhatNPQ, VinhNV, QuangDX, ThiBT |
| 2. Design DB | * Design DB to store bus stops, bus routes. | NhatNPQ |
| 3. Getting Data from buythcm.com.vn | * Study how to get data from buythcm.com.vn | NhatNPQ |
| 4. Convert Data | * Convert data getting from buythcm.com.vn to internal format | ThiBT |
| 5. Compare Data | * Compare data between 2 systems to find which bus routes have changed. | NhatNPQ |
| 6. Implement GUI | * Showing changes in bus routes to users | VinhNV, QuangDX |
| 7. Paging, Filtering | * Allow user to view, search, filter out changes base on some criteria like Code, Name, Status | VinhNV |
| 8. Drawing route on map. | * Visualize bus route changes on map using Google Map API | QuangDX |
| 9. Testing | * Test system behavior and performance | NhatNPQ, QuangDX, VinhNV, ThiBT |
| 10. Document | * Adding SRS, SDD, Installation Guide, Manual Guide | NhatNPQ, QuangDX, VinhNV, ThiBT |

#### Phase 3: Bus Routes & Bus Stops Management

|  |  |  |
| --- | --- | --- |
| Task | Description | Author |
| 1. Identify Requirement & Planning | * Which feature this component should have and how to implement. | NhatNPQ, VinhNV, QuangDX, ThiBT |
| 2. Bus Stop Model | * Dealing with Add, Remove, Modify, Searching Bus Stops | NhatNPQ |
| 3. Bus Route Model | * Dealing with Add, Remove, Modify, Searching Bus Routes and their details. | NhatNPQ |
| 4. Computing Distance | * Computing straight distance between two geography location | QuangDX |
| 5. Implement GUI | * Allow user to view, search, add, remove, edit information of bus stops and bus routes * Interacting with map for ease of editing. | VinhNV, QuangDX |
| 6. Testing | * Test system behavior and performance | NhatNPQ, QuangDX, VinhNV, ThiBT |
| 7. Document | * Adding SRS, SDD, Installation Guide, Manual Guide | NhatNPQ, QuangDX, VinhNV, ThiBT |

#### Phase 4: Direction Suggestion

|  |  |  |
| --- | --- | --- |
| Task | Description | Author |
| 1. Identify Requirement & Planning | * Which feature this component should have and how to implement. | NhatNPQ, VinhNV, QuangDX, ThiBT |
| 2. Study and implement algorithm | * Building graph from DB. * Find solution with Dijkstra, K-shortest-path, and Hamilton algorithms using different weighting methods. | NhatNPQ |
| 3. Result refinement | * Filter out non-human results. | NhatNPQ |
| 4.Implement GUI | * Allow user to input their expectations and show direction suggestion. * Visualize result on map. | VinhNV |
| 5. Mock instance of Time Schedule Component | * A Mock instance of Time Schedule Component due to its absence. | NhatNPQ |
| 6. Testing | * Test system behavior and performance | NhatNPQ, QuangDX, VinhNV, ThiBT |
| 7. Document | * Adding SRS, SDD, Installation Guide, Manual Guide | NhatNPQ, QuangDX, VinhNV, ThiBT |

#### Phase 5: Time Schedule

|  |  |  |
| --- | --- | --- |
| Task | Description | Author |
| 1. Identify Requirement & Planning | * Which feature this component should have and how to implement. | NhatNPQ, VinhNV, QuangDX, ThiBT |
| 2. Design DB | * Design DB to store time schedule rules and their details. | NhatNPQ |
| 3. Time Schedule Model | * Allow user to add, remove, search and edit time schedules. * Building departure time table based on time schedules. | NhatNPQ |
| 4. Implement GUI | * Allow user to view, search, add, remove, edit information time schedule | VinhNV |
| 5. Testing | * Test system behavior and performance | NhatNPQ, QuangDX, VinhNV, ThiBT |
| 6. Document | * Adding SRS, SDD, Installation Guide, Manual Guide | NhatNPQ, QuangDX, VinhNV, ThiBT |

#### Phase 6: Authentication and Authorization

|  |  |  |
| --- | --- | --- |
| Task | Description | Author |
| 1. Identify Requirement & Planning | * Which feature this component should have and how to implement. | NhatNPQ, VinhNV, QuangDX, ThiBT |
| 2. Design DB | * Design DB to store user accounts and their information. | NhatNPQ |
| 3. User Model | * Allow admin to add, remove, search and edit user accounts. * Allow users to edit their information. | NhatNPQ |
| 4.Authentication and Authorization | * Adding authentication and authorization into current system with minimum affecting existing features. | NhatNPQ |
| 5. Testing | * Test system behavior and performance | NhatNPQ, QuangDX, VinhNV, ThiBT |
| 6. Document | * Adding SRS, SDD, Installation Guide, Manual Guide | NhatNPQ, QuangDX, VinhNV, ThiBT |

#### Phase 7: Android Client

|  |  |  |
| --- | --- | --- |
| Task | Description | Author |
| 1. Identify Requirement & Planning | * Which feature this component should have and how to implement. | NhatNPQ, VinhNV, QuangDX |
| 2. Define Communication Interface | * Define how clients communicate with server. | NhatNPQ |
| 3. Implementation | * Adding authentication and authorization into current system with minimum affecting existing features. | QuangDX |
| 5. Testing | * Test system behavior and performance | NhatNPQ, QuangDX, VinhNV |
| 6. Document | * Adding SRS, SDD, Installation Guide, Manual Guide | NhatNPQ, QuangDX, VinhNV |

### Meeting minutes

Refer to MeetingMinutes folder

## Coding Convention

Follow Microsoft Recommendation for C# Coding conventions[[7]](#endnote-7).

# Software Requirement Specification

## User Requirement Specification

Bus Number and Route Suggestion and Management provides publics user an easy mean to search and get information, routes and directions of public bus transport. It also allows staff to manage and control data, time schedules of HCHM bus system. As such, for each user there are respective features the system would provide:

### Public users requirements

* Public users could request directions
* Login to system as an administrator or staff (with a valid account)

### Administrator requirements

* Administrators could view list of accounts.
* Administrators could create an account.
* Administrators could update information of an account.
* Administrators could view system information.

### Staff requirements

* Staff could compare system data with government data.
* Staff could update one or many bus route’s data with government data.
* Staff could view the list of bus stops.
* Staff could add a new bus stop.
* Staff could edit bus stop information.
* Staff could delete a bus stop.
* Staff could view the list of bus routes.
* Staff could add a new bus route.
* Staff could edit a bus route.
* Staff could delete a bus route.
* Staff could view list of schedules for a bus route.
* Staff could create a new schedule.
* Staff could edit a schedule.
* Staff could delete a schedule.

## System Requirement Specification

### System Features Overview

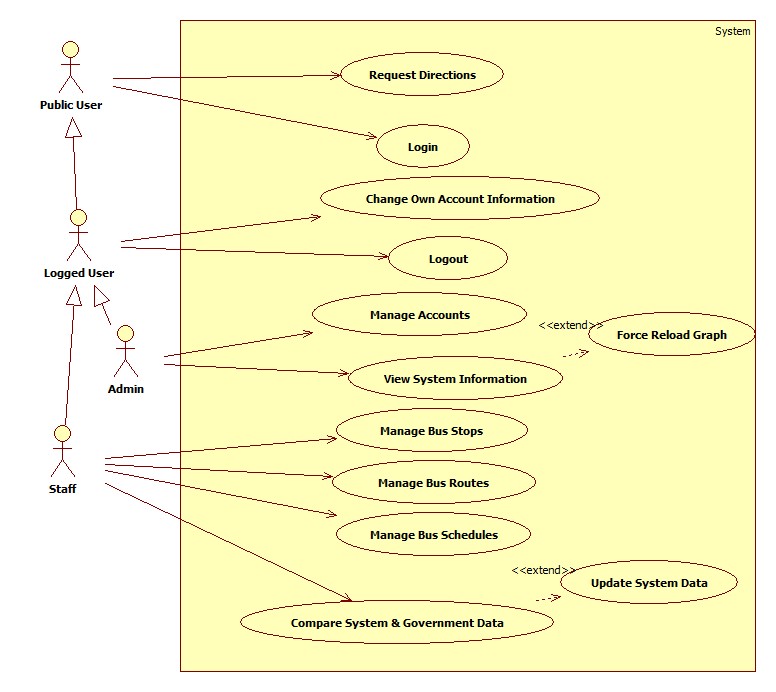


Figure 2: System Features Overview

### Public User

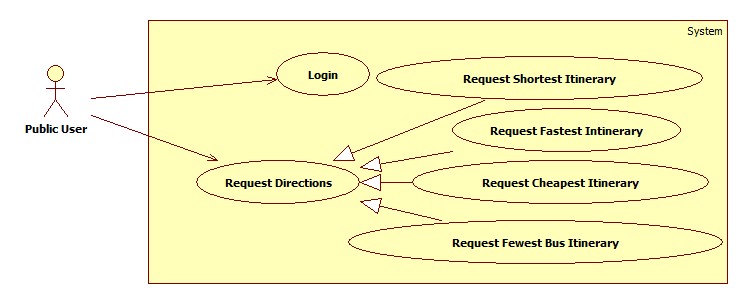


Figure 3: Public User Features

#### Login

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| USE CASE–PU001 | | | | |
| Use-case No. | PU001 | Use-case Version | | 0.1 |
| Use-case Name | Login | | | |
| Author | Nguyen Phan Quang Nhat | | | |
| Date | 23/01/2013 | Priority | High | |
| **Actor:** Public User  **Summary:** This use case allows administrator, staff to log on into the system  **Goal:** Authenticate user.  **Triggers:** User access website area that is intended for staff and administrator.  **Pre-conditions:** User has an account and has not logged in yet  **Post-conditions.**  **Success:** User has logged in.  Redirects to Staff or Admin page  **Fail:** Redirects to Public page  **Main Success Scenario.**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | User enter Staff or Admin area of the website | Display the form to login includes:   * Username: textbox. * Password: password field. * Login: button * Reset: button | | 2 | User fills username and password |  | | 3 | User click “Login” button [Alternative 1] [Exception 1] | Check for account validity and  Redirect to Synchronize Page for Staff or Account Management page for Admin if user’s credential is valid. [Alternative 2] |   **Alternative Scenario.**   |  |  |  | | --- | --- | --- | | No. | Actor Action | System Response | | 1 | User click “Reset” | Clear username and password textbox. | | 2 | User credential is not valid | Display error message “The username or password you entered is incorrect”. |   **Exceptions**.   |  |  |  | | --- | --- | --- | | No. | Actor Action | System Response | | 1 | User leaves username textbox empty. | Display error message “username cannot be empty”. |   Relationships: N/A  Business Rules: N/A | | | | |

Table 2: Use Case PU001 - Login

#### Request Directions

##### Cheapest Itinerary

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| USE CASE–PU002 | | | | |
| Use-case No. | PU002 | Use-case Version | | 0.1 |
| Use-case Name | Request Cheapest Itinerary | | | |
| Author | Nguyen Phan Quang Nhat, Nguyen Viet Vinh | | | |
| Date | 23/01/2013 | Priority | High | |
| **Actor:** Public User  **Summary:** This use case allows user to find the best bus routes for destinations in term of price.  **Goal:** User’s able to get the itinerary of bus routes, from the start point to the end point.  **Triggers:** User enter public service page.  **Pre-conditions:** None  **Post-conditions:**  **Success:** Display the most suitable itineraries.  **Fail:** Display message “No suitable itineraries found”.  **Main Success Scenario.**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | User enters public page | Show Public page includes:   * A map centered at user location [Exception 1]. * Departure Address: textbox. * Destination Address 1: textbox. * Destination Address 2: textbox. * Destination Address 3: textbox. * Max bus used: combo box ( value range: 1-5; default: 3) * Maximum walking transits: combo box (value: none, 100m, 200m, 300m, 400m, 500m; default: 300m) * Search Type: combo box (value: “Rẻ nhất”, “Ngắn nhất”, “Ít buýt nhất”, “Nhanh nhất”;default: “Ít buýt nhất”) | | 1 | User enters address of departure in departure text box [Alternative 1] | Show departure location on map [Exception 2, 3, 4] | | 2 | User enters address of destination in destination text box [Alternative 2] | Show destination location on map [Exception 2, 3, 4] | | 3 | User enters walking transits, max bus used. |  | | 4 | User select “Rẻ nhất” in search type. |  | | 5 | Click “Tìm” | Show the found bus route’s itinerary [Exception 5, 6, 7, 8] and draw the first itinerary on map. | | 6 | User select one of found itinerary | Draw corresponding itinerary on map. |   **Alternative Scenario.**   |  |  |  | | --- | --- | --- | | No. | Actor Action | System Response | | 1 | User point out departure location on map. | Show departure address in departure text box [Exception 3]. | | 2 | User point out destination location on map. | Show destination address in destination text box [Exception 3]. | | 3 | Select “Ngắn nhất” | Refer to Use Case PU003 | | 4 | Select “Ít buýt nhất” | Refer to Use Case PU004 | | 5 | Select “Nhanh nhất” | Refer to Use Case PU005 |   **Exceptions**.   |  |  |  |  | | --- | --- | --- | --- | | No. | Actor Action | System Response | | | 1 | Cannot get user location due to device capability or user privacy | The map is centered at Main City Bus Station. | | | 2 | Cannot find entered address | Display error Message “Cannot find entered address” | | | 3 | Departure or Destination is outside of HCM City | Display error Message “The system cannot provide directions outside of HCM city”. | | | 4 | More than one locations return from Google Map API | Display list of possible locations for user to select. | | 5 | User leaves Departure Address empty. | Display error message “Please select departure location”. | | 6 | User leaves Destination Address empty. | Display error message “Please select destination location”. | | | 7 | The System cannot found any result based on user criteria. | Display error Message “No suitable itinerary found”. | |   Relationships: Shortest itinerary, Fewest bus Itinerary, Fastest Itinerary  Business Rules: Google Map API | | | | |

