



FPT UNIVERSITY

Capstone Project Document

Search Engine Platform for Traveling Information based on Apache Lucene

Group 01	
Group members	Giang Song Trúc – SE61268 Hồ Huy Cường – SE61155 Lê Vĩnh Hoài Trung – SE61074 Nguyễn Duy Khương – 60493
Supervisor	Nguyễn Huy Hùng
Ext. Supervisor	N/A
Capstone Project code	SEPTIAL

- Ho Chi Minh city, January 4th 2016 -

Table of Contents

Definitions, Acronyms, and Abbreviations.....	6
A. Report No. 1 Introduction	7
1. Project Information.....	7
2. Introduction	7
3. Current Situation.....	7
4. Problem Definition.....	7
5. Proposed Solution	8
5.1 Feature functions	8
5.2 Advantages and disadvantages.....	8
6. Functional Requirements	9
7. Role and Responsibility.....	9
B. Report No.2 Software Project Management Plan.....	10
1. Problem Definition.....	10
1.1. Name of this Capstone Project	10
1.2. Problem Abstract	10
1.3. Project Overview	10
2. Project organization	13
2.1. Software Process Model	13
2.2. Roles and responsibilities.....	14
2.3. Tools and Techniques.....	15
3. Project Management Plan.....	15
3.1. Software development life cycle	15
3.2. Phase Detail	17
3.1. All Meeting Minutes.....	20
4. Coding Convention	20
C. Report No. 3 Software Requirement Specification.....	21
1. User Requirement Specification	21
1.1. User Requirement.....	21
1.2. Staff Requirement	21
1.3. Admin Requirement.....	21
1.4. Scheduler Requirement	21
2. System Requirement Specification	21
2.1. External Interface Requirement.....	21
2.2. System Overview Use Case.....	22

2.1. List of Use Case.....	23
3. Software System Attribute.....	78
3.1. Usability	78
3.2. Reliability	78
3.3. Availability.....	78
3.4. Security	78
3.5. Maintainability.....	78
3.6. Portability	78
3.7. Performance.....	78
4. Conceptual Diagram	78
D. Report No. 4 Software Design Description.....	80
1. Design Overview	80
2. System Architecture Design	81
1.1. Web Application architecture description	81
1.2. Mobile Application architecture description.....	81
3. Component Diagram	82
4. Detail Description.....	82
4.1. Class Diagram	82
4.2. Class Diagram Explanation.....	83
4.3. Interaction Diagram	87
5. User Interface Design.....	113
5.1. Web app	113
5.2. Mobile app.....	128
6. Database Design.....	130
6.1. Entity Relationship Diagram.....	130
6.2. Entity dictionary.....	130
7. Algorithms.....	132
7.1. Parse data from Html.....	132
7.2. Parse data from Ajax.....	133
7.3. Remove low-valuable search keywords.....	134
7.4. Search data by keyword	135
7.5. Suggest relevant keyword	136
7.6. Suggest relevant data.....	137
7.7. Suggest near-by hotels	138
7.8. Categorize data by location	139

7.9.	Group hotel services	140
7.10.	Updating data process	141
E.	System Implementation & Test.....	143
1.	Introduction.....	143
1.1.	Overview.....	143
1.2.	Test Approach	143
2.	Database relationship diagram	143
2.1.	Physical Diagram.....	143
2.2.	Data Dictionary	145
3.	Test Plan.....	146
3.1.	Feature to be tested.....	146
3.2.	Feature not to be tested.....	146
4.	System Testing Test Case	146
4.1.	Test case	146
F.	Sofware User's Manual	152
1.	Installation Guide	152
1.1.	Setting up environment at server side.....	152
1.2.	Web Application Deployment Process.....	152
2.	User Guide	155
2.1.	Web application.....	155
2.2.	Mobile application	170
G.	Appendix.....	173

Definitions, Acronyms, and Abbreviations

Name	Definition
SEPTIAL	Search Engine Platform for Traveling Information based on Apache Lucene
Data source	Websites contains traveling information used to parse data
Data item	General name for hotels/tours/flights parsed from data source
Keyword	Search keyword of users
Parsing configuration	A set of settings for each data source that specify CSS selectors for data fields, which are used to check html structure of a web page and parse all the available data
SimilarityScore	The score that measure the relevance between 2 documents
CTR	Click-through rate
Users	The un-authenticated users

A. Report No. 1 Introduction

1. Project Information

- Project name:

Search Engine Platform for Traveling Information based on Apache Lucene

- Abbreviation: **SEPTIAL**
- Product Type: **Web app & Mobile app**
- Start Date: **January 4th 2016**
- End Date: **April 2016**

2. Introduction

In Vietnam, people usually have difficulty searching for traveling information as the current websites usually focus on only one aspect such as hotels, flights, tours, deals. It takes time and effort for customers to find and summarize all the search results from those data sources.

The proposed solution is to build a system that helps people search for all the needed information without having to browse several websites. The system collects data from multiple sources, analyzes users' behaviors and suggests the most appropriate traveling information available.

3. Current Situation

Nowadays, traveling has become an essential part of our lives. Normally, people have two options for their traveling plan: either find a tour or plan everything themselves.

To find the most suitable tour, users have to look through several websites of travel agencies, consider various offers before picking one.

And if they would like to plan their own trips, the workload is often heavier as the process needs to be repeated for every element of the trip, such as flights, hotels, places to go, restaurants to eat.

4. Problem Definition

Because of the current situation, people confront with many problems searching for the necessary information. First of all, most people don't know where to start, what kind of information to look for or which website to visit. Secondly, even when they already have their own intentions, they have to browse through too many websites with a huge amount of data. The main problems with this approach are ineffectiveness, time consuming and low efficiency.

5. Proposed Solution

The proposed solution is a system for searching traveling information collected from various sources, named “EzGo”. This system is not only a searching tool but also an assistant suggesting appropriate information based on other similar users’ behaviors.

EzGo includes a web app and a mobile app, with the following features:

5.1 Feature functions

- Admin/Staff can manage all the data collections and track activities of all the background processes.
- Background processes:
 - Periodically parse data from external data sources then add new data to main storage or update the existed data if there are any changes.
 - Categorize raw data and update to Lucene database.
 - Remove redundant and out-of-date data from main storage and Lucene database.
- Users can search for hotels/tours/flights by keywords, locations, date or price. After search results are displayed, users can sort and filter by criterias.
- By default, search results will be sorted by users’ trends.
- The system will suggest possible keywords when users start typing. The suggestions are based on the similarity of keyword content and numbers of users searching for it.
- Whenever users view detail of a search result, the system will display suggestion of relevant hotels, tours, flights based on the content of those data items, users’ location and other users’ activities.
- User activities will be collected for optimizing search results and suggestions along with search results.
- Mobile app are developed with all the searching features as the web app.

5.2 Advantages and disadvantages

Advantages:

- Data is periodically collected and updated from different sources so that users can compare the results and find the most suitable for their needs.
- Users can search for all the information about hotels, flights or tours without browsing through several websites.
- Search result can be filtered or sorted by criterias.
- Users can see suggested information of what other users are interested in or related information to the search result.
- The search result will be optimized by the time based on users activities.

Disadvantages:

- Data will not be real-time updated.
- The collecting data module must be configured for each websites and can be affected by any changes of the data sources.

6. Functional Requirements

System:

- Parse and collect data from data sources on a schedule
- Periodically update data from main storage to Lucene
- Periodically categorize parsed data
- Optimize search results and recommendations for users
- Provide API for mobile app

Admin:

- Manage data sources
- Manage schedulers
- Configure data parsing
- Manage parsed data

User:

- Search for hotels, flights, tours by location, date or price
- Filter and sort search results
- See suggestions of trending or related information
- Use all the functions in web app and mobile app

7. Role and Responsibility

No	Full Name	Role	Position	Contact
1	Nguyễn Huy Hùng	Product owner	Supervisor	hungnh@fpt.edu.vn
2	Giang Song Trúc	Developer	Leader	trucgsse61268@fpt.edu.vn
3	Hồ Huy Cường	Developer	Member	cuonghhse61155@fpt.edu.vn
4	Lê Vĩnh Hoài Trung	Developer	Member	trunglvhse61074@fpt.edu.vn
5	Nguyễn Duy Khuong	Developer	Member	khuongnd60493@fpt.edu.vn

Table 1 : Roles and Responsibilities

B. Report No.2 Software Project Management Plan

1. Problem Definition

1.1. Name of this Capstone Project

- **Official name:**
Search Engine Platform for Travelling Information Based on Apache Lucene
- **Abbreviation:** SEPTIAL

1.2. Problem Abstract

Nowadays in Vietnam, websites and applications for traveling on the market often focus on only a certain task, e.g hotels, flights and tours.

Those websites that focus on several tasks are often designed for international users and thus, are not particularly suitable for Vietnamese people.

The solution is to provide a system for Vietnamese people which allows them to search for everything needed for a trip in a simple and effective way. People don't have to go through several websites to plan their own trip.

1.3. Project Overview

1.3.1. Current Situation

Below are the problems encountered in this project:

- **Results for real-time data may not be correct:** All results provided are collected periodically, so the information may not be updated immediately whenever data changes occur.
- **Some information require premium package to access:** Valuable information, such as flight fare or number of seats available, require to sign contracts with sellers in order to access efficiently.
- **Data source formatting:** Websites where the system get the data from may change structures or flows, which makes the data parsing modules not working correctly.
- **IP block:** Some websites limit the number of time to access their page, thus the system may encounter IP blocking from these sites.

1.3.2. The Proposed System

- By periodically collecting and updating data from several sources, the system provides synthetic information for users.
- User activities are collected and analyzed for optimizing search results time by time. By studying user behaviors, the system can suggest some related information that users may be interested in.
- Data synchronization can be satisfied by adjusting data parsing schedule. Data can be considered up-to-date with appropriate settings for particular datasources.

- In order to parse data from any data sources, admin is provided utilities to configure the data parsing module depended on structure of data sources and demand of the admin. Flexible configurations help admin easily adapt to changes of data sources.

1.3.2.1. System

System is an essential part with main features:

- Background process:
 - Periodically parse data from data sources and update to the main storage.
 - Periodically categorize and update data from the main storage to Lucene database.
 - Periodically update new keywords and remove low-value keywords.
 - Periodically remove redundant and out-of-date data from the main storage and Lucene database.
- Find and suggest appropriate information for users based on the data contents and users' activities.
- Collect and analyze user behaviors.
- Provide API for mobile app.

1.3.2.2. Web app

Web app consists of two main parts:

- For users:
 - Search and display information about hotels, flights and tours.
 - Filter search results: by data type (hotel/tour/flight), by price range, by hotel services, by flight agency, by locations, by date/time.
 - Sort search results: by price, by rating, by date/time, by trending.
 - View trending and related results: relevant keywords, relevant hotels, relevant tours, relevant flights, nearby hotels.
- For admin & staff:
 - Manage data sources
 - Configure settings and schedule to collect data from data sources
 - Manage schedulers
 - Manage parsed data

1.3.2.3. Mobile app

Mobile application is an optional part used by users. It has the same features as the website counterpart.

1.3.3. Boundaries of the System

- Collect data from data sources
- Allow users to search for information about hotels, flights and tours by locations, date or price as well as filter and sort search results
- Suggest related results to users
- Allow users to redirect to appropriate websites from result page

1.3.4. Future Plans

With further research and development, the system can apply the following features:

- The ranking and optimizing algorithms will be improved to capture the search trends at different periods of time.
- Beside specifying data sources, the system will have a data crawling module to find data sources and parse needed data automatically.
- The supported information can include interesting places, restaurant, etc.
- Beside flight data, the system will be upgraded to collect data of the other means of transport.
- Upgrade parsing modules to be able to collect more detail information such as hotel room prices, deals, offers.

1.3.5. Development Environment

1.3.5.1. Hardware requirements

- For server

Windows	Minimum Requirements	Recommended
Internet Connection	Cable, Wi-Fi (4 Mbps)	Cable, Wi-Fi (8 Mbps)
Operating System	Window Server 2008	Window Server 2012
Computer Processor	Intel® Xeon ® 1.4GHz (12M Cache, 2.50 GHz)	Intel® Xeon ® Quad Core

Table 2: Hardware Requirement for Server

- For Mobile

Windows	Minimum	Recommended
Internet Connection	Wi-Fi or 3G (4 Mbps)	Wi-Fi or 3G (8 Mbps)
Operating System	Android 4.2.2	Android 4.4.2

Computer Processor	Cortex-A7 Dual-Core 1.3GHz	Cortex-A7 Dual-Core 1.3GHz
Computer Memory	1GB RAM	2GB or more

Table 3: Hardware Requirement for Mobile

1.3.5.2. Software requirements

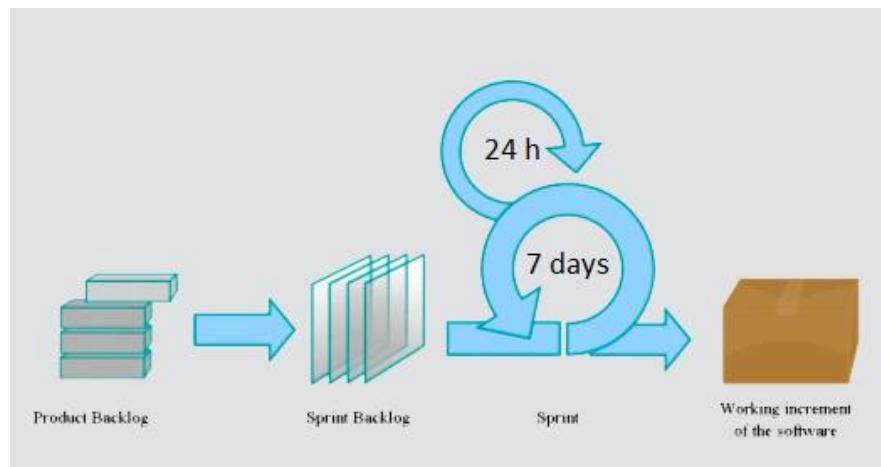
Software	Name / Version	Description
Operating system	Window Server 2012	Operating system and platform for development
Environment	.NET Framework 4.5	Specification for developing web application
IDE	Visual Studio 2013, Android Studio v1.5.1	Used for implement website and Android Mobile App.
Design Model tool	StartUML v2.5.1	Used for creating modal and diagrams.
DBMS	Microsoft SQL Server 2014	Used to create & manage the database for system
Document storage	Google Drive	Used for storing document
Store and manage source code	Team Foundation Server	Used to store all source code

Table 4: Software requirements

2. Project organization

2.1. Software Process Model

The project is developed under Scrum model.