Table 3: Use Case PU002 - Request Cheapest Itinerary

##### Shortest Itinerary

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| USE CASE–PU003 | | | | |
| Use-case No. | PU003 | Use-case Version | | 0.1 |
| Use-case Name | Request Shortest Itinerary | | | |
| Author | Nguyen Phan Quang Nhat, Nguyen Viet Vinh | | | |
| Date | 23/01/2013 | Priority | High | |
| **Actor:** Public User  **Summary:** This use case allows user to find the best bus routes for destinations in term of distance.  **Goal:** User’s able to get the itinerary of bus routes, from the start point to the end point.  **Triggers:** User enter public service page.  **Pre-conditions:** None  **Post-conditions:**  **Success:** Display the most suitable itineraries.  **Fail:** Display message “No suitable itineraries found”.  **Main Success Scenario.**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | User enters public page | Show Public page includes:   * A map centered at user location [Exception 1]. * Departure Address: textbox. * Destination Address 1: textbox. * Destination Address 2: textbox. * Destination Address 3: textbox. * Max bus used: combo box ( value range: 1-5; default: 3) * Maximum walking transits: combo box (value: none, 100m, 200m, 300m, 400m, 500m; default: 300m) * Search Type: combo box (value: “Rẻ nhất”, “Ngắn nhất”, “Ít buýt nhất”, “Nhanh nhất”; default: “Ít buýt nhất”) | | 1 | User enters address of departure in departure text box [Alternative 1] | Show departure location on map [Exception 2, 3, 4] | | 2 | User enters address of destination in destination text box [Alternative 2] | Show destination location on map [Exception 2, 3, 4] | | 3 | User enters walking transits, max bus used. |  | | 4 | User select “Ngắn nhất” in search type. |  | | 5 | Click “Tìm” | Show the found bus route’s itinerary [Exception 5, 6, 7, 8] and draw the first itinerary on map. | | 6 | User select one of found itinerary | Draw corresponding itinerary on map. |   **Alternative Scenario.**   |  |  |  | | --- | --- | --- | | No. | Actor Action | System Response | | 1 | User point out departure location on map. | Show departure address in departure text box [Exception 3]. | | 2 | User point out destination location on map. | Show destination address in destination text box [Exception 3]. | | 3 | Select “Rẻ nhất” | Refer to Use Case PU002 | | 4 | Select “Ít buýt nhất” | Refer to Use Case PU004 | | 5 | Select “Nhanh nhất” | Refer to Use Case PU005 |   **Exceptions**.   |  |  |  |  | | --- | --- | --- | --- | | No. | Actor Action | System Response | | | 1 | Cannot get user location due to device capability or user privacy | The map is centered at Main City Bus Station. | | | 2 | Cannot find entered address | Display error Message “Cannot find entered address” | | | 3 | Departure or Destination is outside of HCM City | Display error Message “The system cannot provide directions outside of HCM city”. | | | 4 | More than one locations return from Google Map API | Display list of possible locations for user to select. | | 5 | User leaves Departure Address empty. | Display error message “Please select departure location”. | | 6 | User leaves Destination Address empty. | Display error message “Please select destination location”. | | | 7 | The System cannot found any result based on user criteria. | Display error Message “No suitable itinerary found”. | |   Relationships: Cheapest itinerary, Fewest bus Itinerary, Fastest Itinerary  Business Rules: Google Map API | | | | |

Table 4: Use Case PU003 - Request Shortest Itinerary

##### Fewest-Bus Itinerary

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| USE CASE–PU004 | | | | |
| Use-case No. | PU004 | Use-case Version | | 0.1 |
| Use-case Name | Request Fewest-Bus Itinerary | | | |
| Author | Nguyen Phan Quang Nhat, Nguyen Viet Vinh | | | |
| Date | 23/01/2013 | Priority | High | |
| **Actor:** Public User  **Summary:** This use case allows user to find the best bus routes for destinations using least number of buses.  **Goal:** User’s able to get the itinerary of bus routes, from the start point to the end point.  **Triggers:** User enter public service page.  **Pre-conditions:** None  **Post-conditions:**  **Success:** Display the most suitable itineraries.  **Fail:** Display message “No suitable itineraries found”.  **Main Success Scenario.**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | User enters public page | Show Public page includes:   * A map centered at user location [Exception 1]. * Departure Address: textbox. * Destination Address 1: textbox. * Destination Address 2: textbox. * Destination Address 3: textbox. * Maximum walking transits: combo box (value: none, 100m, 200m, 300m, 400m, 500m; default: 300m) * Search Type: combo box (value: “Rẻ nhất”, “Ngắn nhất”, “Ít buýt nhất”, “Nhanh nhất”;default: “Ít buýt nhất”) | | 1 | User enters address of departure in departure text box [Alternative 1] | Show departure location on map [Exception 2, 3, 4] | | 2 | User enters address of destination in destination text box [Alternative 2] | Show destination location on map [Exception 2, 3, 4] | | 3 | User enters walking transits. |  | | 4 | User select “Ít buýt nhất” in search type. |  | | 5 | Click “Tìm” | Show the found bus route’s itinerary [Exception 5, 6, 7, 8] and draw the first itinerary on map. | | 6 | User select one of found itinerary | Draw corresponding itinerary on map. |   **Alternative Scenario.**   |  |  |  | | --- | --- | --- | | No. | Actor Action | System Response | | 1 | User point out departure location on map. | Show departure address in departure text box [Exception 3]. | | 2 | User point out destination location on map. | Show destination address in destination text box [Exception 3]. | | 3 | Select “Rẻ nhất” | Refer to Use Case PU002 | | 4 | Select “Ngắn nhất” | Refer to Use Case PU003 | | 5 | Select “Nhanh nhất” | Refer to Use Case PU005 |   **Exceptions**.   |  |  |  |  | | --- | --- | --- | --- | | No. | Actor Action | System Response | | | 1 | Cannot get user location due to device capability or user privacy | The map is centered at Main City Bus Station. | | | 2 | Cannot find entered address | Display error Message “Cannot find entered address” | | | 3 | Departure or Destination is outside of HCM City | Display error Message “The system cannot provide directions outside of HCM city”. | | | 4 | More than one locations return from Google Map API | Display list of possible locations for user to select. | | 5 | User leaves Departure Address empty. | Display error message “Please select departure location”. | | 6 | User leaves Destination Address empty. | Display error message “Please select destination location”. | | | 7 | The System cannot found any result based on user criteria. | Display error Message “No suitable itinerary found”. | |   Relationships: Shortest itinerary, Fewest bus Itinerary, Fastest Itinerary  Business Rules: Google Map API | | | | |

Table 5: Use Case PU004 - Request Fewest-Bus Itinerary

##### Fastest Itinerary

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| USE CASE–PU005 | | | | |
| Use-case No. | PU005 | Use-case Version | | 0.1 |
| Use-case Name | Request Fastest Itinerary | | | |
| Author | Nguyen Phan Quang Nhat, Nguyen Viet Vinh | | | |
| Date | 23/01/2013 | Priority | High | |
| **Actor:** Public User  **Summary:** This use case allows user to find the best bus routes for destinations in term of arrival time.  **Goal:** User’s able to get the itinerary of bus routes, from the start point to the end point.  **Triggers:** User enter public service page.  **Pre-conditions:** None  **Post-conditions:**  **Success:** Display the most suitable itineraries.  **Fail:** Display message “No suitable itineraries found”.  **Main Success Scenario.**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | User enters public page | Show Public page includes:   * A map centered at user location [Exception 1]. * Departure Address: textbox. * Destination Address: textbox. * Max Bus Used: combo box (value range: 1-5; default 3) * Departure Time: Time Picker (value range: 4:00 – 22:00; default: current time plus 5 minutes or 6:00 if current time is out of value range) * Maximum walking transits: combo box (value: none, 100m, 200m, 300m, 400m, 500m; default: 300m) * Search Type: combo box (value: “Rẻ nhất”, “Ngắn nhất”, “Ít buýt nhất”, “Nhanh nhất”;default: “Ít buýt nhất”) | | 1 | User enters address of departure in departure text box [Alternative 1] | Show departure location on map [Exception 2, 3, 4] | | 2 | User enters address of destination in destination text box [Alternative 2] | Show destination location on map [Exception 2, 3, 4] | | 3 | User enters walking transits, max bus used and departure time. |  | | 4 | User select “Nhanh nhất” in search type. |  | | 5 | Click “Tìm” | Show the found bus route’s itinerary [Exception 5, 6, 7, 8] and draw the first itinerary on map. | | 6 | User select one of found itinerary | Draw corresponding itinerary on map. |   **Alternative Scenario.**   |  |  |  | | --- | --- | --- | | No. | Actor Action | System Response | | 1 | User point out departure location on map. | Show departure address in departure text box [Exception 3]. | | 2 | User point out destination location on map. | Show destination address in destination text box [Exception 3]. | | 3 | Select “Rẻ nhất” | Refer to Use Case PU002 | | 4 | Select “Ngắn nhất” | Refer to Use Case PU003 | | 5 | Select “Ít buýt nhất” | Refer to Use Case PU004 |   **Exceptions**.   |  |  |  |  | | --- | --- | --- | --- | | No. | Actor Action | System Response | | | 1 | Cannot get user location due to device capability or user privacy | The map is centered at Main City Bus Station. | | | 2 | Cannot find entered address | Display error Message “Cannot find entered address” | | | 3 | Departure or Destination is outside of HCM City | Display error Message “The system cannot provide directions outside of HCM city”. | | | 4 | More than one locations return from Google Map API | Display list of possible locations for user to select. | | 5 | User leaves Departure Address empty. | Display error message “Please select departure location”. | | 6 | User leaves Destination Address empty. | Display error message “Please select destination location”. | | | 7 | The System cannot found any result based on user criteria. | Display error message “No suitable itinerary found”. | | | 8 | Departure Time is outside of value range. | Display error message: “Invalid departure time”. | |   Relationships: Shortest itinerary, Fewest bus Itinerary, Fastest Itinerary  Business Rules: Google Map API | | | | |

Table 6: Use Case PU005 - Request Fastest Itinerary

### Logged Users

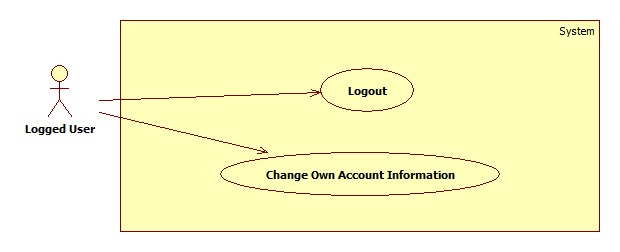


Figure 4: Logged Users Features

#### Logout

#### Change Own Account Information

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| USE CASE–LU002 | | | | |
| Use-case No. | LU002 | Use-case Version | | 0.1 |
| Use-case Name | Edit an Account Information | | | |
| Author | Nguyen Viet Vinh | | | |
| Date | 23/01/2013 | Priority | High | |
| **Actor:** Logged User  **Summary:** This use case allows user to edit his or her account information  **Goal:** Change information of his or her account.  **Triggers:** User clicks “Edit Information” on navigation bar.  **Pre-conditions:** User has logged in as an administrator or staff.  **Post-conditions:**  **Success:** The information of his or her account is updated.  **Fail:** The information of his or her account is not updated. Display error message.  **Main Success Scenario.**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | User clicks “Edit Information” on navigation bar. | Display account information page includes:   * Username: label. * Password: password field (min length: 6, max length: 30) * Full name: textbox (min length: 3, max length: 50) * Email: textbox (regular expression: ^[\_a-z0-9-]+(\.[\_a-z0-9-]+)\*@[a-z0-9-]+(\.[a-z0-9-]+)\*(\.[a-z]{2,4})$) * Is Admin: label. * Is Staff: label. * Save: button * Reset: button | | 2 | User modifies account information |  | | 3 | Click “Save” [Alternative 2, 3] | Valid entered information and update account information. [Exception 1, 2, 3, 4, 5, 6, 7, 8] |   **Alternative Scenario.**   |  |  |  | | --- | --- | --- | | No. | Actor Action | System Response | | 1 | User clicks “Reset” | Refresh the all input controls with original values. | | 2 | User clicks “Reset Password” | Generate a new password for that account, update account password and send it to his or her email |   **Exceptions**.   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Password length is less than 6 | Display error message “Your password is too short”. | | 2 | Password length is more than 30 | Display error message “Your password is too long”. | | 3 | Full name length is less than 3 | Display error message “Your full name is too short”. | | 4 | Full name length is more than 50 | Display error message “Your full name is too long”. | | 5 | Email address is not in correct format. | Display error message “Invalid email address. Please check your email address”. |   Relationships: N/A  Business Rules: N/A | | | | |

Table 7: Use Case LU002 - Change Own Account Information

### Admin

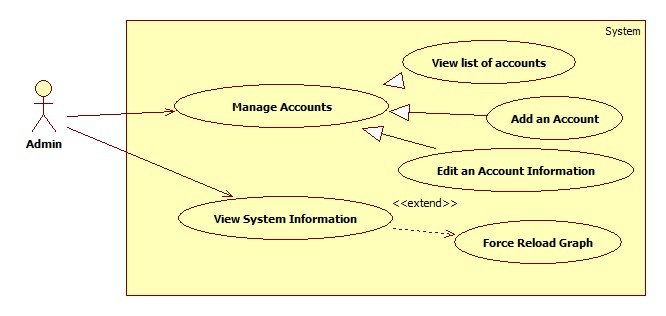


Figure 5: Admin Features

#### View List of Accounts

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| USE CASE–AD001 | | | | |
| Use-case No. | AD001 | Use-case Version | | 0.1 |
| Use-case Name | View List of Accounts | | | |
| Author | Nguyen Phan Quang Nhat, Nguyen Viet Vinh | | | |
| Date | 23/01/2013 | Priority | High | |
| **Actor:** Administrator  **Summary:** This use case allows administrator to view the list of accounts in system.  **Goal:** List of accounts in system is displayed.  **Triggers:** Administrator access Account Management Page.  **Pre-conditions:** User has logged in as an administrator.  **Post-conditions:**  **Success:** List of accounts in system is displayed.  **Fail:** Redirect to General Error Page.  **Main Success Scenario.**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Enter the Account Management page | Display the list of accounts in system and their information includes:   * Add new account: button * A table with these column   + Username: label   + Full name: label   + Email: label   + Is Admin: disabled checkbox.   + Is Staff: disabled checkbox   + Enable: label   + Edit: button |   **Alternative Scenario.**  N/A  **Exceptions**.  N/A  Relationships: N/A  Business Rules: N/A | | | | |