Scrum is an iterative and incremental agile software development methodology for managing product development. It defines "a flexible, holistic product development strategy where a development team works as a unit to reach a common goal", challenges assumptions of the "traditional, sequential approach" to product development, and enables teams to self-organize by encouraging physical co-location or close online collaboration of all team members.

Scrum adopts an empirical approach—accepting that the problem cannot be fully understood or defined, focusing instead on maximizing the team's ability to deliver quickly, to respond to emerging requirements and to adapt to evolving technologies and changes in market conditions.

2.2. Roles and responsibilities

No	Full name	Role in Group	Responsibilities
1	Nguyễn Huy Hùng	Supervisor, Product owner	<ul style="list-style-type: none"> • Specify user requirements • Control the development process • Give out technique and business analysis support
2	Giang Song Trúc	Team leader, B.A, Developer, Tester	<ul style="list-style-type: none"> • Managing process • Designing database • Clarifying requirements • Prepare documents • GUI design • Create test plan • Coding • Testing
3	Hồ Huy Cường	Team member, B.A, Developer, Tester	<ul style="list-style-type: none"> • Designing database • Clarifying requirements • Prepare documents • GUI design • Create test plan • Coding • Test
4	Lê Vĩnh Hoài Trung	Team member, B.A, Developer, Tester	<ul style="list-style-type: none"> • Designing database • Clarifying requirements • Prepare documents • Create test plan • Coding • Test

5	Nguyễn Duy Khương	Team member, B.A, Developer, Tester	<ul style="list-style-type: none"> • Designing database • Clarifying requirements • Prepare documents • Coding • Test
---	-------------------	---	--

Table 5: Roles and responsibilities

2.3. Tools and Techniques

- Front – end technologies:
 - HTML5
 - CSS3
 - JavaScript
 - JQuery
- Back – end:
 - Website: ASP.NET MVC 5 + Entity Framework
 - Web Service: ASP .Net Web API 2.
 - Mobile App: Android
- Web Server: Microsoft IIS.
- Database Management System: MS SQL Server 2014

3. Project Management Plan

3.1. Software development life cycle

Phase	Description	Deliverables	Resource needed	Dependencies and Constrains	Risks
Infrastructure	<ul style="list-style-type: none"> - Identify and clarify overall requirements. - Determine the system architecture. - Build infrastructure for the project. 	<ul style="list-style-type: none"> - Database design. - Project main structure with the components: Data, Parsing, Lucene, WebApp 	20 man-days		<ul style="list-style-type: none"> - Unclear project scope. - Lack of member share of understand.
System & Web app	<ul style="list-style-type: none"> - Identify and clarify the needed requirements. - Build all the background modules. - Design the web UI. - Build the web app. 	<ul style="list-style-type: none"> - Schedulers: Parsing, Data updating, Redundant data removing. - Search algorithms. - Complete web app for admin & end user. 	300 man-days	<ul style="list-style-type: none"> - Depends on "Infrastructure" 	<ul style="list-style-type: none"> - Unclear project scope. - Lack of knowlegde in Apache Lucene. - Lack of experience.
Web services	<ul style="list-style-type: none"> - Indentify and clarify requirements for the mobile app. - Build the required API for mobile app. 	<ul style="list-style-type: none"> - API for end user functions 	20 man-days	<ul style="list-style-type: none"> - Depends on "Web app & System" 	<ul style="list-style-type: none"> - Lack of experience.
Mobile app	<ul style="list-style-type: none"> - Design the mobile UI - Build mobile app for end users. 	<ul style="list-style-type: none"> - Complete android app. 	50 man-days	<ul style="list-style-type: none"> - Depends on "Web services" 	<ul style="list-style-type: none"> - Lack of experience.

Table 6: Software Development Life Cycle Detail

3.2. Phase Detail

3.2.1 Phase 1: Infrastructure

Task	Description	Author
1. Identify & clarify overall requirements	<ul style="list-style-type: none"> - Research some systems currently provide similar service. - Determine core functions. 	<ul style="list-style-type: none"> • TrucGS • CuongHH • TrungLVH • KhuongND
2. Determine system architecture	<ul style="list-style-type: none"> - Determine system architecture: ASP .NET MVC. - Determine software design pattern: Repository & Service. 	<ul style="list-style-type: none"> • TrucGS
3. Build system infrastructure	<ul style="list-style-type: none"> - Create the main structure of project 	<ul style="list-style-type: none"> • TrucGS
4. Create report & SRS document	<ul style="list-style-type: none"> - Report: project overview, problem definition, proposed solutions - SRS: functional requirements 	<ul style="list-style-type: none"> • TrucGS • CuongHH • TrungLVH • KhuongND

Table 7: Phase 1: Infrastructure

3.2.2 Phase 2: System & Web app

Task	Description	Author
1. Identify & clarify requirements	<ul style="list-style-type: none"> - Determine actors and functions. - Create Use Case diagram 	<ul style="list-style-type: none"> • TrucGS • CuongHH • TrungLVH • KhuongND
2. Design web UI	<ul style="list-style-type: none"> - Design and build prototype for web UI 	<ul style="list-style-type: none"> • TrucGS • CuongHH
3. Design database	<ul style="list-style-type: none"> - Create conceptual diagram - Design class diagram - Design database 	<ul style="list-style-type: none"> • TrucGS • CuongHH

4. Build Lucene server	<ul style="list-style-type: none"> - Research: Lucene .NET - Implement all the necessary modules: indexing & CRUD operations. 	• TrucGS
5. Build Web parsing module	<ul style="list-style-type: none"> - Research: CSQuery - Build generic library for html parsing. - Implement static web parsing module. - Implement configuration module for admin/staff. 	• TrucGS • TrungLVH • KhuongND
6. Build Ajax parsing module	<ul style="list-style-type: none"> - Research: Ajax, Proxy - Research the target data sources: Abay, Avia - Find free proxy list provider - Prioritize proxies using algorithm: Round Robin - Implement Ajax parsing using proxies. 	• TrungLVH
7. Implement Front-end functions	<ul style="list-style-type: none"> - Implement all the web UI: layouts, search flow, detail pages, etc. 	• CuongHH
8. Implement Back-end functions	<ul style="list-style-type: none"> - Implement all the functions in controllers. - Build needed utility classes. - Define sorting & filtering criterias - Suggest nearby hotels: Euclidean distance 	• TrucGS • CuongHH • Trung LVH
9. Implement search algorithms	<ul style="list-style-type: none"> - Research & Implement: Lucene search mechanisms, string comparison with N-gram & Levenshtein distance algorithm 	• TrucGS

	<ul style="list-style-type: none"> - Tracking user behaviors: CTR - Suggestion algorithm: Pearson Correlation Coefficient. 	
10. Implement Admin control panel	<ul style="list-style-type: none"> - Implement all the management modules for admin and staff 	<ul style="list-style-type: none"> • TrucGS • KhuongND
11. Build schedulers	<ul style="list-style-type: none"> - Build schedulers for all the background tasks 	<ul style="list-style-type: none"> • TrucGS • CuongHH • TrungLVH

Table 8: Phase 2: System & Web app

3.2.3 Phase 3: Web service

Task	Description	Author
1. Determine needed functions	<ul style="list-style-type: none"> - Determine functions for mobile app. - Write report & SRS 	<ul style="list-style-type: none"> • TrucGS • CuongHH • TrungLVH • KhuongND
2. Create API	<ul style="list-style-type: none"> - Create API for mobile app based on functions on the web app. 	<ul style="list-style-type: none"> • CuongHH • KhuongND

Table 9: Phase 3: Web service

3.2.4 Phase 4: Mobile app

Task	Description	Author
1. Design UI	<ul style="list-style-type: none"> - Design UI for all the functions of mobile app 	<ul style="list-style-type: none"> • TrucGS • CuongHH
2. Implement	<ul style="list-style-type: none"> - Implement all the functions based on the designed UI and the provided API. 	<ul style="list-style-type: none"> • TrucGS

Table 10: Phase 4: Mobile app

3.1. All Meeting Minutes

Meeting minutes are contained in folder “Meeting minutes” in the attached CD.

4. Coding Convention

C#: Using to develop website and web service.

Summary:

- Naming Convention:
 - + For variable's name, use camel case. Eg: minValue, maxValue...
 - + For function name, class name, use Pascal case. Eg: AddIncome, AddExpense...
- Layout Convention:
 - + Indent continuation one tab stop (four spaces).
 - + Add at least one blank line between method definitions and property definitions.
 - + Use parentheses to make clauses in an expression apparent.
- Commenting Convention:
 - + Begin comment text with an uppercase letter.
 - + End comment text with a period.
 - + Insert one space between the comment delimiter (//) and the comment text.
 - + Do not create formatted blocks of asterisks around comments.

Android: Using to develop mobile application

Summary:

- Naming Convention:
 - + For variable's name, use camel case. Eg: minValue, maxValue...
 - + For function name, class name, use Pascal case. Eg: AddIncome, AddExpense...
 - + For resource file names are written in lowercase_underscore. Eg: my_name
- Commenting Convention:
 - + Begin comment text with an uppercase letter.
 - + End comment text with a period.
 - + Insert one space between the comment delimiter (//) and the comment text.
 - + Do not create formatted blocks of asterisks around comments.

C. Report No. 3 Software Requirement Specification

1. User Requirement Specification

1.1. User Requirement

- Search and view detail of search results.
- Sort & filter search result.
- Search hotels/flights/tours by particular criterias
- Login

1.2. Staff Requirement

- Manage data sources & parsing configuration
- Manage schedulers
- Manage parsed data
- Logout

1.3. Admin Requirement

Admin is a staff with more special privileges:

- Manage staff accounts
- Active/Deactive accounts

1.4. Scheduler Requirement

- Periodically parse data from data sources by following the specified configuration to update main data storage.
- Periodically update data from main storage to Lucene server.
- Periodically categorize parsed data
- Periodically update/remove search keywords

2. System Requirement Specification

2.1. External Interface Requirement

2.1.1. User Interface

- The user interface uses Vietnamese language in both web application and android application.
- The user interface displays best on 1366x768 resolution on desktop, and 1080x1920 on mobile.

2.1.2. Hardware Interface

- Desktop:
 - o RAM: 2GB
 - o CPU: 2.5Ghz
- Android Smartphone:
 - o RAM: 512MB
 - o CPU: 1Ghz

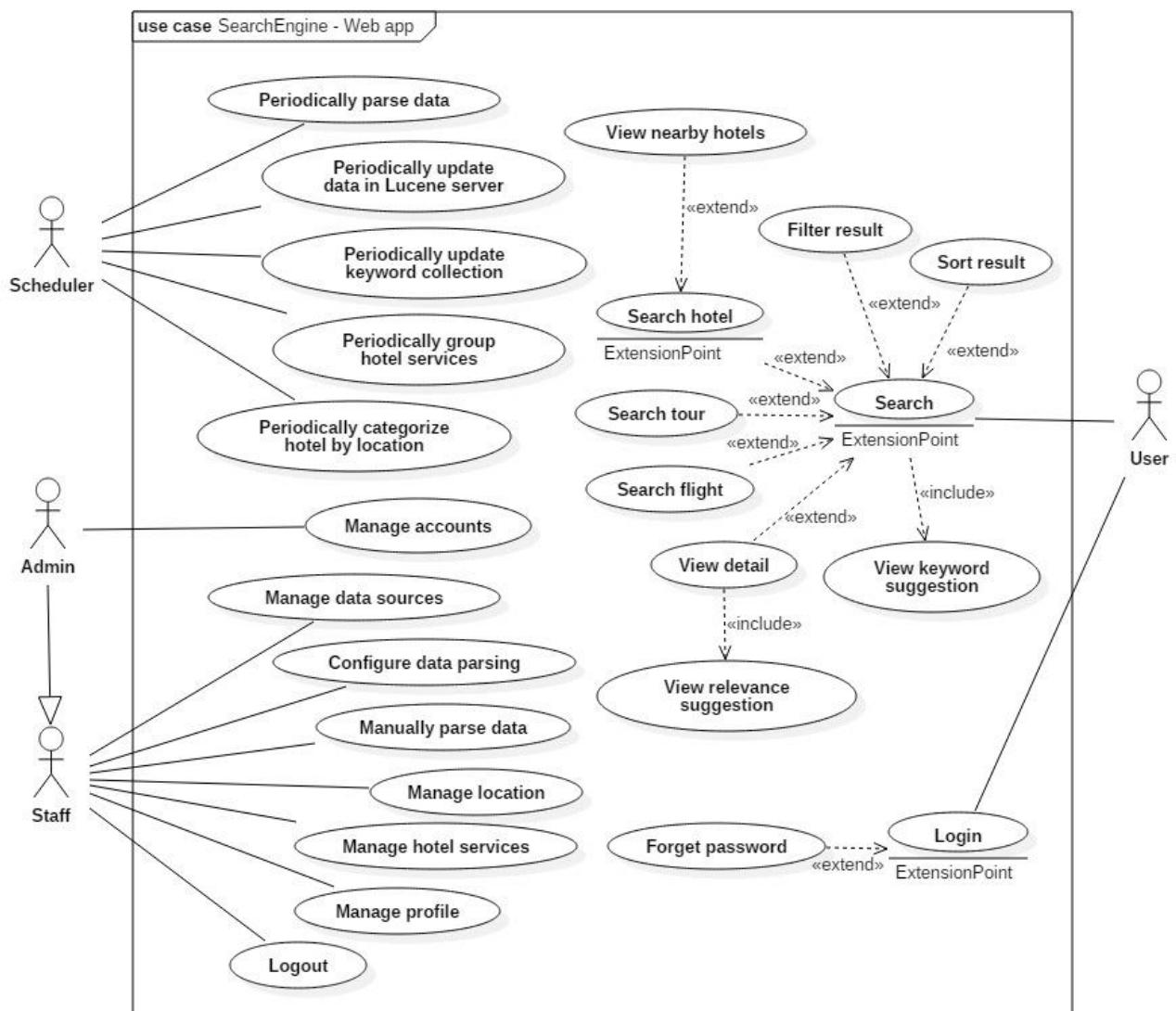
2.1.3. Software Interface

- Web application: Works with Firefox (v30 or above), Google Chrome (v25 or above), Internet Explorer (v10 or above) or with any web browser that supports HTML5 & CSS3.
 - Mobile application: Android operating system (v4.0 or above).