Table 8: Use Case AD001 - View List of Accounts

#### Add an Account

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| USE CASE-AD002 | | | | |
| Use-case No. | AD002 | Use-case Version | | 0.1 |
| Use-case Name | Add an Account | | | |
| Author | Nguyen Phan Quang Nhat, Nguyen Viet Vinh | | | |
| Date | 23/01/2013 | Priority | High | |
| **Actor:** Administrator  **Summary:** This use case allows administrator to add a new account in system.  **Goal:** Create a new account.  **Triggers:** Administrator click “Add New Account” in Account Management Page.  **Pre-conditions:** User has logged in as an administrator.  **Post-conditions:**  **Success:** A new account has been created in system.  **Fail:** None of account is added to the system. Display error message.  **Main Success Scenario.**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Admin clicks “Add New Account” in Account Management Page. | Display account information page includes:   * Username: textbox (min length: 4, max length: 30) * Full name: textbox (min length: 3, max length: 50) * Email: textbox (regular expression: ^[\_a-z0-9-]+(\.[\_a-z0-9-]+)\*@[a-z0-9-]+(\.[a-z0-9-]+)\*(\.[a-z]{2,4})$) * Is Admin: checkbox. * Is Staff: checkbox. * Enable: checkbox. * Add: button * Reset: button | | 2 | Admin enters account information |  | | 3 | Admin clicks “Add” [Alternative 1] | Valid entered information and create a new account with entered information [Exception 1, 2, 3, 4, 5, 6, 7, 8]. |   **Alternative Scenario.**   |  |  |  | | --- | --- | --- | | No. | Actor Action | System Response | | 1 | Admin clicks “Reset” | Clear all input controls on form. |   **Exceptions**.   |  |  |  | | --- | --- | --- | | No. | Actor Action | System Response | | 1 | Username contains whitespaces | Display error message “Username cannot contain white space”. | | 2 | Username’s length less than 4 | Display error message “Your username is too short.” | | 3 | Username’s length is more than 30 | Display error message “Your username is too long”. | | 4 | Full name’s length is less than 3 | Display error message “Your full name is too short”. | | 5 | Full name’s length is more than 50 | Display error message “Your full name is too long”. | | 6 | Email is not in right format. | Display error message “Email is not in right format ” | | 7 | Username is already in the system. | Display error message “Username is already in the system, please choose another one”. |   Relationships: N/A  Business Rules: N/A. | | | | |

Table 9: Use Case AD002 - Add an Account

#### Edit an Account Information

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| USE CASE–AD003 | | | | |
| Use-case No. | AD003 | Use-case Version | | 0.1 |
| Use-case Name | Edit an Account Information | | | |
| Author | Nguyen Phan Quang Nhat, Nguyen Viet Vinh | | | |
| Date | 23/01/2013 | Priority | High | |
| **Actor:** Administrator  **Summary:** This use case allows administrator to edit an account information  **Goal:** Change information of an existing account.  **Triggers:** Administrator click “Edit” associated with an account in Account Management Page.  **Pre-conditions:** User has logged in as an administrator.  **Post-conditions:**  **Success:** The information of edited account is updated.  **Fail:** The information of edited account is not updated. Display error message.  **Main Success Scenario.**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Click “Edit” associated with an account in Account Management page. | Display account information page includes:   * Username: disabled textbox * Full name: textbox (min length: 3, max length: 50) * Email: textbox (regular expression: ^[\_a-z0-9-]+(\.[\_a-z0-9-]+)\*@[a-z0-9-]+(\.[a-z0-9-]+)\*(\.[a-z]{2,4})$) * Is Admin: checkbox. * Is Staff: checkbox. * Enable: checkbox. * Reset Password: button * Save: button * Cancel: button | | 2 | Admin modify account information |  | | 3 | Click “Save” [Alternative 1, 2] | Valid entered information and update account information [Exception 1, 2, 3, 4, 5, 6, 7]. |   **Alternative Scenario.**   |  |  |  | | --- | --- | --- | | No. | Actor Action | System Response | | 1 | Admin clicks “Cancel” | Redirect to Account Management Page. | | 2 | Admin clicks “Reset Password” | Generate a new password for that account, update account password and send it to his or her email |   **Exceptions**.   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Full name’s length is less than 3 | Display error message “Your full name is too short”. | | 2 | Full name’s length is more than 50 | Display error message “Your full name is too long”. | | 2 | Email is not in right format. | Display error message “Email is not in right format ” | | 5 | Admin set himself/herself not an admin | Display error message “You cannot set yourself as non-admin”. |   Relationships: N/A  Business Rules:N/A. | | | | |

Table 10: Use Case AD003 - Edit an account Information

### Staff

#### Compare Local and Government Data

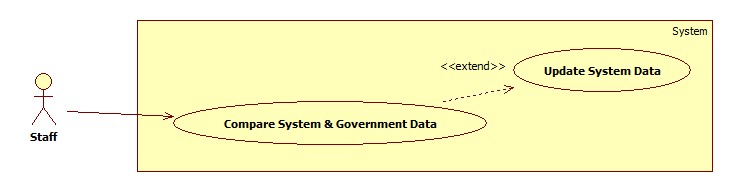


Figure 6: Synchronize Data

##### Compare Local & Government Data

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| USE CASE–ST001 | | | | |
| Use-case No. | ST001 | Use-case Version | | 0.1 |
| Use-case Name | Compare System & Government Data | | | |
| Author | Nguyen Phan Quang Nhat, Doan Xuan Quang, Nguyen Viet Vinh | | | |
| Date | 23/01/2013 | Priority | High | |
| **Actor:** Staff  **Summary:** This use case allows staff to view the differences between system and government bus routes data.  **Goal:** Show differences between system and government bus routes data.  **Triggers:** Staff access “Synchronizer” page.  **Pre-conditions:** User has logged in as a staff.  **Post-conditions.**  **Success:** List of differences is displayed.  **Fail:** List of differences is not displayed or displayed in wrong way.  **Main Success Scenario.**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Staff access Synchronizer page. | Display list of bus routes that system and government data differ includes [Exception 1]:   * A table with these columns:   + No: label   + Code: label + image   + Name: label   + Change: label   + Sync: checkbox * Synchronize: button * A map. | | 2 | Click on a difference (a row in the table of changes) | Draw 2 bus routes on map:   * One from system data. * One from government data. |   **Alternative Scenario.**  N/A  **Exceptions**.   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Cannot access government data. | Display error message “Government data is not available at this time”. | | 2 | Government Data Structure changed | Display error message “Government Data Structure has been changed. Please contract administrator”. |   Relationships: N/A.  Business Rules: Google Map API | | | | |

Table 11: Use Case ST001 - Compare Local & Government Data

##### Update System Data

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| USE CASE–ST002 | | | | |
| Use-case No. | ST002 | Use-case Version | | 0.1 |
| Use-case Name | Update System Data. | | | |
| Author | Nguyen Phan Quang Nhat, Doan Xuan Quang, Nguyen Viet Vinh | | | |
| Date | 23/01/2013 | Priority | High | |
| **Actor:** Staff  **Summary:** This use case allows staff to update system data from government data.  **Goal:** System data is updated from government data.  **Triggers:** Staff click “Synchronize” button displayed in “Synchronizer” page.  **Pre-conditions:** User has logged in as a staff.  **Post-conditions:**  **Success:** System routes data is updated.  **Fail:** System routes data is not updated. Display error message.  **Main Success Scenario.**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Staff access Synchronizer page. | Display list of bus routes that system and government data differ includes [Exception 1]:   * A table with these columns:   + No: label   + Code: label + image   + Name: label   + Change: label   + Sync: checkbox * Synchronize: button * A map. | | 2 | User checks on which bus routes to update. |  | | 3 | User click “Synchronize” button | Update system bus routes data of selected routes. [Exception 3]  Show dialog contain result of synchronization process. | | 4 | User click OK | Close dialog. |   **Alternative Scenario.**  N/A  **Exceptions**.   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Cannot access government data. | Display error message “Government data is not available at this time”. | | 2 | Government Data Structure changed | Display error message “Government Data Structure has been changed. Please contract administrator”. | | 3 | User click “Cancel” while synchronization is in process. | Stop synchronization process. |   Relationships: N/A  Business Rules: Google Map API | | | | |

Table 12: Use Case ST002 - Update System Data

#### Manage Bus Stops

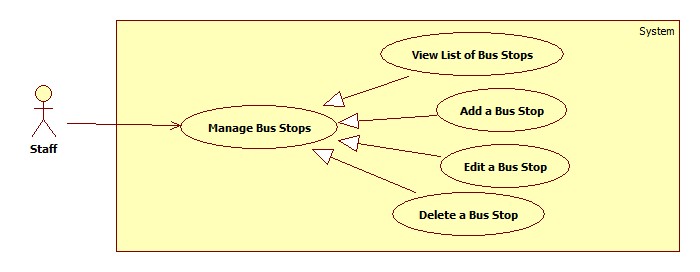


Figure 7: Manage Bus Stops

##### View list of Bus Stops

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| USE CASE–ST003 | | | | |
| Use-case No. | ST003 | Use-case Version | | 0.1 |
| Use-case Name | View List of Bus Stops | | | |
| Author | Nguyen Phan Quang Nhat, Doan Xuan Quang, Nguyen Viet Vinh | | | |
| Date | 23/01/2013 | Priority | High | |
| **Actor:** Staff  **Summary:** This use case allows staff to view the list of bus stops.  **Goal:** List of bus stops in system is displayed.  **Triggers:** Staff access Bus Stops Management Page.  **Pre-conditions:** User has logged in as a staff.  **Post-conditions:**  **Success:** List of bus stops in system is displayed.  **Fail:** Redirect to General Error Page.  **Main Success Scenario.**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Enter the Bus Stops Management page | Display the list of bus stops in system and their information includes:   * Add new bus stop: button * A table with these column   + No: label   + Code: label   + Name: label   + Street: label   + Ward: label   + Edit: button   + Delete: button * A Map to display bus stops location |   **Alternative Scenario.**  N/A  **Exceptions**.  N/A  Relationships: N/A  Business Rules: N/A | | | | |

Table 13: Use Case ST003 - View list of Bus Stops

##### Add a new Bus Stop

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| USE CASE-ST004 | | | | |
| Use-case No. | ST004 | Use-case Version | | 0.1 |
| Use-case Name | Add a new Bus Stop | | | |
| Author | Nguyen Phan Quang Nhat, Doan Xuan Quang, Nguyen Viet Vinh | | | |
| Date | 23/01/2013 | Priority | High | |
| **Actor:** Staff  **Summary:** This use case allows staff to add a new bus stop into system.  **Goal:** Create a new bus stop.  **Triggers:** Staff clicks “Add New Bus Stop” in Account Management Page.  **Pre-conditions:** User has logged in as a staff.  **Post-conditions:**  **Success:** A new bus stop has been created in system.  **Fail:** None of bus stop is added to the system. Display error message.  **Main Success Scenario.**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Staff clicks “Add New Bus Stop” in Bus Stop Management Page. | Display bus stop information page includes:   * Address: textbox (min length: 5, max length 255). * Latitude: number (min: -180.0, max 180.0) * Longitude: number (min: -180.0, max: 180.0) * Add: button * Cancel: button * A map to show bus stop location | | 2 | Staff enters bus stop address [Alternative 1, 2] | * Display bus stop on map. * Show latitude, longitude of new bus stop. | | 3 | Staff clicks “Add” [Alternative 3] | Valid entered information and create a new bus stop with entered information and generated bus stop code [Exception 1, 2, 3, 4, 5]. |   **Alternative Scenario.**   |  |  |  | | --- | --- | --- | | No. | Actor Action | System Response | | 1 | Staff enter Latitude, Longitude | Display bus stop address in Address textbox and show location on map. | | 2 | Staff show bus stop location on map | Display bus stop address in Address textbox, latitude and longitude in Latitude and Longitude inputs. | | 3 | Staff clicks “Cancel” | Redirect to Bus Stop Management page. |   **Exceptions**.   |  |  |  | | --- | --- | --- | | No. | Actor Action | System Response | | 1 | Distance between bus stops is lower than 10m. | Display error message “Duplicate bus stop”. | | 2 | Address’ length is less than 5 | Display error message “Address is too short”. | | 3 | Address’ length is more than 255 | Display error message “Address is too long”. | | 4 | Latitude is less than -180.0 or greater than 180.0 | Display error message “Invalid latitude. It should be in range [-180.0, 180.0]” | | 5 | Longitude is less than -180.0 or greater than 180.0 | Display error message “Invalid longitude. It should be in range [-180.0, 180.0]” |   Relationships: N/A  Business Rules: Google Map API, Latitude, Longitude domain value is [-180, 180] | | | | |

Table 14: Use Case ST004 - Add a new Bus Stop

##### Edit a Bus Stop

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| USE CASE–ST005 | | | | |
| Use-case No. | ST005 | Use-case Version | | 0.1 |
| Use-case Name | Edit a Bus Stop | | | |
| Author | Nguyen Phan Quang Nhat, Doan Xuan Quang, Nguyen Viet Vinh | | | |
| Date | 23/01/2013 | Priority | High | |
| **Actor:** Staff  **Summary:** This use case allows staff to update information of a bus stop.  **Goal:** Change information of an existing bus stop.  **Triggers:** Staff clicks “Edit” associated with a bus stop in Bus Stop Management Page.  **Pre-conditions:** User has logged in as a staff.  **Post-conditions:**  **Success:** The information of edited bus stop is updated.  **Fail:** The information of edited bus stop is not updated. Display error message.  **Main Success Scenario.**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Click “Edit” associated with a bus stop in Bus Stop Management page. | Display bus stop information page includes:   * Code: disabled textbox * Address: textbox (min length 5, max length 255) * Latitude: number (min: -180.0, max 180.0) * Longitude: number (min: -180.0, max: 180.0) * Save: button * Cancel: button * A map to show bus stop location | | 2 | Staff modify account information |  | | 3 | Click “Cancel” [Alternative 1] | Valid entered information and update bus stop information [Exception 1, 2, 3, 4, 5]. |   **Alternative Scenario.**   |  |  |  | | --- | --- | --- | | No. | Actor Action | System Response | | 1 | Staff clicks “Cancel” | Redirect to Bus Stop Management page. |   **Exceptions**.   |  |  |  | | --- | --- | --- | | No. | Actor Action | System Response | | 1 | Distance between bus stops is lower than 10m. | Display error message “Duplicate bus stop”. | | 2 | Address’ length is less than 5 | Display error message “Address is too short”. | | 3 | Address’ length is more than 255 | Display error message “Address is too long”. | | 4 | Latitude is less than -180.0 or greater than 180.0 | Display error message “Invalid latitude. It should be in range [-180.0, 180.0]” | | 5 | Longitude is less than -180.0 or greater than 180.0 | Display error message “Invalid longitude. It should be in range [-180.0, 180.0]” |   Relationships: N/A  Business Rules: Google Map API, Latitude, Longitude domain value is [-180, 180] | | | | |

Table 15: Use Case ST005 - Edit a Bus Stop

##### Delete a Bus Stop

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| USE CASE–ST006 | | | | |
| Use-case No. | ST006 | Use-case Version | | 0.1 |
| Use-case Name | Delete a Bus Stop | | | |
| Author | Nguyen Phan Quang Nhat, Doan Xuan Quang, Nguyen Viet Vinh | | | |
| Date | 23/01/2013 | Priority | High | |
| **Actor:** Staff  **Summary:** This use case allows staff to delete a bus stop.  **Goal:** Remove a bus stop out of the system.  **Triggers:** Staff clicks “Delete” associated with a bus stop in Bus Stop Management Page.  **Pre-conditions:** User has logged in as a staff.  **Post-conditions:**  **Success:** The information of selected bus stop is deleted.  **Fail:** None of bus stop is deleted. Display error message.  **Main Success Scenario.**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Click “Delete” associated with a bus stop in Bus Stop Management page. | Display a confirm message. | | 2 | Click “Yes” [Alternative 1] | Remove the selected bus stop out of the system [Exception 1]. |   **Alternative Scenario.**   |  |  |  | | --- | --- | --- | | No. | Actor Action | System Response | | 1 | Staff clicks “Cancel” | Redirect to Bus Stop Management page. |   **Exceptions**.   |  |  |  | | --- | --- | --- | | No. | Actor Action | System Response | | 1 | The selected bus stop belongs to one or more bus routes. | Display error message “Cannot delete the selected bus stop because it is in use”. |   Relationships: N/A  Business Rules: The deleted bus stop must not be in any bus route. | | | | |

Table 16: Use Case ST006 - Delete a Bus Stop

#### Manage Bus Routes

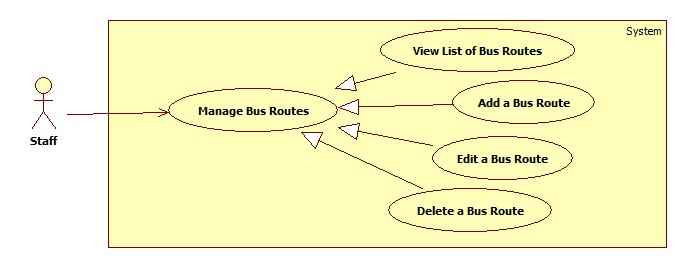


Figure 8: Manage Bus Routes

##### View list of Bus Routes

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| USE CASE–ST007 | | | | |
| Use-case No. | ST007 | Use-case Version | | 0.1 |
| Use-case Name | View List of Bus Routes | | | |
| Author | Nguyen Phan Quang Nhat, Doan Xuan Quang, Nguyen Viet Vinh | | | |
| Date | 23/01/2013 | Priority | High | |
| **Actor:** Staff  **Summary:** This use case allows staff to view the list of bus routes.  **Goal:** List of bus routes in system is displayed.  **Triggers:** Staff access Bus Routes Management Page.  **Pre-conditions:** User has logged in as a staff.  **Post-conditions:**  **Success:** List of bus routes in system is displayed.  **Fail:** Redirect to General Error Page.  **Main Success Scenario.**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Enter the Bus Routes Management page | Display the list of bus routes in system and their information includes:   * Add new bus route: button * A table with these column   + No: label   + Code: label   + Name: label   + Outward: label   + Edit: button   + Delete: button * A Map to display bus route. |   **Alternative Scenario.**  N/A  **Exceptions**.  N/A  Relationships: N/A  Business Rules: Google Map API | | | | |

Table 17: Use Case ST007 - View list of Bus Routes

##### Add a new Bus Route

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| USE CASE-ST008 | | | | |
| Use-case No. | ST008 | Use-case Version | | 0.1 |
| Use-case Name | Add a new Bus Route | | | |
| Author | Nguyen Phan Quang Nhat, Nguyen Viet Vinh | | | |
| Date | 23/01/2013 | Priority | High | |
| **Actor:** Staff  **Summary:** This use case allows staff to add a new bus route into system.  **Goal:** Create a new bus route.  **Triggers:** Staff clicks “Add New Bus Route” in Account Management Page.  **Pre-conditions:** User has logged in as a staff.  **Post-conditions:**  **Success:** A new bus route has been created in system.  **Fail:** None of bus route is added to the system. Display error message.  **Main Success Scenario.**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Staff clicks “Add New Bus Route” in Bus Route Management Page. | Display bus route information page includes:   * Code: textbox (min length: 1, max length: 10) * Name: textbox (min length: 1, max length: 100) * Outward: checkbox * Price: Number * List of box stops: table. * Available bus stop: combo box * Add bus stop: button * Move up: button * Move down: button * Remove bus stop: button * Add: button * Cancel: button * A map to show bus route | | 2 | Staff enters bus route information |  | | 3 | Staff edit list of stop of the route | * Update route on map. | | 3 | Staff clicks “Add” [Alternative 1] | Valid entered information and create a new bus route with entered information [Exception 1, 2, 3, 4, 5, 6]. |   **Alternative Scenario.**   |  |  |  | | --- | --- | --- | | No. | Actor Action | System Response | | 1 | Staff clicks “Cancel” | Redirect to Bus Route Management page. |   **Exceptions**.   |  |  |  | | --- | --- | --- | | No. | Actor Action | System Response | | 1 | Staff leaves code empty. | Display error message “Code cannot be empty”. | | 2 | Code’s length is more than 10 | Display error message “Code is too long”. | | 3 | Name’s length is less than 1 | Display error message “Name is too short” | | 4 | Name’s length is more than 100 | Display error message “Name is too long”. | | 5 | There is another bus route with the same value of code and outward. | Display error message “Duplicate bus route”. | | 6 | Entered is less than 1000. | Display error message “Invalid price” |   Relationships: N/A  Business Rules: Google Map API. A Bus Route is identified by its code and outward value. | | | | |

Table 18: Use Case ST008 - Add a new Bus Route

##### Edit a Bus Route

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| USE CASE-ST009 | | | | |
| Use-case No. | ST009 | Use-case Version | | 0.1 |
| Use-case Name | Edit a Bus Route | | | |
| Author | Nguyen Phan Quang Nhat, Nguyen Viet Vinh | | | |
| Date | 23/01/2013 | Priority | High | |
| **Actor:** Staff  **Summary:** This use case allows staff to edit a bus route in system.  **Goal:** Update information of a bus route.  **Triggers:** Click “Edit” associated with a bus stop in Bus Route Management page.  **Pre-conditions:** User has logged in as a staff.  **Post-conditions:**  **Success:** The information of edited bus route is updated.  **Fail:** The information of edited bus route is not updated. Display error message.  **Main Success Scenario.**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Click “Edit” associated with a bus stop in Bus Route Management page. | Display bus route information page includes:   * Code: disabled textbox * Name: textbox (min length: 1, max length: 100) * Outward: checkbox * Price: Number (min value: 0) * List of stops: table. * Available bus stop: combo box * Add bus stop: button * Move up: button * Move down: button * Remove bus stop: button * Save: button * Cancel: button * A map to show bus route | | 2 | Staff modify bus route information |  | | 3 | Staff edit list of stop of the route | * Update route on map. | | 3 | Staff clicks “Save” [Alternative 1] | Valid entered information and update information of the bus [Exception 1, 2, 3, 4]. |   **Alternative Scenario.**   |  |  |  | | --- | --- | --- | | No. | Actor Action | System Response | | 1 | Staff clicks “Cancel” | Redirect to Bus Route Management page. |   **Exceptions**.   |  |  |  | | --- | --- | --- | | No. | Actor Action | System Response | | 1 | Name’s length is less than 1 | Display error message “Name is too short” | | 2 | Name’s length is more than 100 | Display error message “Name is too long”. | | 3 | There is another bus route with the same value of code and outward. | Display error message “Duplicate bus route”. | | 4 | Entered price is less than 1000 | Display error message “Invalid price”. |   Relationships: N/A  Business Rules: Google Map API. A Bus Route is identified by its code and outward value. | | | | |

Table 19: Use Case ST009 - Edit a Bus Route

##### Delete a Bus Route

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| USE CASE–ST010 | | | | |
| Use-case No. | ST010 | Use-case Version | | 0.1 |
| Use-case Name | Delete a Bus Route | | | |
| Author | Nguyen Phan Quang Nhat, Nguyen Viet Vinh | | | |
| Date | 23/01/2013 | Priority | High | |
| **Actor:** Staff  **Summary:** This use case allows staff to delete a bus route.  **Goal:** Remove a bus route out of the system.  **Triggers:** Staff clicks “Delete” associated with a bus route in Bus Route Management Page.  **Pre-conditions:** User has logged in as a staff.  **Post-conditions:**  **Success:** The information of selected bus route is deleted.  **Fail:** None of bus route is deleted. Display error message.  **Main Success Scenario.**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Click “Delete” associated with a bus route in Bus Route Management page. | Display a confirm message. | | 2 | Click “Yes” [Alternative 1] | Remove the selected bus route out of the system and its associated schedule. |   **Alternative Scenario.**   |  |  |  | | --- | --- | --- | | No. | Actor Action | System Response | | 1 | Staff clicks “Cancel” | Redirect to Bus Route Management page. |   **Exceptions**.  N/A  Relationships: N/A  Business Rules: N/A | | | | |

Table 20: Use Case ST010 - Delete a Bus Route

#### Manage Bus Schedules

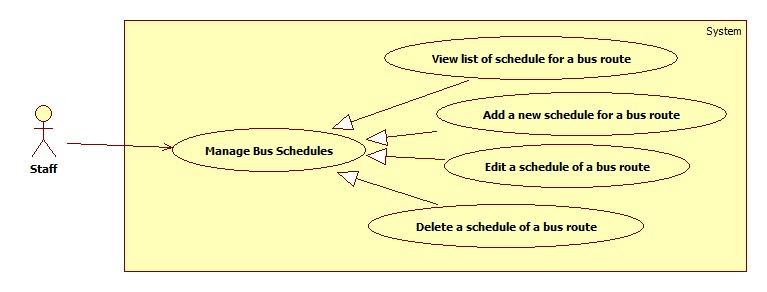


Figure 9: Manage Bus Schedules

##### View list of Schedule for a Bus Route

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| USE CASE–ST011 | | | | |
| Use-case No. | ST011 | Use-case Version | | 0.1 |
| Use-case Name | View List of Schedules for a bus route | | | |
| Author | Nguyen Phan Quang Nhat, Nguyen Viet Vinh | | | |
| Date | 23/01/2013 | Priority | High | |
| **Actor:** Staff  **Summary:** This use case allows staff to view the list of schedules for a bus route.  **Goal:** List of schedules for a bus route in system is displayed.  **Triggers:** Staff access Schedule Management Page.  **Pre-conditions:** User has logged in as a staff.  **Post-conditions:**  **Success:** List of schedule for a bus route in system is displayed.  **Fail:** Redirect to General Error Page.  **Main Success Scenario.**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Enter the Schedule Management page | * Display the list of bus route with these information:   + Code   + Name   + Outward | | 2 | Staff choose a bus route from the list | * Display the list of week days. * A Date picker (default: current date) | | 3 | Staff chooses a week day or select specific date from Data picker. | * Display the list of departure time of selected bus route in selected date. |   **Alternative Scenario.**  N/A  **Exceptions**.  N/A  Relationships: N/A  Business Rules: N/A | | | | |

Table 21: Use Case ST011 - View list of Schedule for a Bus Route

##### Add a new Schedule for a Bus Route

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| USE CASE–ST012 | | | | |
| Use-case No. | ST012 | Use-case Version | | 0.1 |
| Use-case Name | Add a new schedule for a bus route | | | |
| Author | Nguyen Phan Quang Nhat, Nguyen Viet Vinh | | | |
| Date | 23/01/2013 | Priority | High | |
| **Actor:** Staff  **Summary:** This use case allows staff to add a new schedule for a bus route.  **Goal:** Add a new schedule for a bus route.  **Triggers:** Staff access Schedule Management Page and click “Add” in the schedule list.  **Pre-conditions:** User has logged in as a staff.  **Post-conditions:**  **Success:** A new schedule is added for a bus route.  **Fail:** None of schedule is added. Display error message.  **Main Success Scenario.**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Enter the Schedule Management page | * Display the list of bus route with these information:   + Code   + Name   + Outward | | 2 | Staff choose a bus route from the list | * Display the list of week days. * A Date picker (default: current date). | | 3 | Staff chooses a week day or select specific date from Data picker. | * Display the list of departure time of selected bus route in selected date. | | 4 | Click “Add Schedule” button | * Display a dialog:   + Departure time: time picker (min: 4:00, max 22:00).   + Add: button   + Cancel: button | | 5 | Staff selects departure time and clicks “Add” [Alternative 1] | * Valid information and add a new schedule for the selected bus route in selected date [Exception 1, 2]. |   **Alternative Scenario.**   |  |  |  | | --- | --- | --- | | No. | Actor Action | System Response | | 1 | Staff clicks “Cancel” | Dialog disappears without any affect. |   **Exceptions**.   |  |  |  | | --- | --- | --- | | No. | Actor Action | System Response | | 1 | There is an existing schedule with the same time in the same date for the same bus route. | Display error message “The departure time already exists”. | | 2 | Departure time is less than 4:00 or greater than 22:00 | Display error message “Invalid departure time”. |   Relationships: N/A  Business Rules: Departure must be from 4:00 to 22:00 | | | | |