2.1.4. Communication Protocol

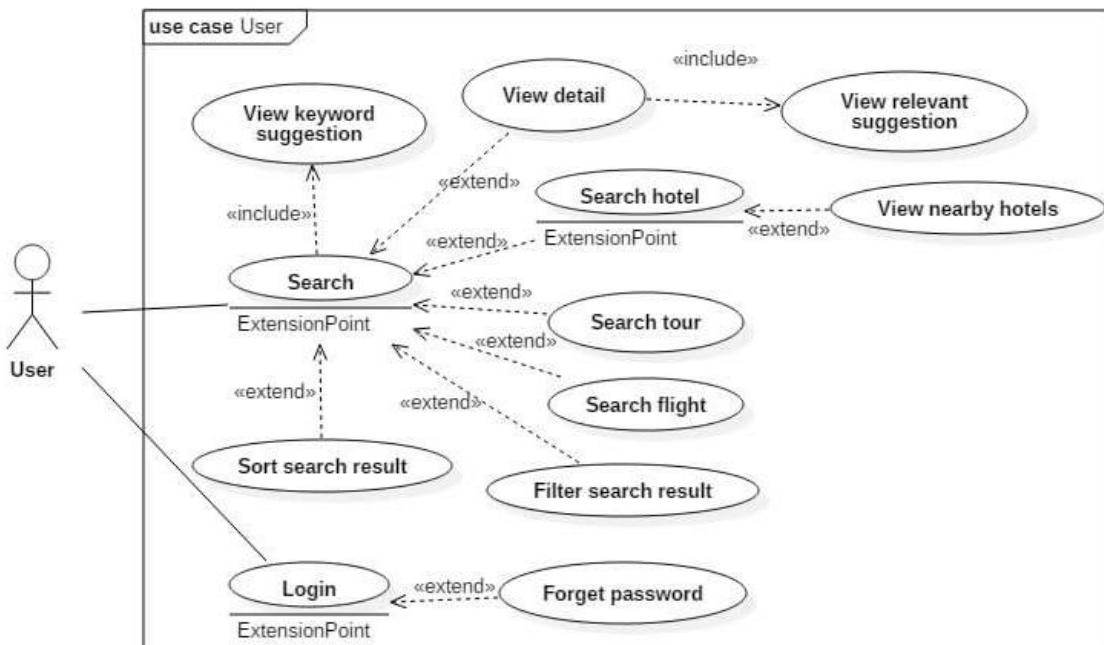
- Use HTTP 1.1 to communicate between the web browser and the web server.

2.2. System Overview Use Case

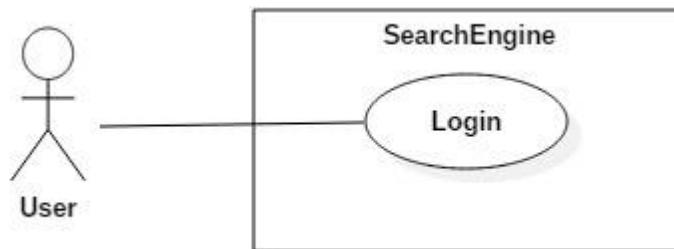


2.1. List of Use Case

2.2.1. <User> Overview Use Case



2.2.1.1. <User> Login



USE CASE - <UC number>

Use Case No.	UC 1.1	Use Case Version	1.0
Use Case Name	Login		
Author	KhuongND		
Date	3/30/2016	Priority	Normal

Actor:

- User

Summary:

- This use case allows authorized user to log in system.

Goal:

- User can log in the system.

Triggers:

- User sends the login command.

Preconditions:

- N/A

Post Conditions:

- **Success:** User login successful into the system and can use all functions follow role.
- **Fail:** Show error message

Main Success Scenario:

Step	Actor Action	System Response
1	User goes to login page.	System require information: - Username: free text input. - Password: free text input.
2	User inputs information (username and password). [Exception 2, 3]	
3	User sends command to login (Click Login button after input login form). [Alternative 1]	User will login system with their specific roles. [Exception 1,2,3]

Alternative Scenario:

[Alternative 1]

Step	Actor Action	System Response
1	User click "Forgot Password" link.	System will direct to "Forgot password" page.

Exceptions:

[Exception 1]

No	Cause	System Response
1	Wrong username or password.	Show error message: "Your username and password are invalid. Try again".

[Exception 2]

No	Cause	System Response
2	"Username" is empty or least than 6 words.	Show error message: "Please enter username more than 6 words".

[Exception 3]

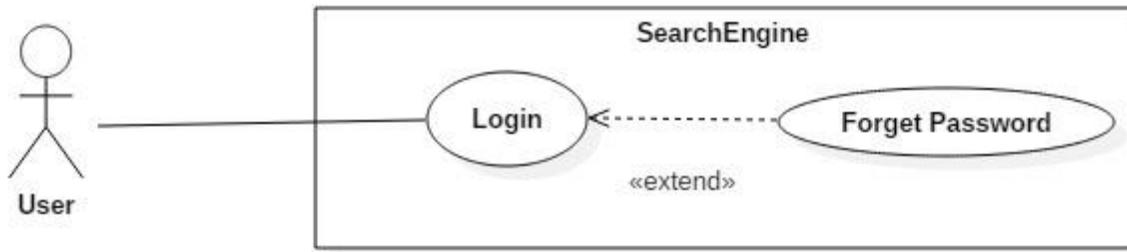
No	Cause	System Response
2	"Password" is empty or least than 6 words.	Show error message: "Please enter password more than 6 words".

Relationships: N/A

Business Rules:

- Username: textbox, at least 6 words, required.
- Password: textbox, at least 6 words, required.
- Logged in users have 2 role: staff and admin

2.2.1.2. <User> Forget password



USE CASE - <UC number>

Use Case No.	UC 1.2	Use Case Version	1.0
Use Case Name	Forget passwod		
Author	KhuongND		
Date	3/30/2016	Priority	Normal

Actor:

- User

Summary:

- This use case allows user can send request to reset their password.

Goal:

- User will receive an email from system to request reset password.

Triggers:

- User sends the forget password command.

Preconditions:

- N/A

Post Conditions:

- **Success:** Send email confirm reset password.
- **Fail:** Show error message.

Main Success Scenario:

Step	Actor Action	System Response
1	User goes to login page and click link “Forgot password?”.	<p>System require information:</p> <ul style="list-style-type: none"> - Email: textbox ,email type, required. - Email link: button send forgot password command. - Reset password: link direct to “Reset password” page. - Login: link direct to “Login” page.
2	User input information (email).	
3	User sends command to forgot password with inputted information to system(Click Email Link button after fills forgot password form).	User will be notified “Please check your email to have code for reset your password at this link”.

	[Alternative 1,2]	[Exception 1,2]
--	-------------------	-----------------

Alternative Scenario:

[Alternative 1]

Step	Actor Action	System Response
1	User click "Reset password" link.	System will direct to "Reset password" page.

[Alternative 2]

Step	Actor Action	System Response
1	User click "Login" link.	System will direct to "Login" page.

Exceptions:

[Exception 1]

No	Cause	System Response
1	Email does not valid email	System show error message: "Email is incorrect. Your input please email".

[Exception 2]

No	Cause	System Response
2	Email is not found in the database.	System show error message: "This account does not exist".

Relationships: extends UC "Login"

Business Rules:

- A token code will be created with 8 words which will be random:**[a-zA-Z0-9]**. This token code will be sent to the email user and stored in session (will be used in "Reset password" function).
- Email pattern should be: **/^ [a-zA-Z0-9]!#\$%&'*+ \ /=?^_`{|}~ -] + @ [a-zA-Z0-9] (?: [a-zA-Z0-9-] {0,61} [a-zA-Z0-9])? (?: \. [a-zA-Z0-9] (?: [a-zA-Z0-9-] {0,61} [a-zA-Z0-9])?) * \$ /** (following RFC 5322 standard).
- Form Email will be sent:

Thank nguyenromio@gmail.com have used our services.

If you have forgotten your password , you can put it on : [http://\[REDACTED\]/Account/ResetPassword](http://[REDACTED]/Account/ResetPassword) to reset your password

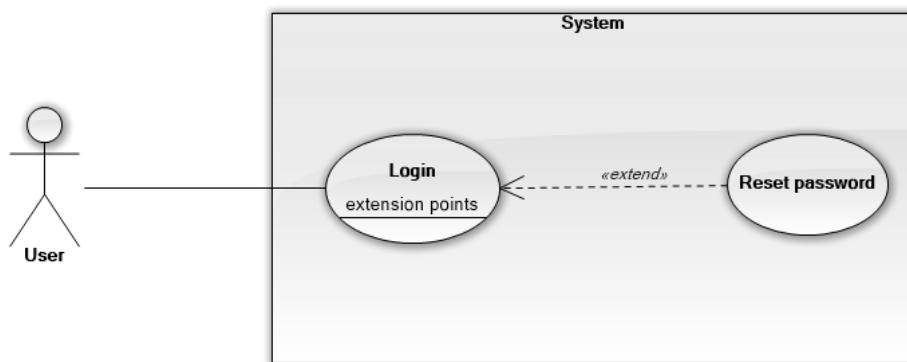
Retrieve password information:

+Your username: [REDACTED]

+Your token code: [REDACTED]

- This token code stored in session, so will expire after session time out.
- Email will be checked already exist in database before do process.

2.2.1.3. <User> Reset password



USE CASE - <UC number>

Use Case No.	UC 1.3	Use Case Version	1.0
Use Case Name	Reset passwod		
Author	KhuongND		
Date	3/30/2016	Priority	Normal

Actor:

- User

Summary:

- This use case allows user can reset their password.

Goal:

- User will reset password and use this password to login.

Triggers:

- User sends the reset password command.

Preconditions:

- N/A

Post Conditions:

- **Success:** Reset password successful.
- **Fail:** Show error message.

Main Success Scenario:

Step	Actor Action	System Response
1	<p>+The first way: User goes to url login and click link "Forgot password?". At "Forgot password" page click link "Reset password".</p> <p>+The second way: Click link reset password form email (Email sent at "Forgot password" function).</p> <p>+The third way: When finish "Forgot password" function, system will show message success and link "Reset password".</p>	<p>System navigates to "Reset password" page:</p> <ul style="list-style-type: none"> - Username: textbox, length at least 6 words, required. - Token code: textbox, length 8 words, required. - Password: textbox, length at least 6 words, required. - Confirm password: textbox, length at least 6 words, required. - Reset: button send reset password command.

2	User fills in information of their account (username,token,password,confirm password).	
3	User sends command to reset password with inputted information to system(Click Reset button after fills reset password form).	System will be show success message: "Your password has been reset. Please click here to login ". [Exception 1,2,3,4,5,6,7]

Alternative Scenario:N/A

Exceptions:

[Exception 1]

No	Cause	System Response
1	"Username" is empty or least at 6 words.	System show error message: "Incorrect username. Please fills username and more than 6 words".

[Exception 2]

No	Cause	System Response
2	"Username" does not already exist in database.	System show error message: "Cant find your account".

[Exception 3]

No	Cause	System Response
3	"Password" is empty or least than 6 words.	Show error message: "Please enter password more than 6 words".

[Exception 4]

No	Cause	System Response
4	"Confirm password" is empty or least than 6 words.	Show error message: "Please enter confirm password more than 6 words".

[Exception 5]

No	Cause	System Response
5	"Password" and "Confirm password" do not match.	Show error message: "Please enter a password and confirm password match".

[Exception 6]

No	Cause	System Response
6	"Token code" is empty or not equal	Show error message: "Please enter token"

	8 words.	with 8 words".
--	----------	----------------

[Exception 7]

No	Cause	System Response
7	"Token code" is incorrect.	Show error message: "Please check your email again. Token code is incorrect".

[Exception 8]

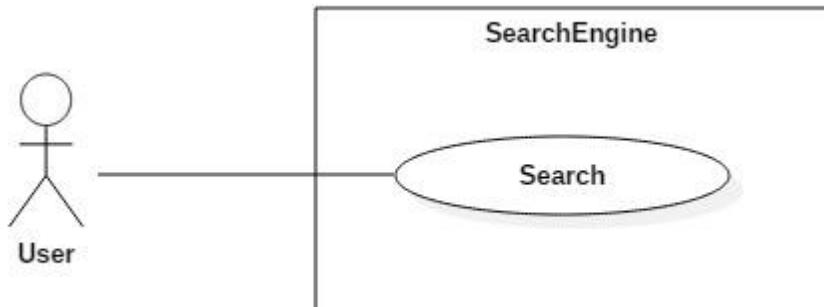
No	Cause	System Response
8	Session contains token code time out.	Show error message: "Your token code time out. Please try again".