Table 22: Use Case ST012 - Add a new Schedule for a Bus Route

##### Edit a Schedule of a Bus Route

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| USE CASE–ST013 | | | | |
| Use-case No. | ST013 | Use-case Version | | 0.1 |
| Use-case Name | Edit a schedule for a bus route | | | |
| Author | Nguyen Phan Quang Nhat, Nguyen Viet Vinh | | | |
| Date | 23/01/2013 | Priority | High | |
| **Actor:** Staff  **Summary:** This use case allows staff to edit a schedule of a bus route.  **Goal:** Update a schedule of a bus route.  **Triggers:** Staff access Schedule Management Page and double click on a schedule in the list.  **Pre-conditions:** User has logged in as a staff.  **Post-conditions:**  **Success:** The edited schedule is update for the bus route.  **Fail:** None of schedule is updated. Display error message.  **Main Success Scenario.**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Enter the Schedule Management page | * Display the list of bus route with these information:   + Code   + Name   + Outward | | 2 | Staff choose a bus route from the list | * Display the list of week days. * A Date picker (default: current date). | | 3 | Staff chooses a week day or select specific date from Data picker. | * Display the list of departure time of selected bus route in selected date. | | 4 | Double click on a schedule in the list | * Display a dialog:   + Departure time: time picker (min: 4:00, max: 22:00).   + Save: button   + Cancel: button | | 5 | Staff selects departure time and clicks “Save” [Alternative 1] | * Valid information and update the schedule for the selected bus route in selected date [Exception 1]. |   **Alternative Scenario.**   |  |  |  | | --- | --- | --- | | No. | Actor Action | System Response | | 1 | Staff clicks “Cancel” | Dialog disappears without any affect. |   **Exceptions**.   |  |  |  | | --- | --- | --- | | No. | Actor Action | System Response | | 1 | There is an existing schedule with the same time in the same date for the same bus route. | Display error message “The departure time already exists”. | | 2 | Departure time is less than 4:00 or greater than 22:00 | Display error message “Invalid departure time”. |   Relationships: N/A  Business Rules: Departure must be between 4:00 and 22:00 | | | | |

Table 23: Use Case ST013 - Edit a Schedule for a Bus Route

##### Delete a Schedule of a Bus Route

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| USE CASE–ST014 | | | | |
| Use-case No. | ST014 | Use-case Version | | 0.1 |
| Use-case Name | Delete a schedule of a bus route | | | |
| Author | Nguyen Phan Quang Nhat, Nguyen Viet Vinh | | | |
| Date | 23/01/2013 | Priority | High | |
| **Actor:** Staff  **Summary:** This use case allows staff to delete a schedule of a bus route.  **Goal:** Delete a schedule of a bus route.  **Triggers:** Staff access Schedule Management Page and click “x” button associated with a schedule in the list.  **Pre-conditions:** User has logged in as a staff.  **Post-conditions:**  **Success:** The selected schedule is removed.  **Fail:** None of schedule is removed. Display error message.  **Main Success Scenario.**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Enter the Schedule Management page | * Display the list of bus route with these information:   + Code   + Name   + Outward | | 2 | Staff choose a bus route from the list | * Display the list of week days. * A Date picker (min, max, default). | | 3 | Staff chooses a week day or select specific date from Data picker. | * Display the list of departure time of selected bus route in selected date. | | 4 | Click on “x” button associated with a schedule in the list | * Display a confirm dialog | | 5 | Staff clicks “Yes” [Alternative 1] | * Remove the selected schedule out of system. |   **Alternative Scenario.**   |  |  |  | | --- | --- | --- | | No. | Actor Action | System Response | | 1 | Staff clicks “Cancel” | Dialog disappears without any affect. |   **Exceptions**.  N/A  Relationships: N/A  Business Rules: N/A | | | | |

Table 24: Use Case ST014 - Delete a Schedule of a Bus Route

### Software System Attributes

#### Usability

##### Graphic User Interface

* + All the text, label and image in public user page should be in Vietnamese.
  + All the text, label and image in Admin page, Staff page should be in English.

##### Usability for public user

* + Can use some public function without login.
  + Can use their android devices to access service easily.

##### Usability for admin and staff

* + Admin and staff must understand and use English.
  + Website admin and staff should need no more than one week of training to be productive with the system.

##### Installation

* + The system must 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. User can read and do itself without developers’ help.

#### Reliability

N/A

#### Availability

N/A

#### Security

* + Privacy: Each role of user has a specific permission to interact with system. System always checks authorization and authenticated before do anything. Only admin can grant permission to users.

#### Maintainability

#### Portability

#### Performance

## Main Flows

### Public User

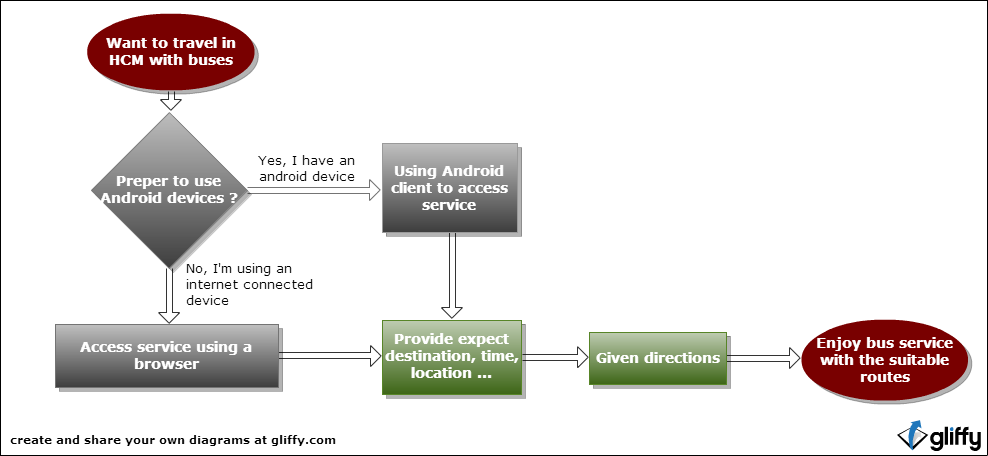


Figure 10: Public User Main Flow

### Staff

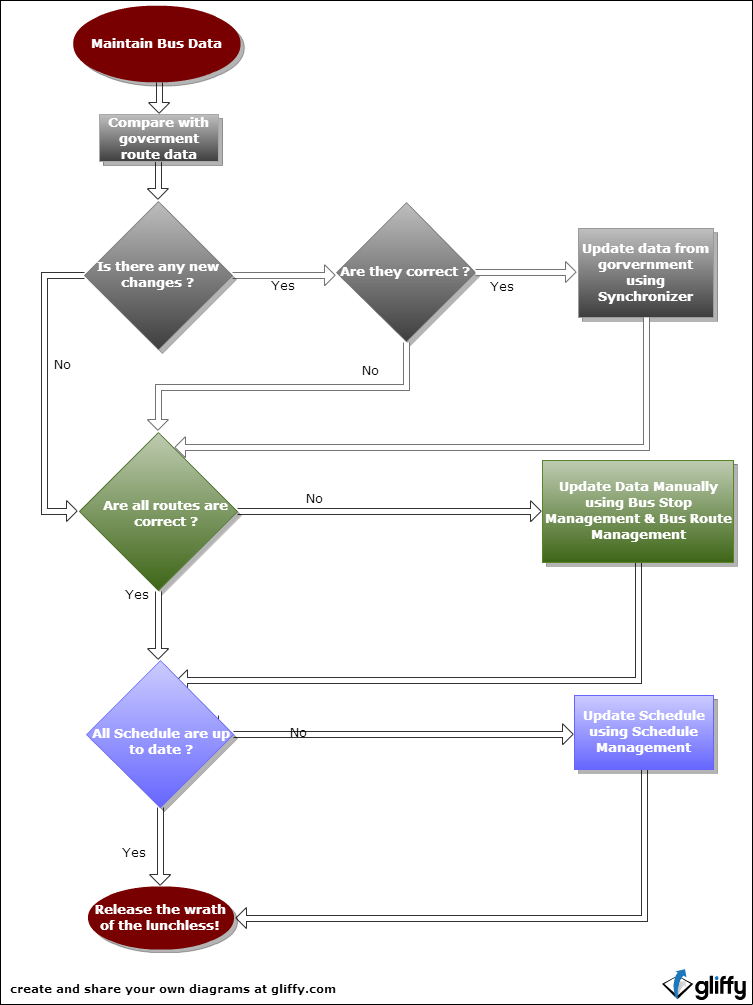


Figure 11: Staff Main Flow

## Entity Relationship - Conceptual

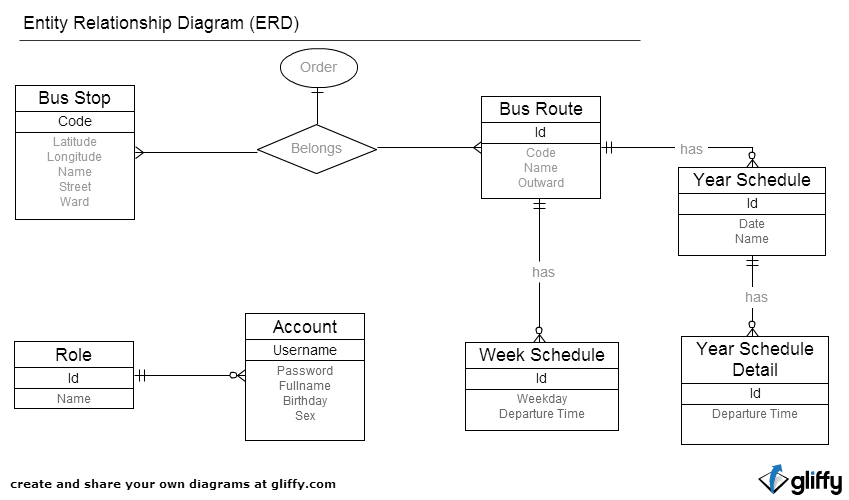


Figure 12: Entity Relationship

# Software Design Description (SDD)

## Design Overview

This part provides for project’s stakeholders multi specific views of BUSSUG system like:

* + System architecture design which describes the planned model of system, its’ layers, and each layer function
  + Component design which describes all system’s component (class, interface…) together with relationships between them. It also explains clearly their purposes, methods within each component and detailed algorithms, pseudo code to implement them.
  + Sequence diagram design which shows how system processes operate and in what order. Besides, this section depicts the objects which contained in each scenario and theirs interaction by transferred messages.
  + User interface design which describes all screens’ interface; each screen includes which types of input, output element and used by what action.
  + Database design section shows and explains the physical model of system database which includes all tables and their relationship

## System Architectural Design

### Choice of System Architecture



Figure 13: MVC Architecture[[8]](#endnote-8)