Relationships: N/A

Business Rules:

- The token code user enter will be check match with the token in session which stored at "Forgot password" function.
- Check session contains token code. If session is null, system show error message request user use "Forgot password" function again.
- After reset password successful, all session login with this account will be logged off.
- After reset password successful, the session contains token code will be removed.
- Password must be encrypted before saving to server.

2.2.1.4. <User> General search



USE CASE - <UC number>			
Use Case No.	UC 1.4	Use Case Version	1.0
Use Case Name	General Search		
Author	TrucGS		
Date	3/30/2016	Priority	High
Actor:	<ul style="list-style-type: none"> - User 		
Summary:	<ul style="list-style-type: none"> - This use case allows user to search for general information by keyword. 		
Goal:			

- User can see the trending hotels/tours/flights related to the input keyword.

Triggers:

- User clicks search button on tab “Tìm kiếm chung”.

Preconditions:

- N/A

Post Conditions:

- **Success:** Redirect to search result page and show search results.
- **Fail:** Show error message

Main Success Scenario:

Step	Actor Action	System Response
1	In tab “Tìm kiếm chung”, enter keyword.	<ul style="list-style-type: none"> - Suggest relevant keywords in the dropdown list of the auto-complete text field.
2	Click search button.	<ul style="list-style-type: none"> - Update the input keyword. - Redirect to result page and show search result with 3 groups of data: Hotels, Tours, Flights - [Exceptions 1]

Alternative Scenario:

Step	Actor Action	System Response
1	In the search result page, click on 1 of 3 tabs: “Khách sạn”, “Tour”, “Chuyến bay”.	<ul style="list-style-type: none"> - Filter the search results by the selected data type. - Switch to the selected tab and display search results.

Exceptions:

[Exception 1]

No	Cause	System Response
1	Keyword is empty.	Show error message: “Vui lòng nhập từ khóa”.

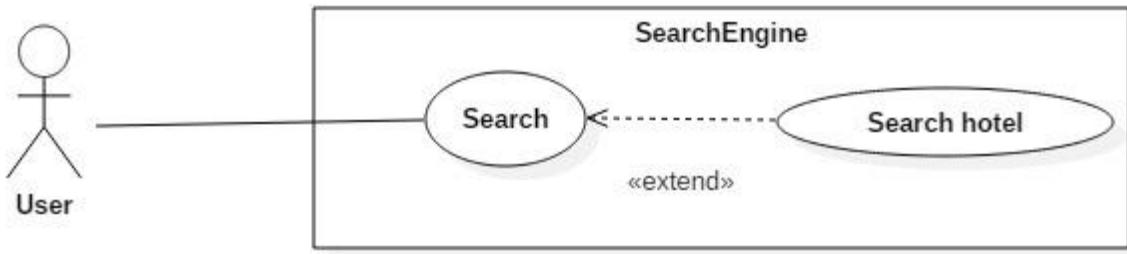
Relationships: N/A

Business Rules:

- Keyword cannot be empty.
- Update keyword:
 - If the input keyword does not exist, the keyword is added to database with SearchCount = 1;
 - Else, increase the SearchCount of the existed keyword by 1 and update to database.
- Suggest: The relevant keywords are retrieved using Pearson Correlation algorithm (mentioned in section D.7.5)

- Search: The search results are scored by $\text{RelevanceScore} = \text{SimilarityScore} * \text{Click-through rate}$ (mentioned in section D.7.4)
- Display order: The 3 groups of data (Hotels/Tours/Flights) are sorted by the average RelevanceScore of each group. In group "Hotel" and "Tour", data is sorted by RelevanceScore. In group "Flight", data is sorted by price ascending.
- Filter by data type: The filtered search results are also sorted by the RelevanceScore.

2.2.1.5. <User> Search hotel



USE CASE - <UC number>

Use Case No.	UC 1.5	Use Case Version	1.0
Use Case Name	Search hotel		
Author	TrucGS		
Date	3/30/2016	Priority	High

Actor:

- User

Summary:

- This use case allows user to search for hotels by keyword and location.

Goal:

- User can limit the search result in the scope of a city/province when searching for hotels.

Triggers:

- User clicks search button on tab "Khách sạn".

Preconditions:

- N/A

Post Conditions:

- **Success**: Redirect to search result page and show search results.
- **Fail**: Show error message.

Main Success Scenario:

Step	Actor Action	System Response
1	In tab "Khách sạn", select location and enter keyword.	- Suggest relevant keywords in the dropdown list of the auto-complete text field.
2	Click search button.	- Update the input keyword. - Redirect to result page and show search results.

		- [Exceptions 1]
--	--	------------------

Alternative Scenario: N/A

Exceptions:

[Exception 1]

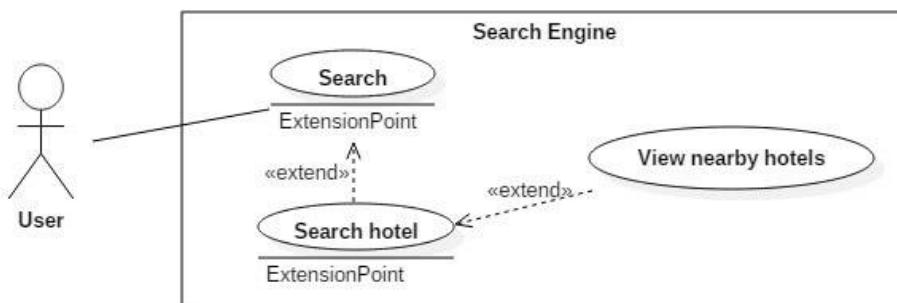
No	Cause	System Response
1	Keyword is empty.	Show error message: "Vui lòng nhập từ khóa".

Relationships: extends UC "General Search"

Business Rules:

- Keyword cannot be empty.
- Update keyword:
 - If the input keyword does not exist, the keyword is added to database with SearchCount = 1 and match to the selected location.
 - If the input keyword exists in database without relationship to the selected location, match the keyword to the location and update SearchCount.
 - Else, increase the SearchCount of the existed keyword by 1 and update to database.
- Suggest: The relevant keywords are retrieved using Pearson Correlation algorithm (mentioned in section D.7.5) and filtered by location.
- Search: The search results are scored by RelevanceScore = SimilarityScore*Click-through rate (mentioned in section D.7.4)
- Display order: By default, hotels are sorted by the RelevanceScore.

2.2.1.6. <User> View nearby hotels



USE CASE - <UC number>			
Use Case No.	UC 1.6	Use Case Version	1.0
Use Case Name	View nearby hotels		
Author	CuongHH		
Date	3/30/2016	Priority	High
Actor:			
- User			
Summary:			
- This use case allows user to view nearby hotels of a particular hotel search			

result.

Goal:

- User can find the other hotels in the same area as the selected hotel to choose the most appropriate hotel for their demand.

Triggers:

- User clicks on button “Khách sạn lân cận” in the hotel search result.

Preconditions:

- User is on the hotel search result page.

Post Conditions:

- **Success:** Show the nearby hotels.
- **Fail:** Show error message.

Main Success Scenario:

Step	Actor Action	System Response
1	In tab “Khách sạn”, click on button “Khách sạn lân cận” in the search result.	- Find 5 nearest hotels to the selected hotel and display the results.

Alternative Scenario: N/A

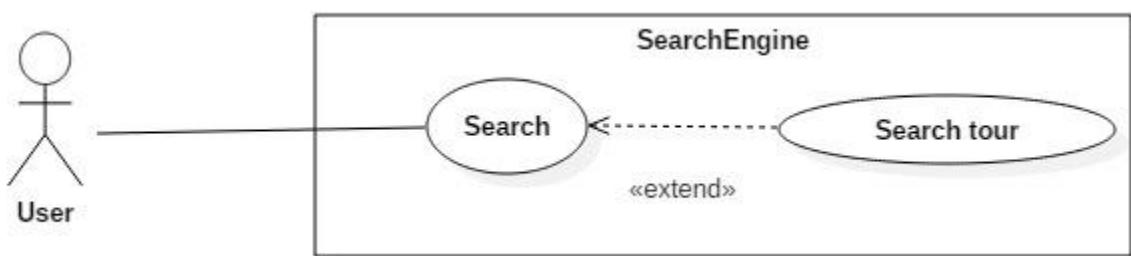
Exceptions: N/A

Relationships: extends UC “Search hotel”

Business Rules:

- The nearby hotels are found based on longitude and latitude (retrieved using hotel address and Google Map API when parsing data).
- The distance between hotels are calculated using Euclidean Distance (mention in section D.7.7).

2.2.1.7. <User> Search tour



USE CASE - <UC number>

Use Case No.	UC 1.7	Use Case Version	1.0
Use Case Name	Search tour		
Author	CuongHH		
Date	3/30/2016	Priority	High
Actor:	<ul style="list-style-type: none"> - User 		
Summary:			

- This use case allows user to search for tours by keyword and location.

Goal:

- User can limit the search result in the scope of a city/province when searching for tours.

Triggers:

- User clicks search button on tab “Tour”.

Preconditions:

- N/A

Post Conditions:

- **Success:** Redirect to search result page and show search results.
- **Fail:** Show error message

Main Success Scenario:

Step	Actor Action	System Response
1	In tab “Tour”, select location and enter keyword.	- Suggest relevant keywords in the dropdown list of the auto-complete text field.
2	Click search button.	- Update the input keyword. - Redirect to result page and show search results. - [Exceptions 1]

Alternative Scenario: N/A

Exceptions:

[Exception 1]

No	Cause	System Response
1	Keyword is empty.	Show error message: “Vui lòng nhập từ khóa”.

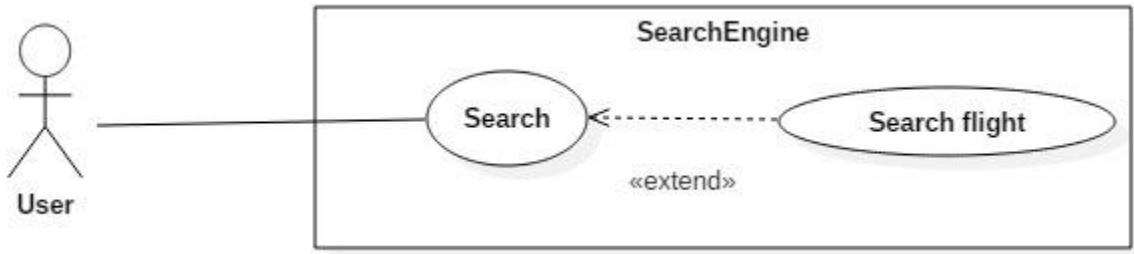
Relationships: extends UC “General Search”

Business Rules:

- Keyword cannot be empty.
- Update keyword:
 - If the input keyword does not exist, the keyword is added to database with SearchCount = 1 and match to the selected location.
 - If the input keyword exists in database without relationship to the selected location, match the keyword to the location and update SearchCount.
 - Else, increase the SearchCount of the existed keyword by 1 and update to database.
- Suggest: The relevant keywords are retrieved using Pearson Correlation algorithm (mentioned in section D.7.5) and filtered by location.
- Search: The search results are scored by RelevanceScore = SimilarityScore*Click-through rate (mentioned in section D.7.4)

- Display order: By default, tours are sorted by the RelevanceScore.

2.2.1.8.<User> Search flight



USE CASE - <UC number>

Use Case No.	UC 1.8	Use Case Version	1.0
Use Case Name	Search flight		
Author	TrungLVH		
Date	3/30/2016	Priority	High

Actor:

- User

Summary:

- This use case allows user to search for flights by date and location.

Goal:

- User can search for flights more exactly by specifying the departure/arrival date/time and locations.

Triggers:

- User clicks search button on tab “Chuyến bay”.

Preconditions:

- N/A

Post Conditions:

- **Success:** Redirect to search result page and show search results.
- **Fail:** Show error message.

Main Success Scenario:

Step	Actor Action	System Response
1	In tab “Chuyến bay”, select departure/arrival date and locations, then click search button.	<ul style="list-style-type: none"> - Redirect to result page and show search flight result. - [Exceptions 1,2,3,4,5,6]

Alternative Scenario: N/A

Exceptions:

[Exception 1]

No	Cause	System Response
1	Date start is empty.	Show error message: “Vui lòng chọn ngày đi”.

--	--	--

[Exception 2]

No	Cause	System Response
2	Date end is empty.	Show error message: "Vui lòng chọn ngày về".

[Exception 3]

No	Cause	System Response
3	Start location is empty.	Show error message: "Vui lòng chọn địa điểm khởi hành".

[Exception 4]

No	Cause	System Response
4	End location is empty.	Show error message: "Vui lòng chọn địa điểm đến".

[Exception 5]

No	Cause	System Response
5	Date start is large than date end	Show error message: "Ngày đi phải nhỏ hơn ngày về".

[Exception 6]

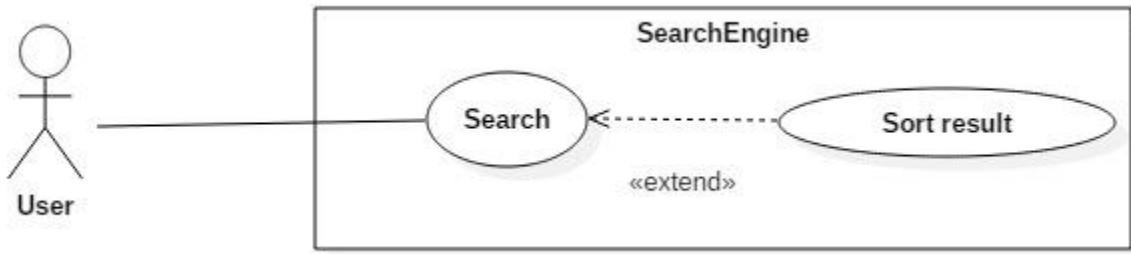
No	Cause	System Response
6	Start location and end location are duplicated.	Show error message: "Điểm đi và điểm đến không được trùng nhau".