### Discussion of Alternative Designs

#### MVP (Model-View-Presenter)

N/A

#### MVVM (Model-View-View-Model)

N/A

### Description of System Interface

N/A

## Component Diagram

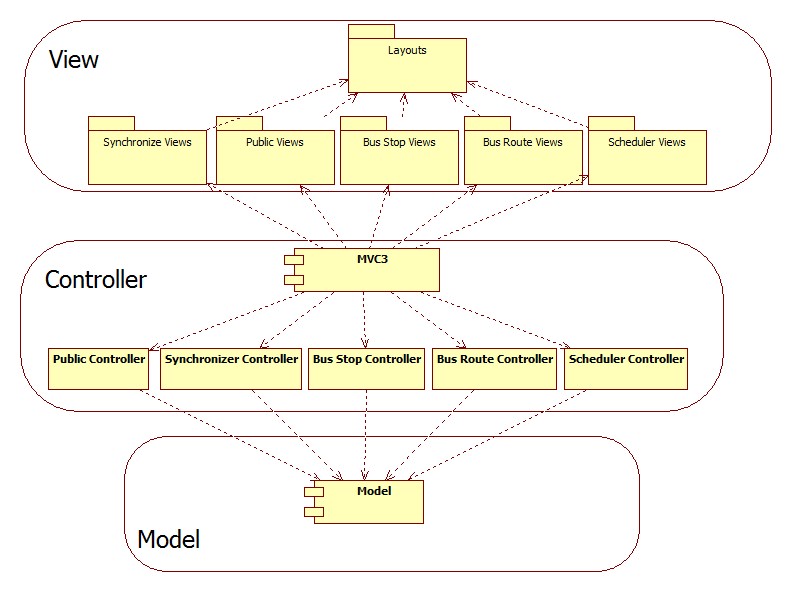


Figure 14: Components Overview

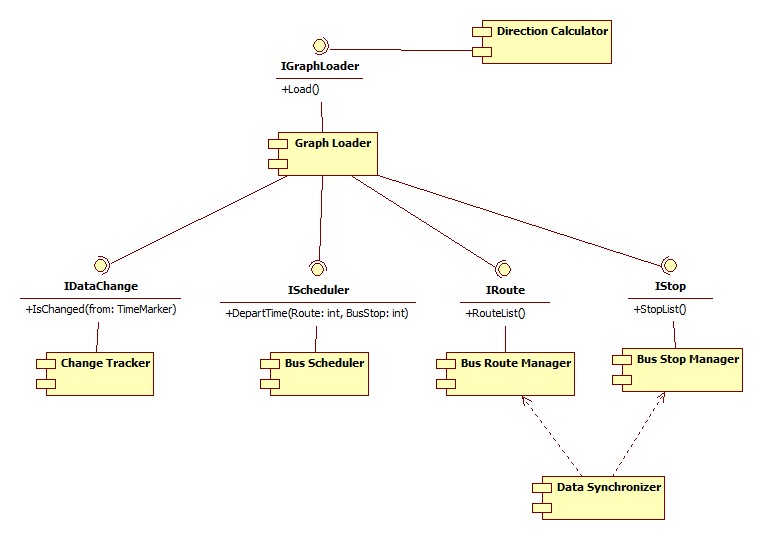


Figure 15: Model Main Components

## Detailed Description of Components

### Graph Loader & Direction Calculator

#### Class Diagram

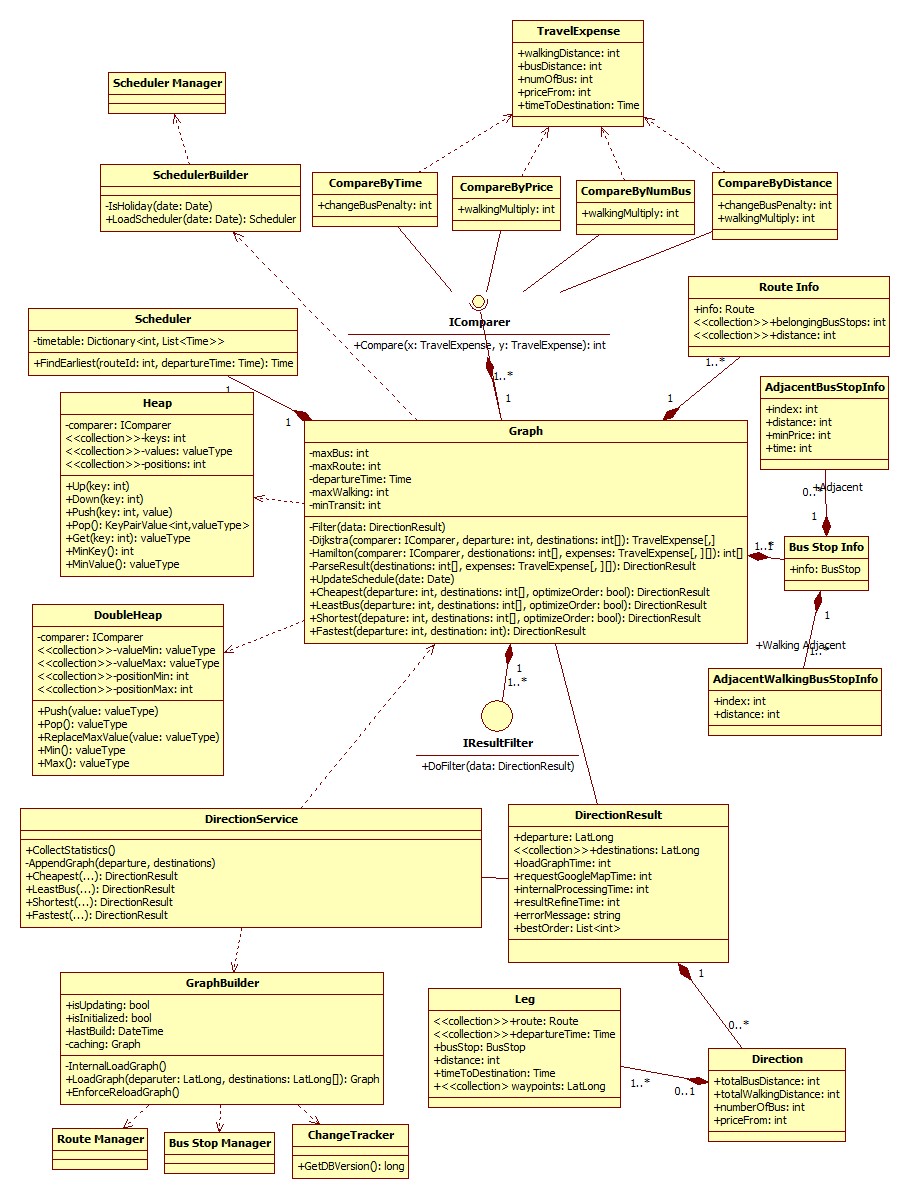


Table 25: Graph Builder & Direction Service

#### Class Diagram Explanation

##### TravelExpense

This class contains information of expenses used to travel from one node to another in graph.

###### Attributes

|  |  |  |
| --- | --- | --- |
| **Name** | **Type** | **Description** |
| walkingDistance | int | The total walking distance. |
| busDistance | int | The total distance traveled by buses. |
| numOfBus | int | The number of buses used. |
| priceFrom | int | The minimum price. |
| timeToDestination | Time | The earliest point to time to reach destination. |

##### Graph

This class contains all information about the city map buses. These information are used to perform searching algorithms finding out suitable directions depends on different criteria such as: time, number of bus, price, distance, walking distance, transit time, departure time…

###### Attributes

|  |  |  |
| --- | --- | --- |
| **Name** | **Type** | **Description** |
| maxBus | int | The maximum number of bus to be used for travelling from departure to one destination (is not used for FewestBus) |
| maxRoute | int | The maximum number of alternative results. |
| departureTime | Time | The departure time (is used for Fastest) |
| maxWalking | int | The maximum walking distance between bus stops transit in result. Except from the departure to the first bus stop and from the last bus stop to the destination. |
| minTransit | int | The minimum time for traveler changing buses. |
| routes | <<collection>>RouteInfo | The information of all bus routes. |
| busStops | <<collection>>BusStopInfo | The information of all bus stops. |
| scheduler | <<collection>>Scheduler | The information of departure time of all bus routes. |
| filters | <<collection>>IResultFilter | Rules to filter out non-human results. |

###### Methods

- Dijkstra(comparer: IComparer, departure: int, destinations: int[]): TravelExpense[,]

This method implements the generic modified version of Dijkstra algorithms. This allows finding out **maxRoute** shortest paths from one **departure** to all **destinations** in the graph. The term shortest is defined by the **comparer**. The compare specifies which criterion should be optimized: time, distance, price, and number of bus used or a function combining some of them.

The produced result must satisfy criteria of maxBus (maximum of bus used), maxWalking (maximum walking distance between transits) and minTransit (minimum time for changing buses).

|  |  |  |
| --- | --- | --- |
| **Parameter/Return** | **Type** | **Description** |
| comparer | IComparer | The concrete implementation of IComparer which describes the optimizing criterion. |
| departure | int | The index of departure node in graph. |
| destinations | int[] | The indexes of destination nodes in graph that we focus on. |
| Return | TravelExpense[,] | The expense matrix. [x,k] is the expense the kth shortest path to travel from departure node to node x. |

-Hamilton(destinations: int[], expenses: TravelExpense[,][]):int[]

This method find out which order of travelling will give the best expense.

|  |  |  |
| --- | --- | --- |
| **Parameter/Return** | **Type** | **Description** |
| comparer | IComparer | The concrete implementation of IComparer which describes the optimizing criterion. |
| destinations | int[] | The indexes of destination nodes in graph that we focus on. |
| expenses | TravelExpense[,][] | List of matrix expense corresponding to each destination. |
| Return | int[] | The order of travelling that give best expense. |

ParseResult(destinations:int[], expenses:TravelExpense[,][]):DirectionResult

The method converts expense matrixes to DirectionResult object.

|  |  |  |
| --- | --- | --- |
| **Parameter/Return** | **Type** | **Description** |
| destinations | int[] | The indexes of destination nodes in graph that we focus on. |
| expenses | TravelExpense[,][] | List of matrix expense corresponding to each destination. |
| Return | DirectionResult | The DirectionResult object containing information only related to the final result. |

Filter(data: DirectionResult)

The method filter out directions those are not reasonable to human.

|  |  |  |
| --- | --- | --- |
| **Parameter/Return** | **Type** | **Description** |
| data | DirectionResult | The result produced from base algorithms. |

UpdateSchedule(date: Date)

Reload schedules of all routes for specified date.

|  |  |  |
| --- | --- | --- |
| **Parameter/Return** | **Type** | **Description** |
| date | Date | The operating date. |

Cheapest(departure: int, destinations: int[], optimizeOrder: bool): DirectionResult

Find cheapest directions to travel from departure to all destinations.

|  |  |  |
| --- | --- | --- |
| **Parameter/Return** | **Type** | **Description** |
| departure | int | The index of departure node in graph. |
| destinations | int[] | The indexes of destination nodes in graph that we concern. |
| optimizeOrder | bool | Indicate that should we compute the best travelling order. |
| Return | DirectionResult | The directions to travel that are cheapest and satisfy criteria. |

LeastBus(departure: int, destinations: int[], optimizeOrder: bool): DirectionResult

Find directions to travel from departure to all destinations that use fewest number of buses.

|  |  |  |
| --- | --- | --- |
| **Parameter/Return** | **Type** | **Description** |
| departure | int | The index of departure node in graph. |
| destinations | int[] | The indexes of destination nodes in graph that we concern. |
| optimizeOrder | bool | Indicate that should we compute the best travelling order. |
| Return | DirectionResult | The directions to travel that use fewest buses and satisfy criteria. |

Shortest(depature: int, destinations: int[], optimizeOrder: bool): DirectionResult

Find shortest (in term of distance) directions to travel from departure to all destinations.

|  |  |  |
| --- | --- | --- |
| **Parameter/Return** | **Type** | **Description** |
| departure | int | The index of departure node in graph. |
| destinations | int[] | The indexes of destination nodes in graph that we concern. |
| optimizeOrder | bool | Indicate that should we compute the best travelling order. |
| Return | DirectionResult | The directions to travel that use fewest buses and satisfy criteria. |

Fastest(departure: int, destination: int): DirectionResult

Find fastest directions to travel from departure to all destinations.

|  |  |  |
| --- | --- | --- |
| **Parameter/Return** | **Type** | **Description** |
| departure | int | The index of departure node in graph. |
| destinations | int[] | The indexes of destination nodes in graph that we concern. |
| optimizeOrder | bool | Indicate that should we compute the best travelling order. |
| Return | DirectionResult | The directions to travel that are shortest in term of distance and satisfy criteria. |

##### DirectionService

This class wraps and exposes the main functions of Graph class to outside world as services. It is also responsible for collecting statistics on using services such as executing time.

##### CompareByTime

This class implements IComparer that express time as the optimizing criterion using this function:

f(x) = x.Time + x.NumberOfBus \* changeBusPenalty

where changeBusPenalty is a constant. We use it because people tend to avoid changing bus even it takes a little longer.

##### CompareByPrice

This class implements IComparer that express price as the optimizing criterion using this function:

f(x) = x.MinPrice \* MAX\_DISTANCE + x.WalkingDistance \* walkingMultiply + x.BusDistance

where MAX\_DISTANCE is a constant represent the very long distance that never happens in real situation. It means that we optimize price first, then distance later.

We use walkingMultiply constant here because people tend to avoid walking.

##### CompareByNumOfBus

This class implements IComparer that express the number of buses used as the optimizing criterion using this function:

f(x) = x. NumberOfBus \* MAX\_DISTANCE + x.WalkingDistance \* walkingMultiply + x.BusDistance

where MAX\_DISTANCE is a constant represent the very long distance that never happens in real situation. It means that we optimize number of bus first, then distance later.

We use walkingMultiply constant here because people tend to avoid walking.

##### CompareByDistance

This class implements IComparer that express distance as the optimizing criterion using this function:

f(x) = x.BusDistance + x.WalkingDistance \* walkingMultiply + x.NumberOfBus \* changeBusPenalty

where walkingMultiply and changeBusPenalty are constants. We use them because people tend to avoid walking as well as changing buses.

##### RouteInfo

This class contains information of a route in graph. Its structure differs from the underlying storage (database structure) to support graph operations.

|  |  |  |
| --- | --- | --- |
| **Name** | **Type** | **Description** |
| info | Route | General information of route. |
| belongingBusStops | <<collection>> int | The indexes of bus stops on the route. |
| distance | <<collection>> int | Corresponding distance of bus stops from the first one. |

##### BusStopInfo

This class contains information of a bus stop in graph. Its structure differs from the underlying storage (database structure) to support graph operations.

|  |  |  |
| --- | --- | --- |
| **Name** | **Type** | **Description** |
| info | BusStop | General information of bus stop. |
| AdjacentBusStops | <<collection>> AdjacentBusStopInfo | Bus stops that are connected directly by a bus route. |
| AdjacentWalkingBusStops | <<collection>> AdjacentWalkingBusStopInfo | Bus stops that are connected directly by a short walking distance (<= 1000m) |

##### AdjacentBusStopInfo

|  |  |  |
| --- | --- | --- |
| **Name** | **Type** | **Description** |
| index | int | Index in graph |
| distance | int | Travelling distance by bus |
| minPrice | int | Minimum price |
| time | int | Time (in minutes) |

##### AdjacentWalkingBusStopInfo

|  |  |  |
| --- | --- | --- |
| **Name** | **Type** | **Description** |
| index | int | Index in graph |
| distance | int | Travelling distance by walking |

##### Scheduler

This class contains information of departure time of all bus routes. Its structure differs from underlying storage (database structure) to support fast searching required by graph operations.

###### Attributes

|  |  |  |
| --- | --- | --- |
| **Name** | **Type** | **Description** |
| timetable | Dictionary<int, List<Time>> | List of departure times corresponding to each route id. |

###### Methods

FindEarliest(routeId: int, departureTime: Time): Time

Find the first departure time of a route after a specified point of time.

|  |  |  |
| --- | --- | --- |
| **Parameter/Return** | **Type** | **Description** |
| routeId | int | The id of route. |
| departureTime | Time | The point of time that we cannot depart earlier |
| Return | Time | The first departure time of specified route that is not earlier than specified point of time. |

##### SchedulerBuilder

This class responsible for convert schedule information of specified date in form of underlying structure (database) to memory structure of Scheduler to support fast searching required by graph operations.

###### IsHoliday(date: Date):boolean

Check if a date is a holiday.

###### LoadScheduler(date: Date): Scheduler

Load schedule information of a specified date and convert into memory structure.

##### Heap

This class implements a priority queue that has pre-defined maximum element and support these operations:

* + Get min value in O(1).
  + Push a new value into priority queue in O(lgN).
  + Pop the min value out of queu in O(lgN).
  + Update value of an existing element in queue in O(lgN).