Relationships: extends UC "General Search"

Business Rules:

- Departure date and Arrival date are in the format "dd/MM/yyyy"; Arrival date must not be less than Departure date.
- The location lists don't contain cities/provinces without airports.
- Departure location and Arrival location must not be the same.
- By default, the flight results are grouped by flight agency and sorted by date and price ascending.

2.2.1.9. <User> Sort search result



USE CASE - <UC number>

Use Case No.	UC 1.9	Use Case Version	1.0
Use Case Name	Sort result		
Author	CuongHH		
Date	3/30/2016	Priority	Normal

Actor:

- User

Summary:

- This use case allows user to sort search results by criterias.

Goal:

- The search result will be displayed in the appropriate order of user's demand.

Triggers:

- User sends sorting result request.

Preconditions:

- User is on the search result page of hotels, tours or flights.

Post Conditions:

- **Success:** Show sorted results.
- **Fail:** Show error message.

Main Success Scenario:

Step	Actor Action	System Response
1	In search result page, select sorting criterias.	- System display search result in the appropriate order.

Alternative Scenario:

N/A

Exceptions:

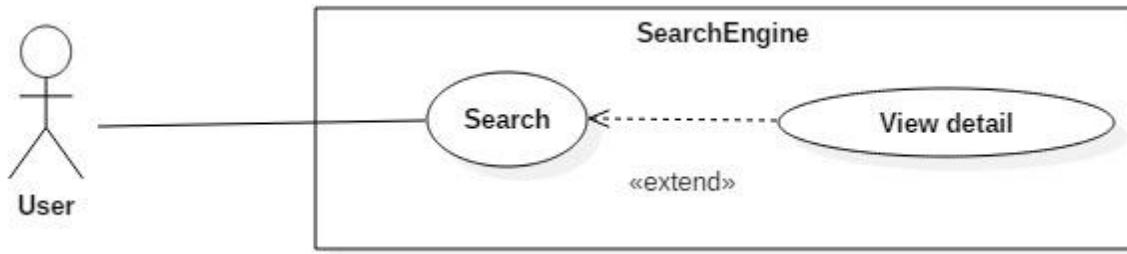
N/A

Relationships: extends UC "General Search"

Business Rules:

- Sorting criterias for each data type:
 - Hotel: price (ascending/descending), rating (ascending/descending)
 - Tour: price (ascending/descending)
 - Flight: price (ascending/descending), departure time (ascending/descending)
- When user deselect all the sorting criterias, the search results are sorted by RelevanceScore (mentioned in section D.7.4).

2.2.1.10. <User> View detail



USE CASE - <UC number>

Use Case No.	UC 1.10	Use Case Version	1.0
Use Case Name	View Detail		
Author	CuongHH		
Date	3/30/2016	Priority	Normal

Actor:

- User

Summary:

- This use case allows user view detail of a hotel/tour.

Goal:

- User can view detail information of hotel/tour along with the suggestion of relevant hotels and relevants tours.

Triggers:

- User clicks on a hotel/tour search result.

Preconditions:

- User is on the search result page of hotels or tours.

Post Conditions:

- **Success:** Open a new tab to view the detail page.
- **Fail:** Show error page.

Main Success Scenario:

Step	Actor Action	System Response
1	Click link on a search result.	Open new tab to view detail hotel/tour and relevant suggestion.

Alternative Scenario: N/A

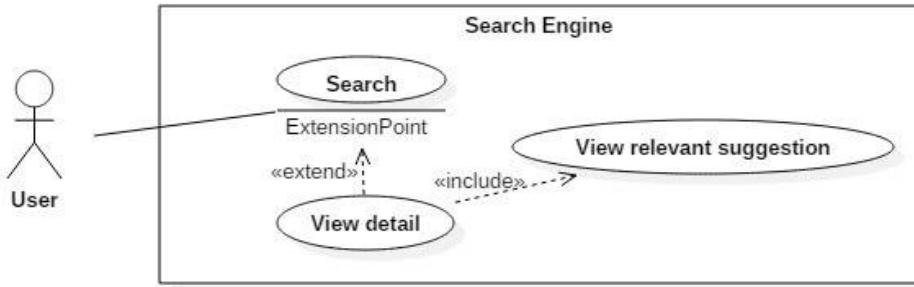
Exceptions: N/A

Relationships: extends UC "General Search"

Business Rules:

- Detail information for each data type:
 - Hotel: preview image, address, hotel name, full description, hotel services, comments, price
 - Tour: preview image, tour name, full description, tour route plan, price
- Relevant suggestion are rendered into 3 groups: hotels, tours, flights.

2.2.1.11. <User> View relevance suggestion



USE CASE - <UC number>

Use Case No.	UC 1.11	Use Case Version	1.0
Use Case Name	View relevance suggestion		
Author	TrucGS		
Date	3/30/2016	Priority	High

Actor:

- User

Summary:

- This use case allows user view relevance suggestion when view detail of a hotel/tour.

Goal:

- User can view the suggestion of relevant hotels/tours/flights to the current hotel/tour to get more options for their intent.

Triggers:

- User view detail of a hotel/tour.

Preconditions:

- User is on the detail page of hotels or tours.

Post Conditions:

- **Success:** Display relevant hotels/tours/flights.
- **Fail:** Show error page.

Main Success Scenario:

Step	Actor Action	System Response
1	View detail of a particular hotel/tour.	Find relevance suggestion and show results a 3 groups: hotels, tours, flights.

Alternative Scenario: N/A

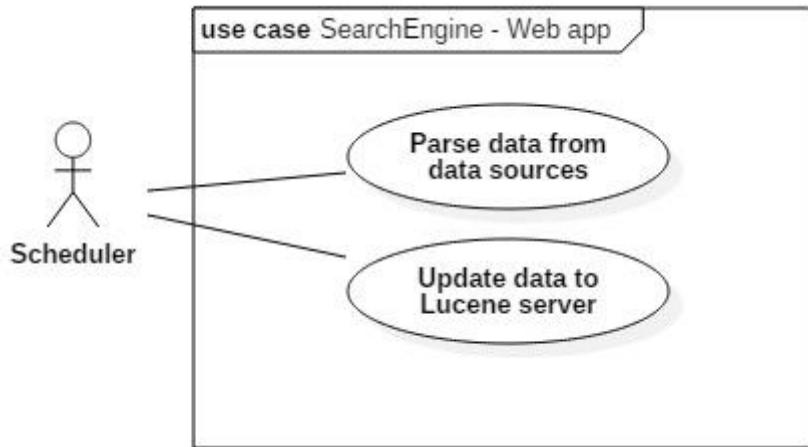
Exceptions: N/A

Relationships: included in UC "View Detail"

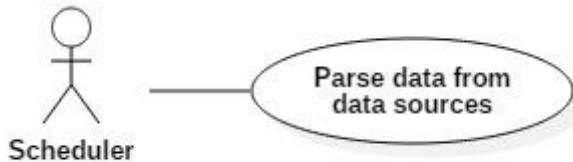
Business Rules:

- Relevant suggestion are:
 - Relevant hotels and relevant tours: use Pearson Correlation algorithm (mention in section D.7.6)
 - Relevant flights: the system will detect user's current location and find the nearest airport, then suggest 5 cheapest flights within 7 days from the found airport to the location of the selected tour/hotel. If there no flight found, the section of relevance flights will be hidden.

2.2.2.<Scheduler> Overview Use Case



2.2.2.1.<Scheduler> Parse data from data sources



USE CASE - <UC number>

Use Case No.	UC 2.1	Use Case Version	1.0
Use Case Name	Periodically parse data		
Author	TrungLVH		
Date	3/29/2016	Priority	High

Actor:

- Scheduler

Summary:

- This use case allows the scheduler to parse data automatically.

Goal:

- Automatically collect data from a particular external data source.

Triggers:

- The timer hits configured time.

Preconditions:

- The data source must be enabled.

Post Conditions:

- **Success:** The parsed data will be add/update to database.
- **Fail:** Error details will be logged in a log file.

Main Success Scenario:

Step	Actor Action	System Response
1	Automatically start after an interval of pending.	Crawl all the url of the data source and parse all the data available. [Exception 1]

Alternative Scenario: N/A

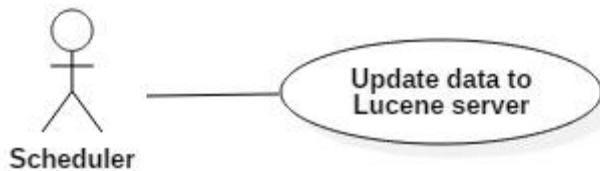
Exceptions:

[Exception 1]

No	Cause	System Response
1	Timeout request	Error details will be logged in a log file with the following structure: [Time]: [Scheduler Name] - [Error details]

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

- There are 2 type of parsing:
 - Html parsing (for hotels and tours): used to parse those data sources that rendered fully information in html without lazy loading.
 - Ajax parsing (for flights): used to parse those data sources that rendered partial information via Ajax request.
- The parsing process is mentioned in section D.7.1 and D.7.2.

2.2.2.2. <Scheduler> Update data to Lucene server**USE CASE - <UC number>**

Use Case No.	UC 2.2	Use Case Version	1.0
Use Case Name	Periodically update data in Lucene server		
Author	CuongHH		
Date	3/29/2016	Priority	High

Actor:

- Scheduler

Summary:

- This use case allows scheduler to update data from main storage to Lucene database periodically .

Goal:

- Data will be indexed and updated to Lucene database for searching with high performance.
-

Triggers:

- The timer hits configured time.
-

Preconditions:

- The scheduler is enabled.

Post Conditions:

- **Success:** Data will be updated to Lucene database.
- **Fail:** Error details will be logged in a log file.

Main Success Scenario:

Step	Actor Action	System Response
1	Automatically start after an interval of pending.	<ul style="list-style-type: none"> - Get all un-updated hotels/tours/flights from mainstorage. - Add/Override all the retrieved data to the corresponding data in Lucene database.

Alternative Scenario: N/A

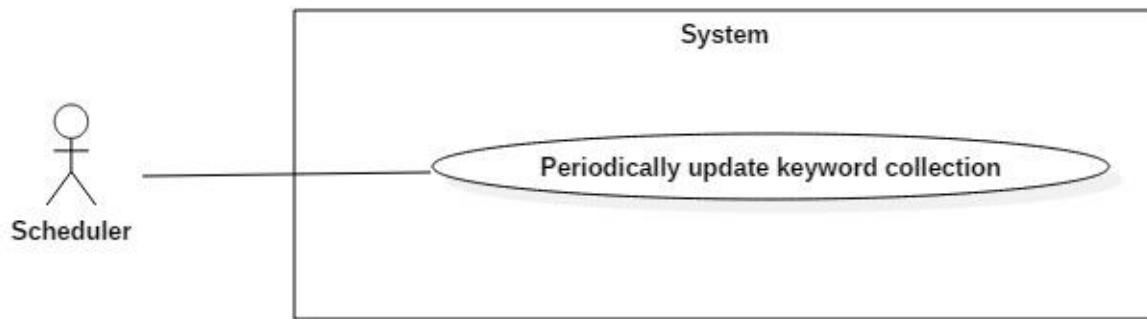
Exceptions: N/A

Relationships: N/A

Business Rules:

- Un-updated hotels/tours/flights are those data with the property:
 - IsLuceneUpdated = false.
- Hotel updating, Tour updating, Flight updating are 3 independent threads that execute concurrently.
- Data are indexed in Lucene using ShingleAnalyzer (works based on n-gram).

2.2.2.3. <Scheduler> Update keyword collection



USE CASE - <UC number>

Use Case No.	UC 2.3	Use Case Version	1.0
Use Case Name	Periodically update keyword collection		
Author	KhuongND		
Date	3/29/2016	Priority	High

Actor:

- Scheduler

Summary:

- This use case allows scheduler to update the keyword collection in Lucene database periodically.

Goal:

- Periodically update new keywords and remove redundant keywords from main storage and Lucene database to optimize the keyword suggestion.

Triggers:

- The timer hits configured time.
-

Preconditions:

- The scheduler is enabled.

Post Conditions:

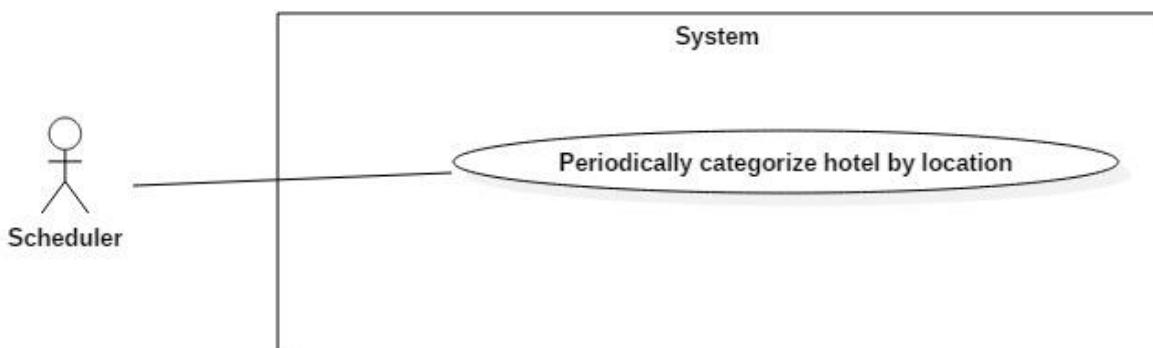
- **Success:** New keywords will be added; Redundant keywords will be removed.
- **Fail:** Error details will be logged in a log file.

Main Success Scenario:

Step	Actor Action	System Response
1	Automatically start after an interval of pending.	<ul style="list-style-type: none"> - Get all the new keywords from main storage and add to Lucene database. - Remove all the redundant keywords from main storage and Lucene database.

Alternative Scenario: N/A**Exceptions:** N/A**Relationships:** N/A**Business Rules:**

- New keywords are those keywords with the property:
 - IsLuceneUpdated = false
- Redundant keywords are found and removed as mentioned in section D.7.