Formal definition of Priority Queue: <http://en.wikipedia.org/wiki/Priority_queue>

Heap Struture: <http://en.wikipedia.org/wiki/Heap_(data_structure)>

##### DoubleHeap

This class implements a double-ended priority queue that has pre-defined maximum element and support these operations:

* + Get min value in O(1).
  + Get max value in O(1).
  + Push a new value into queue in O(lgN).
  + Pop the min value out of queue in O(lgN).
  + Pop the max value out of queue in O(lgN).
  + Update value of an existing element in queue in O(lgN).

Formal definition of Double-ended Priority Queue: <http://en.wikipedia.org/wiki/Double-ended_priority_queue>

We implement this structure using 2 heaps and 2 arrays to track position of each element in each heap.

##### GraphBuilder

This class loads information of all routes and all bus stops from underlying storage (database) and converts them into memory structure that is suitable for graph operations.

##### DirectionResult

This class contains all travelling information needed for a response as well as statistics information.

|  |  |  |
| --- | --- | --- |
| **Name** | **Type** | **Description** |
| departure | Latlong | Coordinate of departure |
| destinations | <<collection>> Latlong | Coordinates of destinations |
| loadGraphTime | int | Time to load graph (in milliseconds) |
| requestGoogleMap | int | Time to request information from Google Map service (in milliseconds) |
| internalProcessingTime | int | Time for internal processing (in milliseconds) |
| resultRefineTime | int | Time for further processing such as filtering (in milliseconds) |
| errorMessage | string | Error message if something wrong happens. |
| directions | <<collection>> Direction | Information to travel with least expenses. Each corresponding to each result. |

##### Direction

This contains travelling information for a single result.

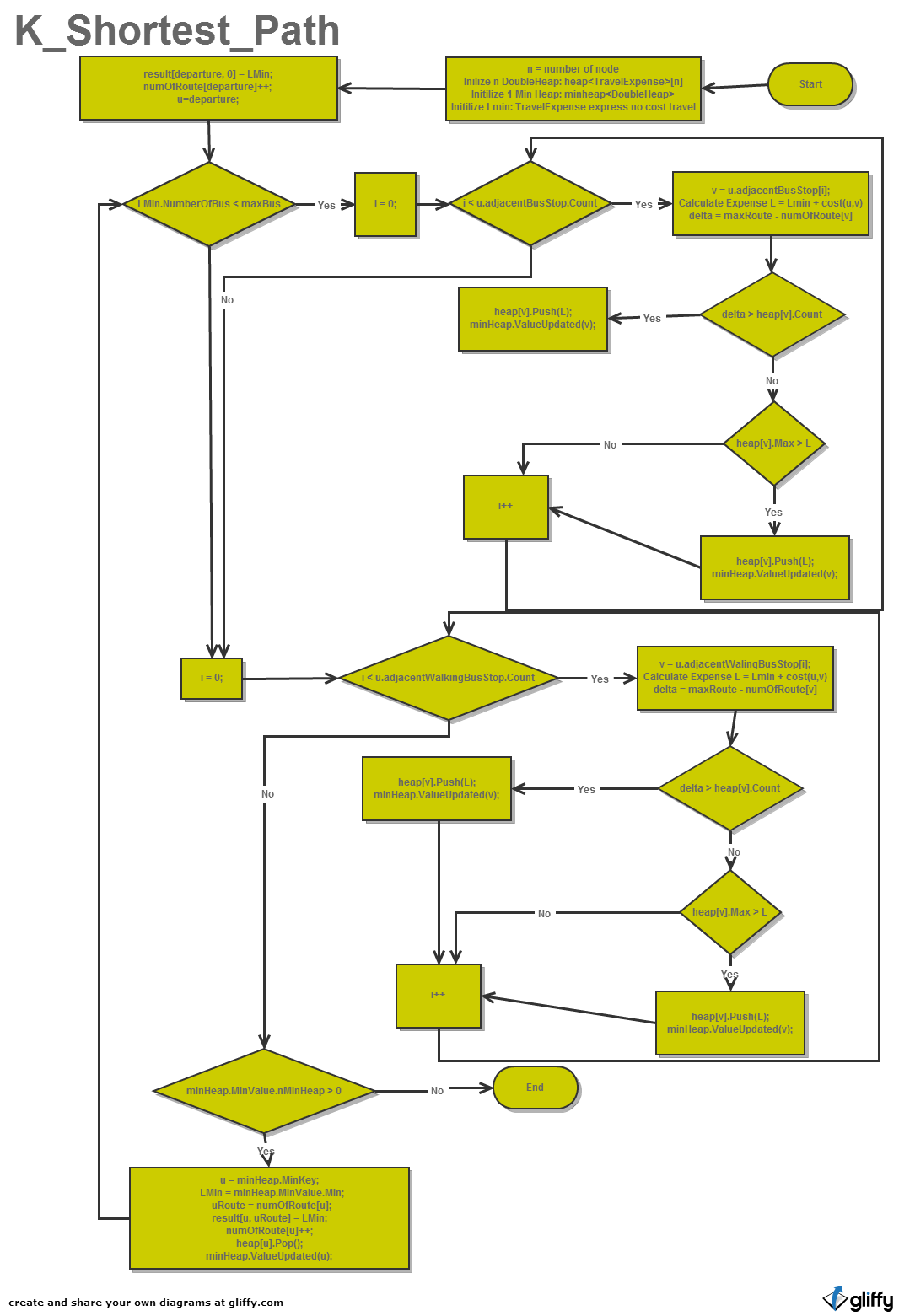
|  |  |  |
| --- | --- | --- |
| **Name** | **Type** | **Description** |
| totalBusDistance | int | Total travelling distance by bus (in meters) |
| totalWalkingDistance | int | Total walking distance (in meters) |
| numOfBus | int | The number of buses used. |
| priceFrom | int | The minimum price. |
| legs | <<collection>> Leg | Information for each travel segment |

##### Leg

|  |  |  |
| --- | --- | --- |
| **Name** | **Type** | **Description** |
| routes | <<collection>>Route | List of bus routes can be used. |
| departureTimes | <<collection>> Time | Corresponding departure time for each bus route. |
| busStop | BusStop | Information of the next bus stop. |
| distance | int | Distance to the next bus stop (in meters) |
| timeToDestination | Time | The next bus stop arrival time. |
| waypoints | <<collection> Latlong | These are used to draw route correctly. |

#### Algorithms of important methods in each class specified in pseudo code or by Flow-Chart

##### Dijkstra



### Data Synchronizer

### Bus Stop Manager

### Bus Route Manager

### Scheduler

### Change Tracker

## Sequence Diagram

## User Interface Design

### Description of the User Interface

#### Screen Images

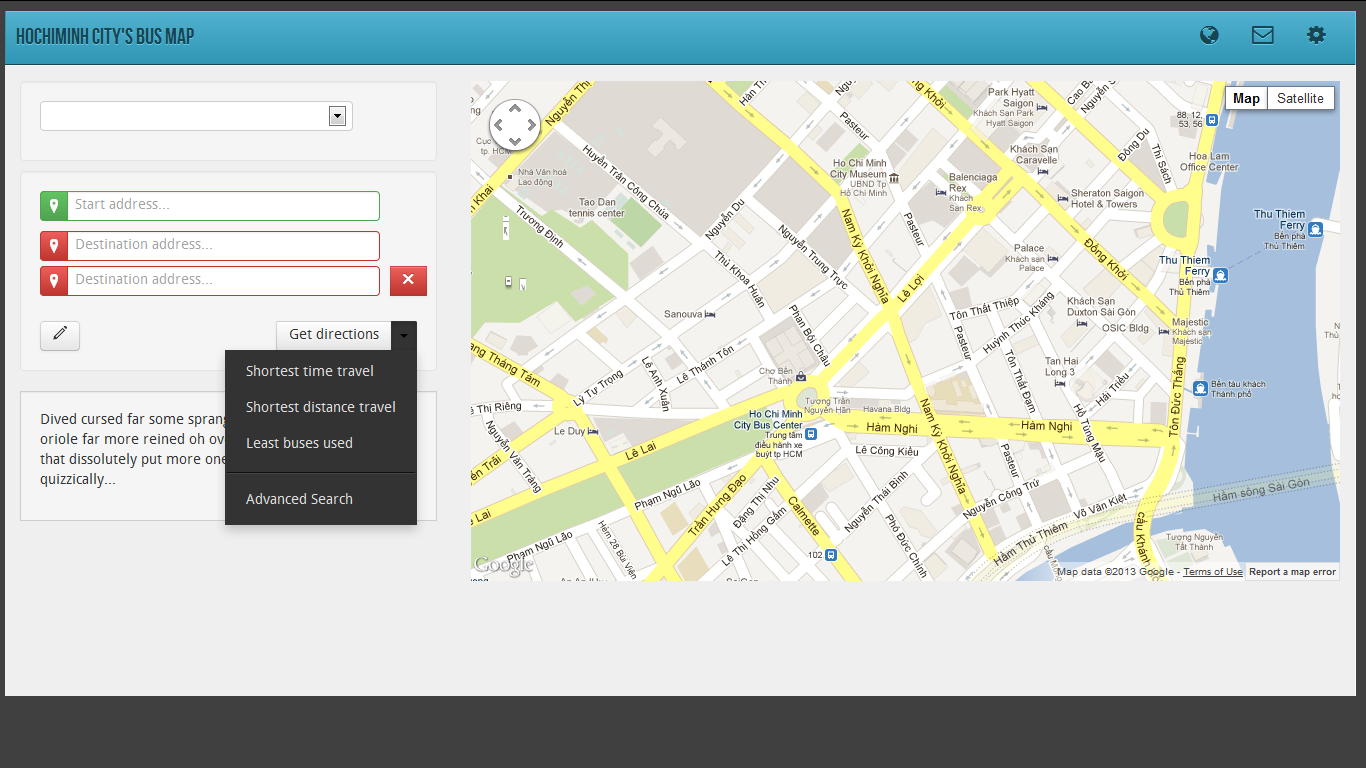
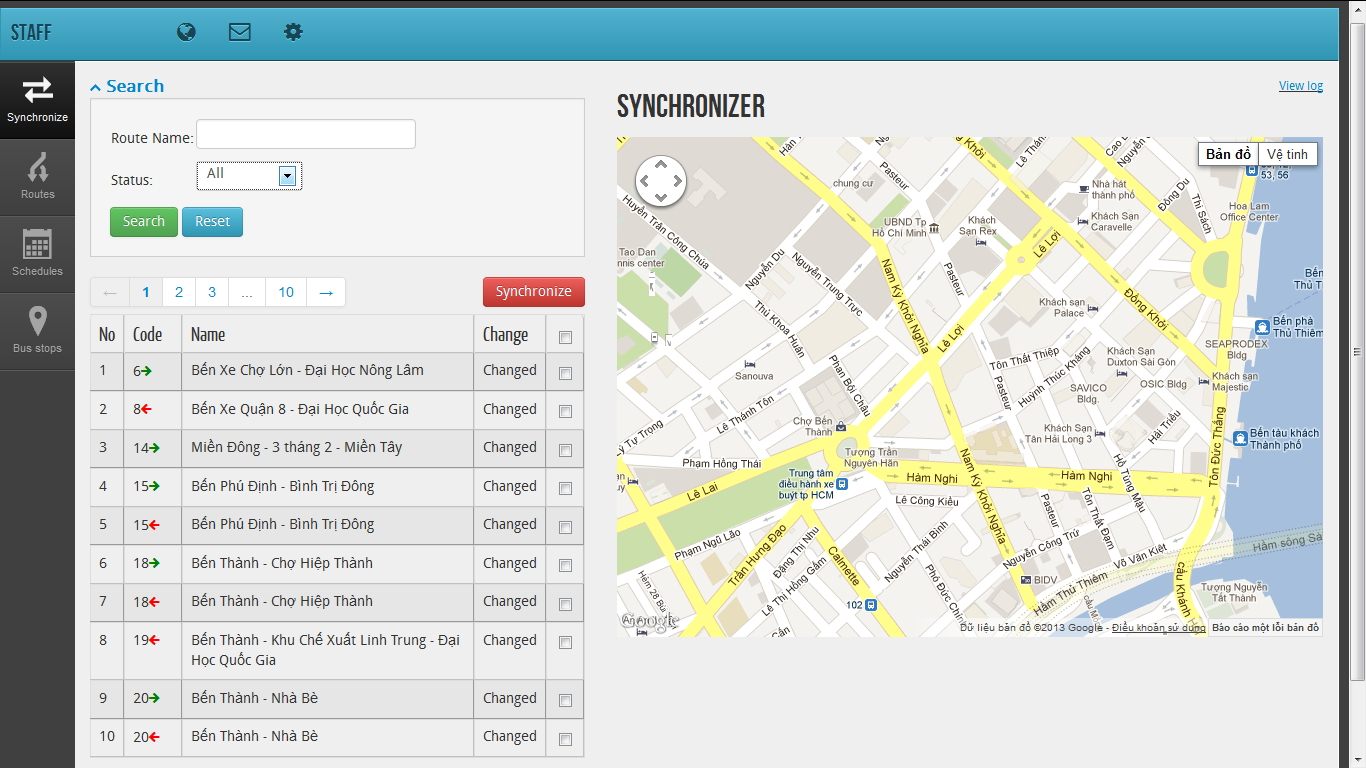


Figure 16: Public user page



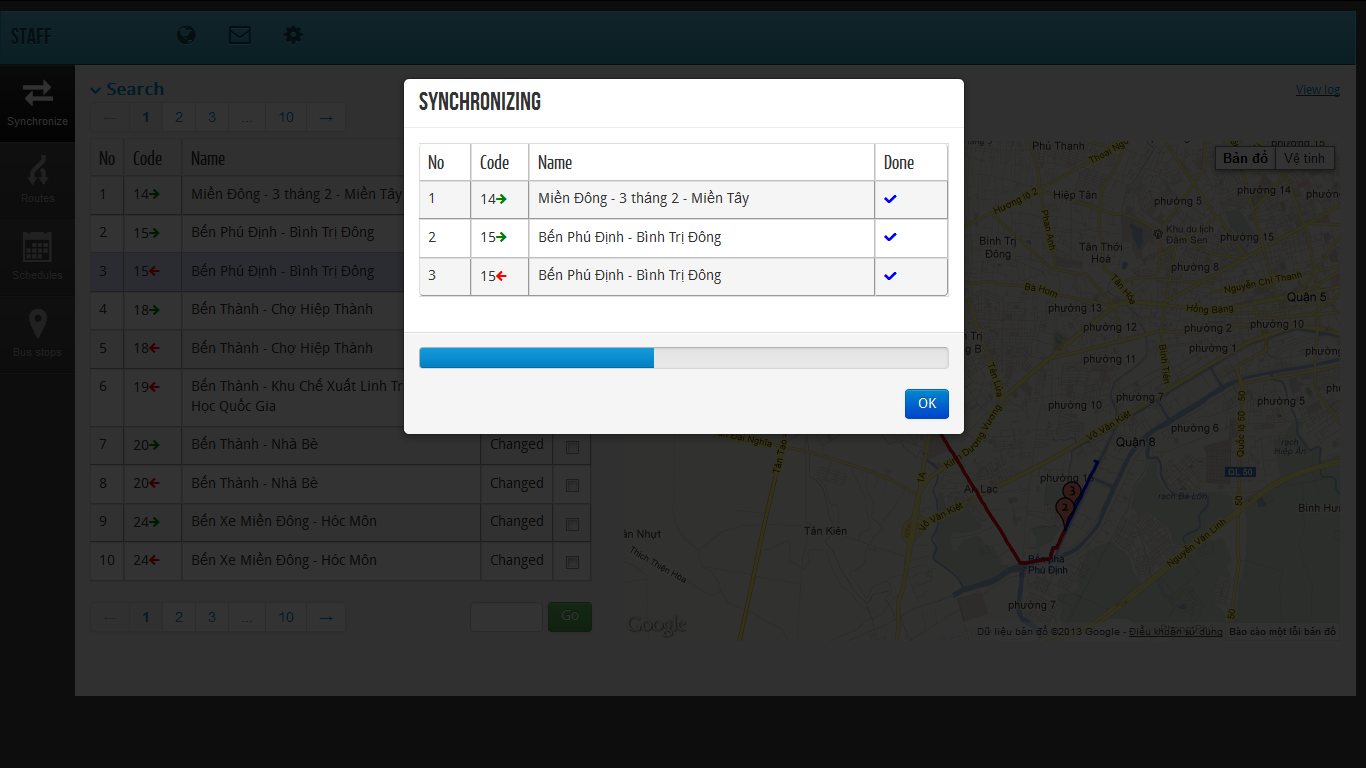


Figure 17: Synchronize bus routes page and modal for synchronizing process

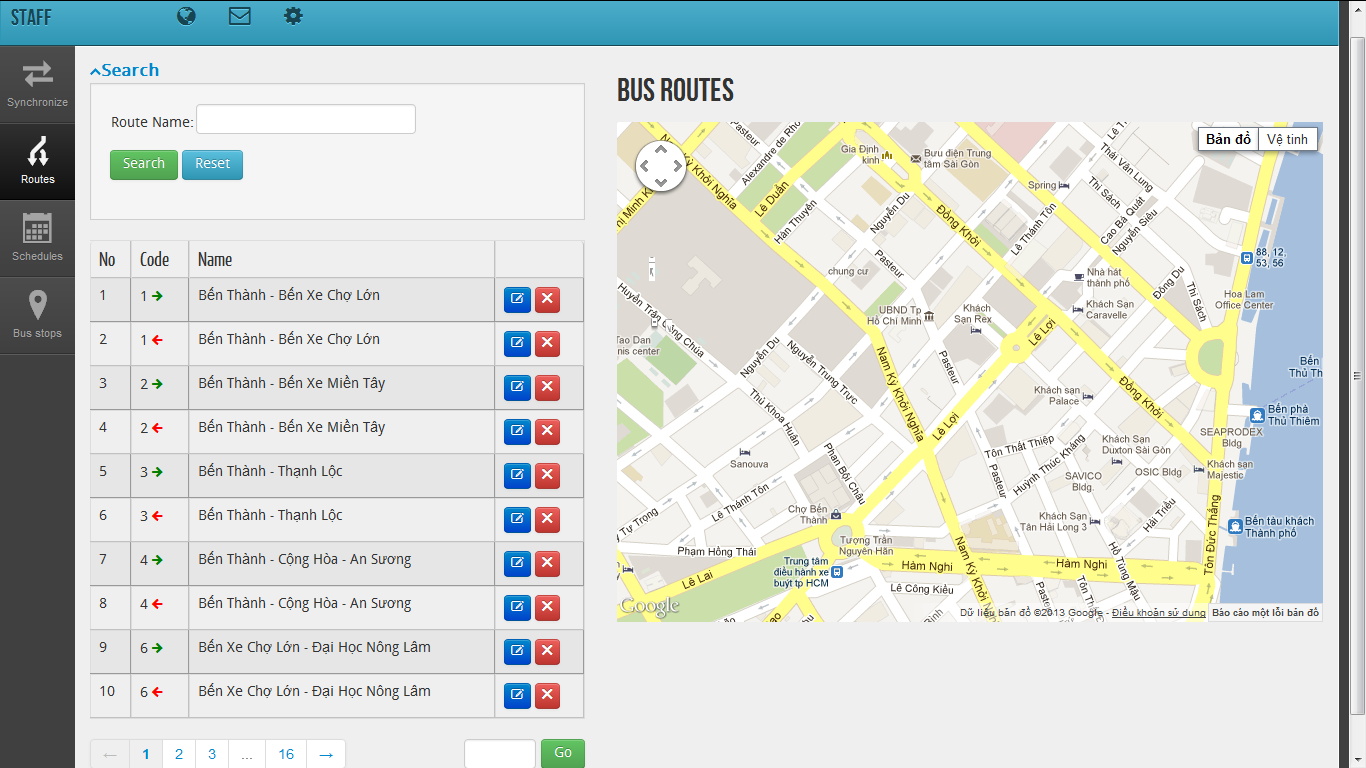
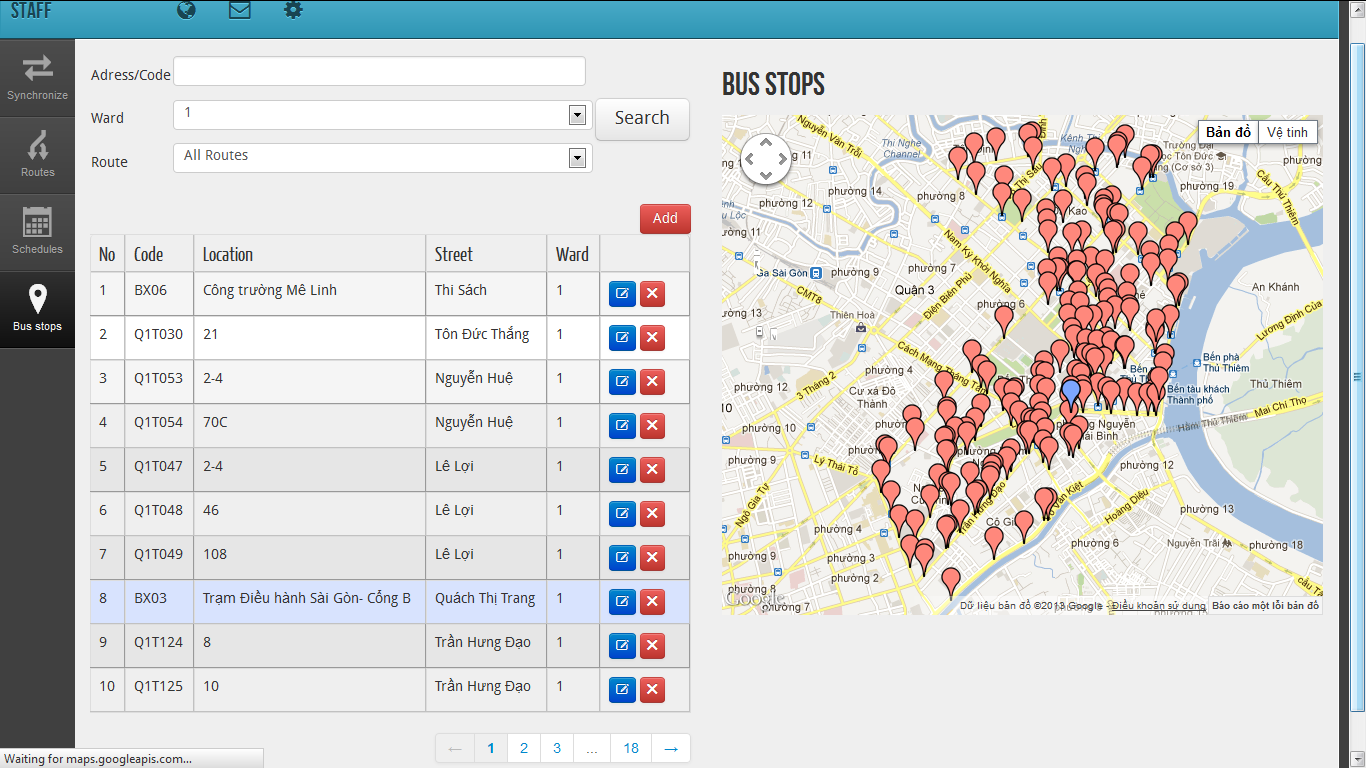


Figure 18: Routes management page



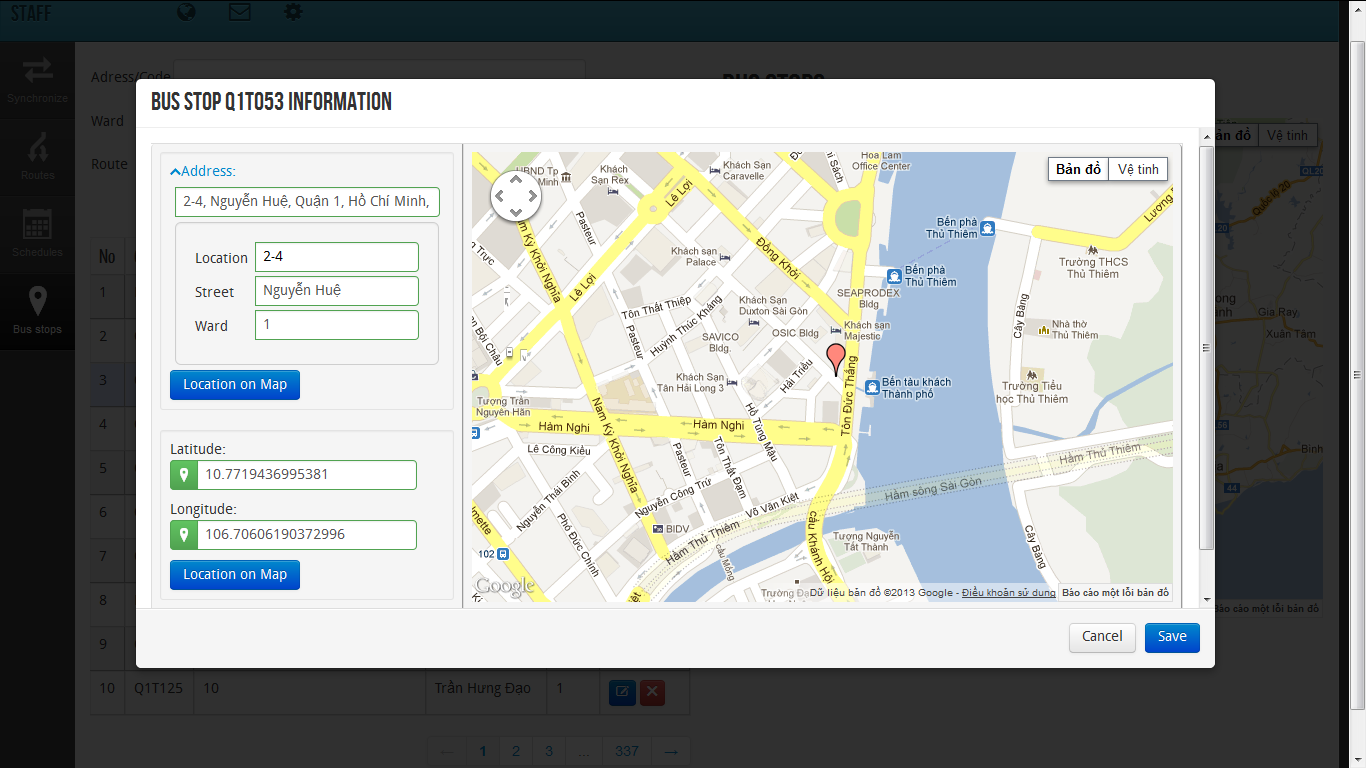


Figure 19: Bus stop management page and modal for editing bus stop

#### Objects and Actions

## Database Design or Data Structures

*<Provide the detailed database design for the system here. If your team uses a file or in-memory storage facility instead of database, remove this section; use the ‘Data Structures’ section. >*

1. http://www.nctr.usf.edu/jpt/pdf/JPT12-1Fujii.pdf [↑](#endnote-ref-1)
2. http://www.tinmoi.vn/nam-2012-tphcm-co-370-trieu-luot-nguoi-di-xe-buyt-011153226.html [↑](#endnote-ref-2)
3. http://www.buyttphcm.com.vn/TTLT.aspx [↑](#endnote-ref-3)
4. https://www.assembla.com/spaces/free-private-subversion-repository-only/prepare\_copy?type=private [↑](#endnote-ref-4)
5. https://www.assembla.com/spaces/free-agile-task-management/prepare\_copy?type=private [↑](#endnote-ref-5)
6. http://www.agilekiwi.com/other/agile/definition-of-agile-development/ [↑](#endnote-ref-6)
7. http://msdn.microsoft.com/en-us/library/vstudio/ff926074(v=vs.100).aspx [↑](#endnote-ref-7)
8. http://www.esri.com/news/arcuser/0609/aspnetmvc.html [↑](#endnote-ref-8)