2.2.2.4. <Scheduler> Categorize hotels by location**USE CASE - <UC number>**

Use Case No.	UC 2.4	Use Case Version	1.0
Use Case Name	Periodically categorize hotel by location		
Author	TrucGS		
Date	3/29/2016	Priority	Normal

Actor:

- Scheduler

Summary:

- This use case allows scheduler to categorize hotel by location periodically.

Goal:

- The parsed hotels will be categorized by location based on the hotel address.

Triggers:

- The timer hits configured time.

Preconditions:

- The scheduler is enabled.

Post Conditions:

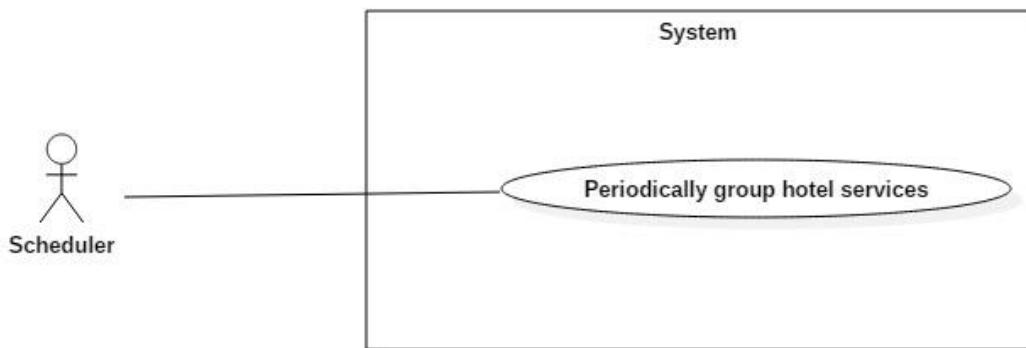
- **Success:** Hotels will be categorized by location.
- **Fail:** Error details will be logged in a log file.

Main Success Scenario:

Step	Actor Action	System Response
1	Automatically start after an interval of pending.	<ul style="list-style-type: none"> - Get all the un-located hotels from main storage. - Categorize those hotels using hotel address and the current location list.

Alternative Scenario: N/A**Exceptions:** N/A**Relationships:** N/A**Business Rules:**

- Un-located hotels are those hotels with the property: LocationID = null
- A hotel is supposed to be associated with the location having the highest score using Levenshtein Distance (mentioned in section D.7.8).

2.2.2.5. <Scheduler> Categorize hotel services**USE CASE - <UC number>**

Use Case No.	UC 2.5	Use Case Version	1.0
Use Case Name	Periodically group hotel services		
Author	TrucGS		
Date	3/29/2016	Priority	Normal

Actor:

- Scheduler

Summary:

- This use case allows scheduler to group hotel services periodically.

Goal:

- The parsed hotel services from different data sources will be grouped based on the similarity of service names.

Triggers:

- The timer hits configured time.

Preconditions:

- The scheduler is enabled.

Post Conditions:

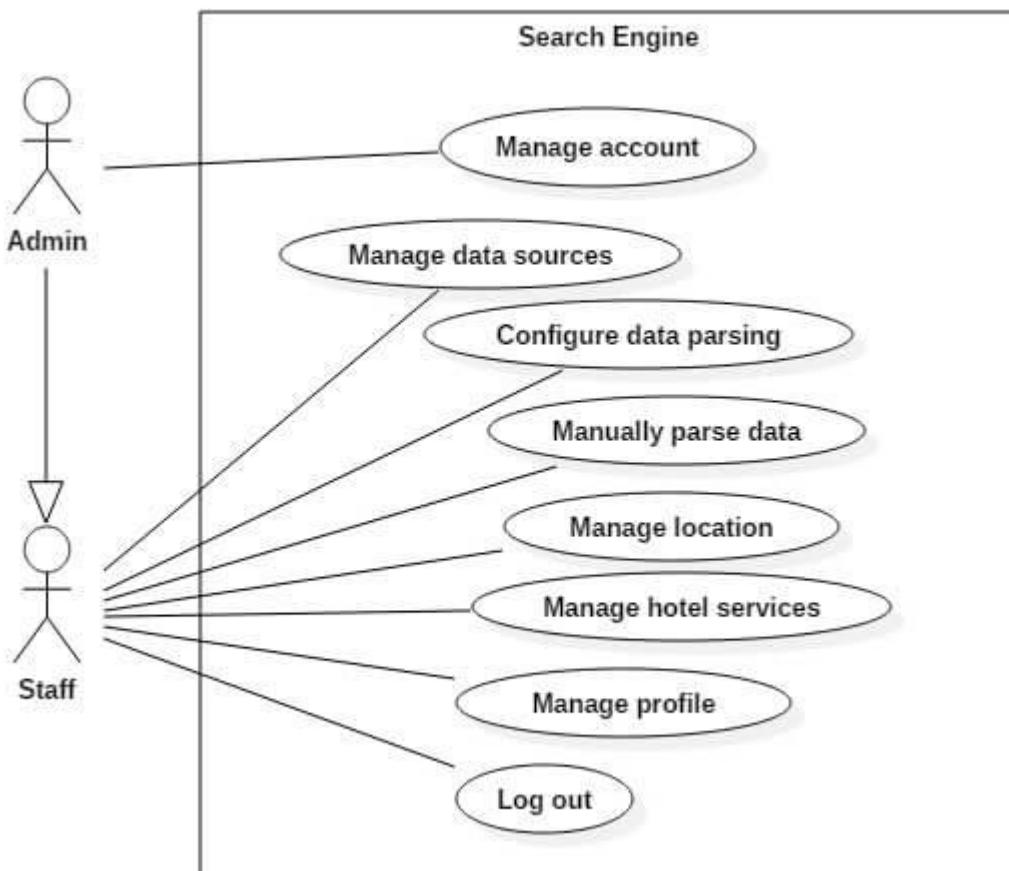
- **Success:** Hotel services will be grouped to the most relevant services.
- **Fail:** Error details will be tracked in a log file.

Main Success Scenario:

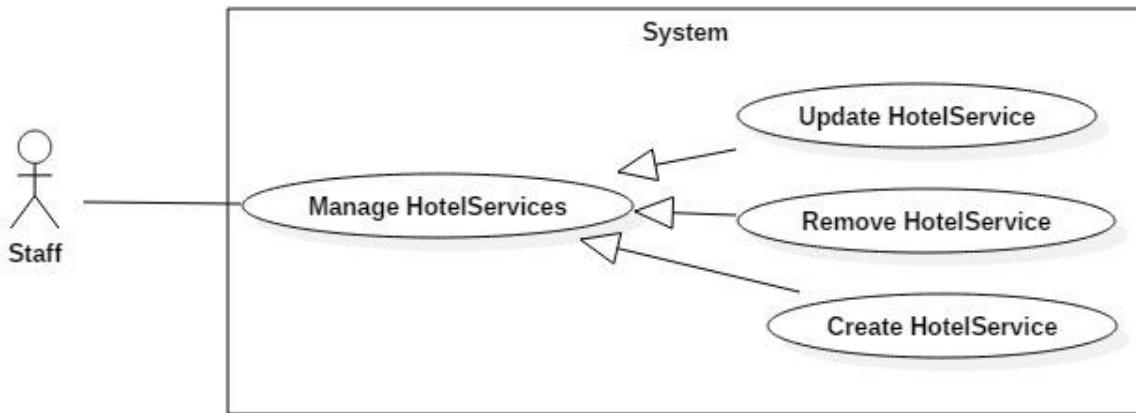
Step	Actor Action	System Response
1	Automatically start after an interval of pending.	Group all the un-categorized hotel services into the most relevant group.

Alternative Scenario: N/A**Exceptions:** N/A**Relationships:** N/A**Business Rules:**

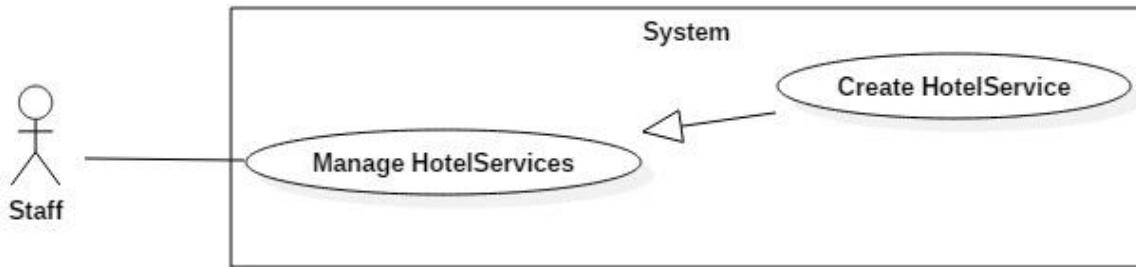
- Un-categorized hotel services are those services with the property: IsCategorized = false
- A hotel service will be temporarily categorized to the most relevant service group (based on ServiceName and Synonyms) and need the approval of admin/staff to be the official service (mentioned in section D.7.9).

2.2.3. <Staff> Overview Use Case

2.2.3.1. <Staff> Manage HotelService



2.2.3.1.1. < Staff > Create HotelService



USE CASE - <UC number>

Use Case No.	UC 3.1	Use Case Version	1.0
Use Case Name	Create HotelService		
Author	KhuongND		
Date	3/29/2016	Priority	Normal
Actor:			
- Staff			
Summary:			
- This use case allows staff to create new hotel service.			
Goal:			
- User can create new hotel services using for filtering hotel search results.			
Triggers:			
- User sends the create new hotel service command.			
Preconditions:			
- User logged in to the system with role staff (status in-active) or admin.			
Post Conditions:			
- Success: New hotel service is create into database and show success message			
- Fail: Show error message			

Main Success Scenario:

Step	Actor Action	System Response
1	Staff go to “Manage hotel services” page and choose “Services” tab.	System list following information in “Services” tab. - Name: textbox,max length:250, required. - Synonyms: textbox. - Icon plus: button sends create new hotel services command.
2	Staff fills in information of hotel services.	
3	Staff sends command to create new hotel services with inputted information to system(Click Icon plus).	New hotel services will be created into database. Show success message: “Created successfully!” and back “Manage hotel services” page. [Exception 1]

Alternative Scenario:N/A

Exceptions:

[Exception 1]

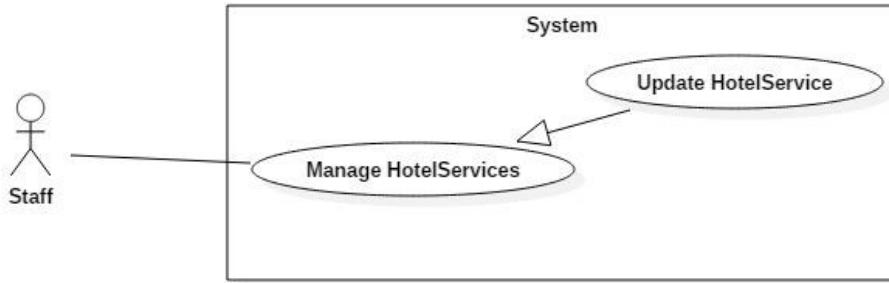
No	Cause	System Response
1	“Name” is empty or longer than 250 words.	Show error message: “Please enter name hotel service less than 250 words”.

Relationships: N/A

Business Rules:

- Name: textbox, max length:250, required.
- Synonyms: textbox.
- Icon plus: button sends create new hotel services command.
- After create new hotel service, data table will be auto update.

2.2.3.1.2. < Staff > Update HotelService



USE CASE - <UC number>

Use Case No.	UC 3.2	Use Case Version	1.0
Use Case Name	Update HotelService		
Author	KhuongND		
Date	3/29/2016	Priority	Normal

Actor:

- Staff

Summary:

- This use case allows staff to update existed hotel service.

Goal:

- Update existed hotel service.

Triggers:

- Staff sends the update hotel service information command.

Preconditions:

- User logged in to the system with role staff (status in-active) or admin.
- Staff selects a hotel service exist in “Manage hotel service” page.

Post Conditions:

- **Success:** Hotel service information will be updated and show success message
- **Fail:** Show error message

Main Success Scenario:

Step	Actor Action	System Response
1	Staff go to “hotel service” page, choose “Service” or “New Service”, click link “Edit”	<p>System show modal following information:</p> <ul style="list-style-type: none"> - Name: textbox,max length:250, required. - Synonyms: textbox. - If choose hotel service in “New Services” tab to update, modal will add a field: <ul style="list-style-type: none"> + Expected Service: dropdownlist,(all hotel service with “IsApproved” by “true” in database),required. - Button “Save Changes” sends update data sources command. - Butten “Close” cancels update process.
2	Staff fills in updated information of hotel service.	

3	Staff click on “Save Changes” button. [Alternative 1]	Hotel service will be updated to database. [Exception 1,2]
---	--	---

Alternative Scenario:

[Alternative 1]

Step	Actor Action	System Response
1	Staff click on “Close” button.	System close modal.

Exceptions:

[[Exception 1]

No	Cause	System Response
1	“Name” is empty or longer than 250 words.	Show error message: “Please enter name hotel service less than 250 words”.

[Exception 2]

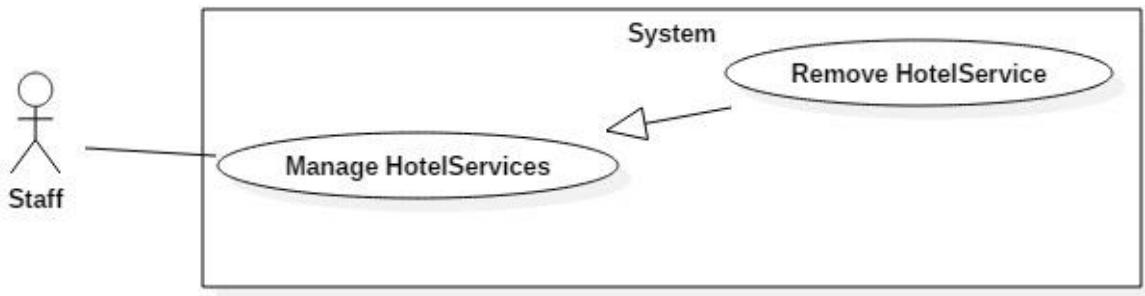
No	Cause	System Response
3	Hotel service which the staff chose dont exist (another staff remove when update).	Show error message: “Hotel service not found.”.

Relationships: N/A

Business Rules:

- A “Hotel service” can have one or more “Synonyms”.

2.2.3.1.3. < Staff > Remove HotelService



USE CASE - <UC number>

Use Case No.	UC 3.3	Use Case Version	1.0
Use Case Name	Remove HotelService		
Author	KhuongND		
Date	3/29/2016	Priority	Normal

Actor:

- Staff

Summary:

- This use case allows staff to remove hotel service.

Goal:

- System removes hotel service.

Triggers:

- Staff sends the remove hotel service command.

Preconditions:

- User logged in to the system with role staff (status in-active) or admin.
- Staff selects a hotel service exist in "Manage hotel service" page.

Post Conditions:

- **Success:** Hotel service will be removed and show success message.
- **Fail:** Show error message

Main Success Scenario:

Step	Actor Action	System Response
1	Staff go to "Hotel Service" page choose "Service" or "New Service", click link "Remove"	-System confirm with staff to remove hotel service.
2	Staff confirms "Yes" to remove hotel service. [Alternative 1]	Hotel service will be removed to database. [Exception 1]

Alternative Scenario:

[Alternative 1]

Step	Actor Action	System Response
1	Staff click on "Cancel" button.	

		System close confirm, cancel remove process.
--	--	--

Exceptions:

[Exception 1]

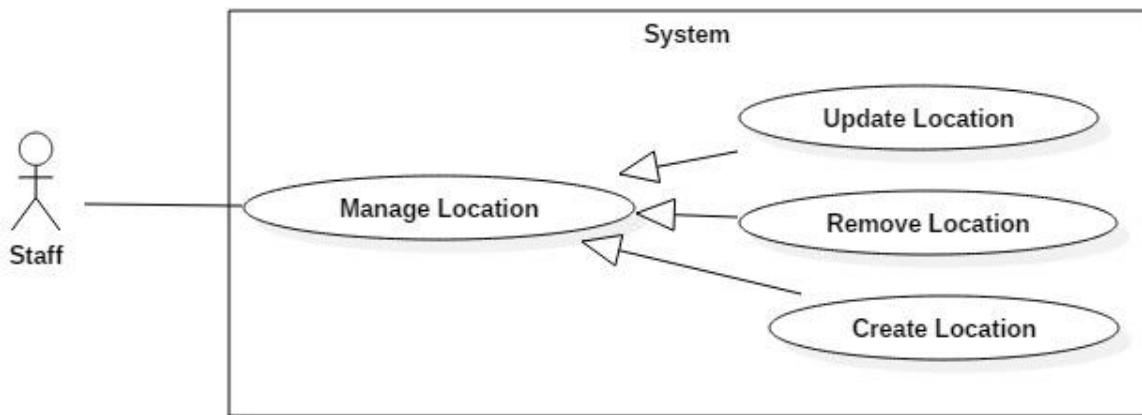
No	Cause	System Response
1	Hotel service which the staff chose dont exist (another staff remove when remove).	Show error message: "Hotel service not found.".

Relationships: N/A

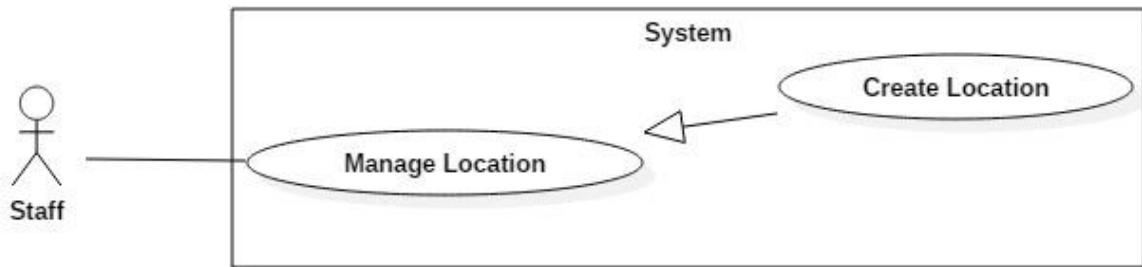
Business Rules:

- If remove hotel service success, a "Hotel service" will be removed in database immediately.
- List "Hotel service" in "Manage Hotel Service" page will be updated after this function finished.
- If this "Hotel service" is being used, system will not remove this "Hotel service".
- Before remove a "Hotel service", system must find this "Hotel service" in database exist or not? then removed if exist to avoid errors.

2.2.3.2. < Staff > Manage Location



2.2.3.2.1. < Staff > Create Location



USE CASE - <UC number>			
Use Case No.	UC 3.4	Use Case Version	1.0
Use Case Name	Create Location		
Author	KhuongND		
Date	3/29/2016	Priority	Normal
Actor:	<ul style="list-style-type: none"> - Staff 		
Summary:	<ul style="list-style-type: none"> - This use case allows staff to create new location. 		
Goal:	<ul style="list-style-type: none"> - User can create new location. 		
Triggers:	<ul style="list-style-type: none"> - User sends the create new location command. 		
Preconditions:	<ul style="list-style-type: none"> - User logged in to the system with role staff (status in-active) or admin. 		
Post Conditions:	<ul style="list-style-type: none"> - Success: New location is create into database and show success message - Fail: Show error message 		
Main Success Scenario:			
Step	Actor Action	System Response	
1	Staff go to "Manage location" page and click "Add New" button.	<p>System show modal following information:</p> <ul style="list-style-type: none"> - ShortName: textbox, max length:50. - FullName: : textbox, required - Location type: dropdownlist (Location, Airport), required. - If choose "Location type" with value "Airport", modal will add a field: - + Location: dropdownlist,(all location with "Location type" by "Location" in database),required - Synonyms: textbox. - Button "Add New" sends create new location command. - Button "Close" cancel create process. 	
2	Staff fills in information location.		
3	Staff sends command to create new location with inputted information to system(Click "Add New" button). [Alternative 1]	New location will be created into database. [Exception 1,2]	

Alternative Scenario:**[Alternative 1]**

Step	Actor Action	System Response
1	Staff click "Close" button to cancel process	System close modal and cancel create process.

Exceptions:**[Exception 1]**

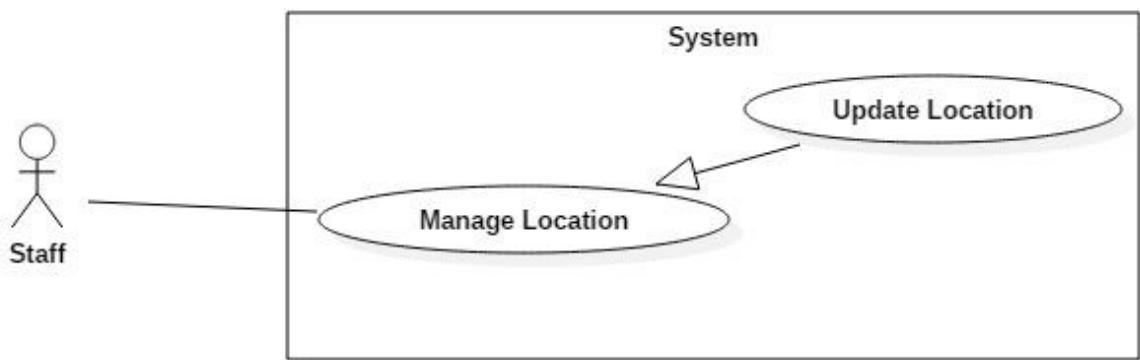
No	Cause	System Response
1	"FullName" is empty.	Show error message: "Please enter FullName".

[Exception 2]

No	Cause	System Response
2	"ShortName" is longer than 50 words.	Show error message: "Please enter ShortName location less than 50 words".

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

- List location in "Manage location" page will be updated with new location.
- List "Location" and list "Airport" will be using a lot of other functions in system .
- When this function finish, a "Location" or "Airport" will be added in "Location" table in database immediately.
- A "Location" will have more "Synonyms", all synonyms of this "Location" will save 1 field in database and separated by commas.

2.2.3.2.2. < Staff > Update Location

USE CASE - <UC number>			
Use Case No.	UC 3.5	Use Case Version	1.0
Use Case Name	Update Location		
Author	KhuongND		
Date	3/29/2016	Priority	Normal

Actor:

- Staff

Summary:

- This use case allows staff to update existed location.

Goal:

- Update existed location.

Triggers:

- Staff sends the update location information command.

Preconditions:

- User logged in to the system with role staff (status in-active) or admin.
- Staff selects a location exist in "Manage location" page.

Post Conditions:

- **Success:** Location information will be updated and show success message
- **Fail:** Show error message

Main Success Scenario:

Step	Actor Action	System Response
1	Staff go to "Location Manager" page and click link "Edit"	<p>System show modal following information:</p> <ul style="list-style-type: none"> - ShortName: textbox, max length:50. - FullName: : textbox, required - Location type: dropdownlist (Location, Airport), required. - If choose "Location type" with value "Airport", modal will add a field: <ul style="list-style-type: none"> + Location: dropdownlist,(all location with "Location type" by "Location" in database),required. - Synonyms: textbox. - Button "Save Changes" sends create new location command. - Button "Close" cancel create process.
2	Staff fills in updated information of location.	
3	Staff click on "Save Changes" button. [Alternative 1]	<p>Location will be updated to database. [Exception 1,2,3]</p>

Alternative Scenario:

[Alternative 1]

Step	Actor Action	System Response
------	--------------	-----------------

1	Staff click on “Close” button.	System close modal.
---	--------------------------------	---------------------

Exceptions:

[Exception 1]

No	Cause	System Response
1	“FullName” is empty.	Show error message: “Please enter FullName”.

[Exception 2]

No	Cause	System Response
2	“ShortName” is longer than 50 words.	Show error message: “Please enter ShortName location less than 50 words”.

[Exception 4]

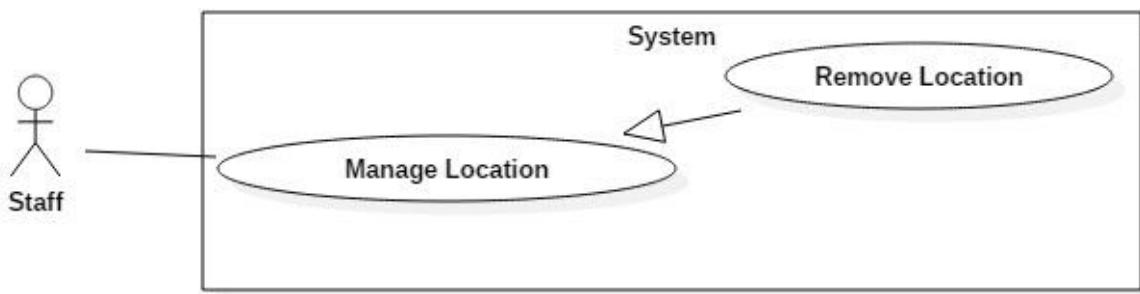
No	Cause	System Response
3	Location which the staff chose dont exist (another staff remove when update).	Show error message: “Location not found.”.

Relationships: N/A

Business Rules:

- If update location success, a “Location” will be updated in database immediately.
- List “Location” in “Manage location” page will be updated after this function finished.
- A “Location” will have more “Synonyms”, all synonyms of this “Location” will update 1 field in database and separated by commas.
- Before update a “Location”, system must find this “Location” in database exist or not? then updated if exist to avoid errors.

2.2.3.2.3. < Staff > Remove Location



USE CASE - <UC number>			
Use Case No.	UC 3.6	Use Case Version	1.0
Use Case Name	Remove Location		
Author	KhuongND		
Date	3/29/2016	Priority	Normal

Actor:

- Staff

Summary:

- This use case allows staff to remove location.

Goal:

- System removes location.

Triggers:

- Staff sends the remove location command.

Preconditions:

- User logged in to the system with role staff (status in-active) or admin.
- Staff selects a location exist in "Manage location" page.

Post Conditions:

- **Success**: Location will be removed and show success message.
- **Fail**: Show error message

Main Success Scenario:

Step	Actor Action	System Response
1	Staff go to "Location Manager" page and click link "Remove"	-System confirm with staff to remove location.
2	Staff confirms "Yes" to remove location. [Alternative 1]	Location will be removed to database. [Exception 1]

Alternative Scenario:

[Alternative 1]

Step	Actor Action	System Response
1	Staff click on "Cancel" button.	System close confirm, cancel remove process.

Exceptions:

[Exception 1]

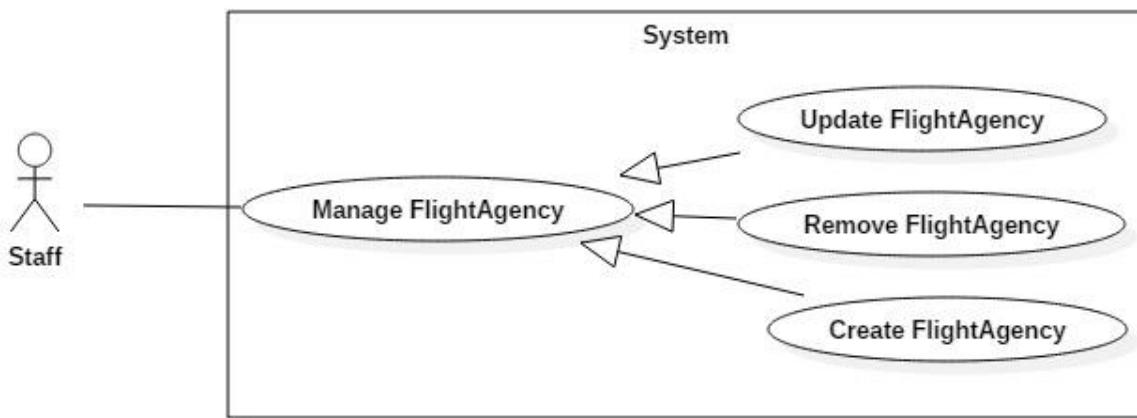
No	Cause	System Response
1	Location which the staff chose dont exist (another staff remove when remove).	Show error message: "Location not found.".

Relationships: N/A

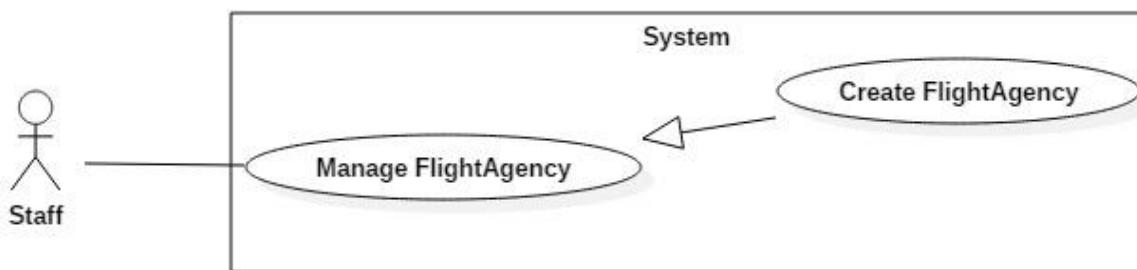
Business Rules:

- If remove location success, a "Location" will be removed in database immediately.
- List "Location" in "Manage Location" page will be updated after this function finished.
- If this "Location" is being used, system will not remove this "Location".
- Before remove a "Location", system must find this "Location" in database exist or not? then removed if exist to avoid errors.

2.2.3.3. < Staff > Manage FlightAgency



2.2.3.3.1. < Staff > Create FlightAgency



USE CASE - <UC number>

Use Case No.	UC 3.7	Use Case Version	1.0
Use Case Name	Create FlightAgency		
Author	TrungLVH		
Date	3/29/2016	Priority	Normal

Actor:

- Staff

Summary:

- This use case allows staff to create new flight agency.

Goal:

- User can create new flight agency.

Triggers:

- User sends the create flight agency command.

Preconditions:

- User logged in to the system with role staff (status in-active) or admin.

Post Conditions:

- **Success:** New flight agency is create into database and show success message
- **Fail:** Show error message

Main Success Scenario:

Step	Actor Action	System Response
1	Staff go to "Manage flight agency" page and click "Add New" button.	System show modal following information: <ul style="list-style-type: none"> - Name: textbox, max length:250, required. - Logo: fileupload, extension of fileupload contain [jpg,jpeg,png,gif]. - Button "Add New" sends create new flight agency command. - Button "Close" cancel create process.
2	Staff fills in information of flight agency.	
3	Staff sends command to create new flight agency with inputted information to system(Click "Add New" button). [Alternative 1]	New flight agency will be created into database. [Exception 1,2]

Alternative Scenario:

Step	Actor Action	System Response
1	Staff click "Close" button to cancel process	System close modal and cancel create process.

Exceptions:

Exception 1]

No	Cause	System Response
1	"Name" is empty or longer than 250 words.	Show error message: "Please enter name flight agency less than 250 words".

[Exception 2]

No	Cause	System Response
2	"Logo" doesn't have true extension [jpg,jpeg,png,gif].	Show error message: "Please choose

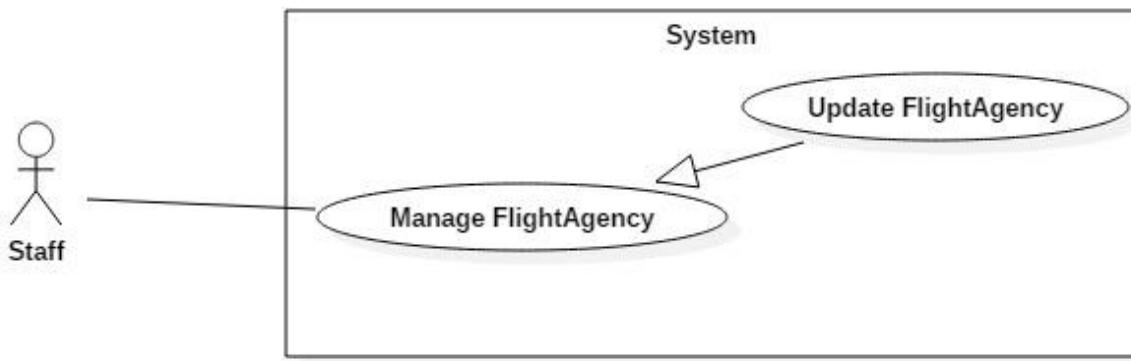
	image for logo ”.
--	-------------------

Relationships: N/A

Business Rules:

- “Logo” must type image (jpg/jpeg/png/gif), check type before added.
- List flight agency in “Manage flight agency” page will be updated with new flight agency after this function finished.
- When this function finish, a “Flight agency” will be added in “FlightAgency” table in database immediately.

2.2.3.3.2. < Staff > Update FlightAgency



USE CASE - <UC number>

Use Case No.	UC 3.8	Use Case Version	1.0
Use Case Name	Update flight agency		
Author	TrungLVH		
Date	3/29/2016	Priority	Normal

Actor:

- Staff

Summary:

- This use case allows staff to update existed flight agency.

Goal:

- Update existed flight agency.

Triggers:

- Staff sends the update flight agency information command.

Preconditions:

- User logged in to the system with role staff (status in-active) or admin.
- Staff selects a flight agency exist in “Manage flight agency” page.

Post Conditions:

- **Success:** Flight agency information will be updated and show success message
- **Fail:** Show error message

Main Success Scenario:

Step	Actor Action	System Response
1	Staff go to “Flight Agency”	

	Manager" page and click "Edit".	System show modal following information: - Name: textbox, max length:250, required. - Logo: fileupload, extension of fileupload contain [jpg,jpeg,png,gif]. - Button "Save Changes" sends update flight agency command. - Button "Close" cancel create process.
2	Staff fills in updated information of flight agency.	
3	Staff click on "Save Changes" button. [Alternative 1]	Flight agency will be updated to database. [Exception 1,2,3]

Alternative Scenario:

[Alternative 1]

Step	Actor Action	System Response
1	Staff click on "Close" button.	System close modal.

Exceptions:

[Exception 1]

No	Cause	System Response
1	"Name" is empty or longer than 250 words.	Show error message: "Please enter name flight agency less than 250 words".

[Exception 2]

No	Cause	System Response
2	"Logo" doesn't have true extension [jpg,jpeg,png,gif].	Show error message: "Please choose image for logo".

[Exception 3]

No	Cause	System Response
3	Flight agency which the staff chose dont exist (another staff remove when update).	Show error message: "Flight agency not found.".

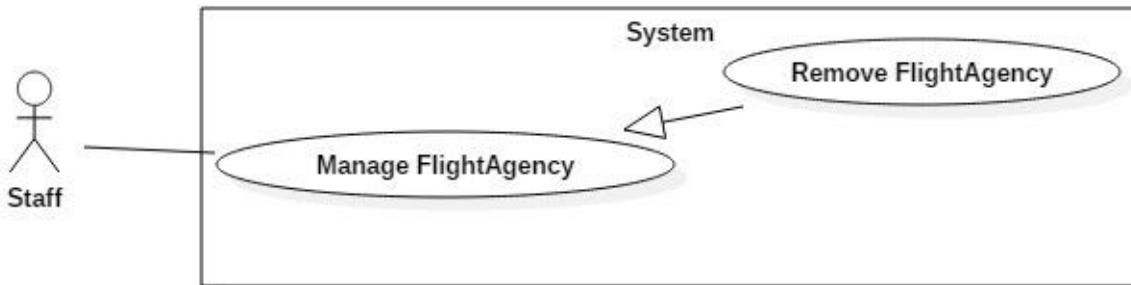
Relationships: N/A

Business Rules:

- "Logo" must have contension image [jpg,jpeg,png,gif], check type image before updated.
- If update flight agency success, a "flight agency" will be updated in database

- immediately.
- List “Flight agency” in “Manage Flight Agency” page will be updated after this function finished.
 - Before update a “Flight agency”, system must find this “Flight agency” in database exist or not? then updated if exist to avoid errors.

2.2.3.3.3. < Staff > Remove FlightAgency



USE CASE - <UC number>

Use Case No.	UC 3.9	Use Case Version	1.0
Use Case Name	Remove FlightAgency		
Author	TrungLVH		
Date	3/29/2016	Priority	Normal

Actor:

- Staff

Summary:

- This use case allows staff to remove flight agency.

Goal:

- System removes flight agency.

Triggers:

- Staff sends the remove flight agency command.

Preconditions:

- User logged in to the system with role staff (status in-active) or admin.
- Staff selects a flight agency exist in “Manage flight agency” page.

Post Conditions:

- **Success:** Flight agency will be removed and show success message.
- **Fail:** Show error message

Main Success Scenario:

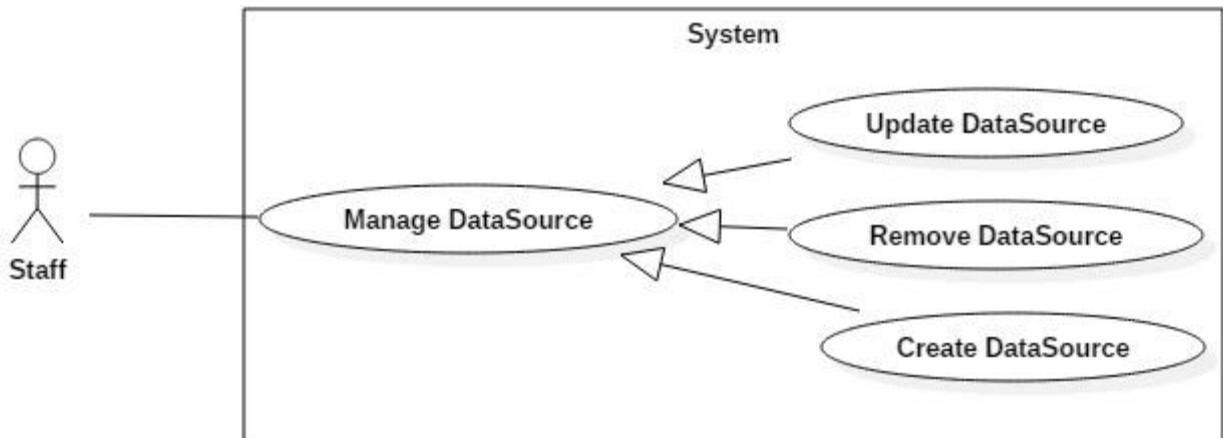
Step	Actor Action	System Response
1	Staff go to “Flight Agency Manager” page and click link “Remove”.	-System confirm with staff to remove flight agency.
2	Staff confirms “Yes” to remove flight agency . [Alternative 1]	Flight agency will be removed to database.

		[Exception 1]
Alternative Scenario:		
[Alternative 1]		
Step	Actor Action	System Response
1	Staff click on "Cancel" button.	System close confirm, cancel remove process.

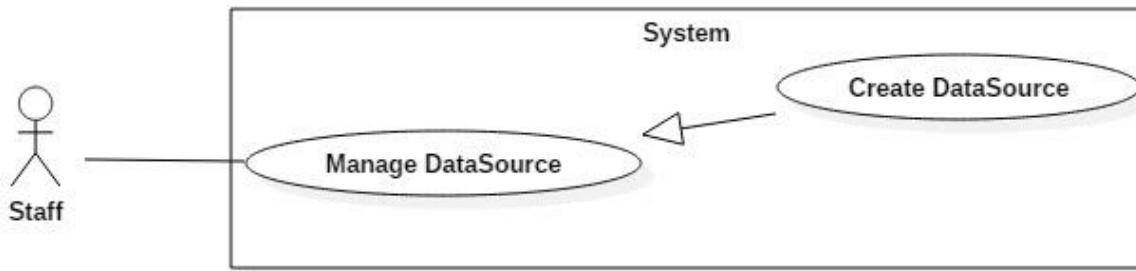
Exceptions:		
[Exception 1]		
No	Cause	System Response
1	Flight agency which the staff chose dont exist (another staff remove when remove).	Show error message: "flight agency not found.".

Relationships: N/A		
Business Rules:		
<ul style="list-style-type: none"> - If remove flight agency success, a "Flight agency" will be removed in database immediately. - List "Flight agency" in "Manage Flight Agency" page will be updated after this function finished. 		

2.2.3.4. < Staff > Manage DataSource



2.2.3.4.1. < Staff > Create DataSource



Web Application Use Case – < Staff > Create DataSource

USE CASE - <UC number>			
Use Case No.	UC 3.10	Use Case Version	1.0
Use Case Name	Create data source		
Author	KhuongND		
Date	3/29/2016	Priority	Normal
Actor:	<ul style="list-style-type: none"> - Staff 		
Summary:	<ul style="list-style-type: none"> - This use case allows staff to create new data sources. 		
Goal:	<ul style="list-style-type: none"> - User can create new data sources. 		
Triggers:	<ul style="list-style-type: none"> - User sends the create new data sources command. 		
Preconditions:	<ul style="list-style-type: none"> - User logged in to the system with role staff (status in-active) or admin. 		
Post Conditions:	<ul style="list-style-type: none"> - Success: New data sources is create into database and show success message - Fail: Show error message 		
Main Success Scenario:			
Step	Actor Action	System Response	
1	Staff go to “Manage data sources”.	System navigates to “Manage data sources” page. <ul style="list-style-type: none"> - Data type: dropdownlist (Hotel, Flight,Tour), required. - Data source URL: textbox, contain “http” or “https”, required. - Icon plus: button sends create new data sources command. 	
2	Staff fills in information of datasources.		
3	Staff sends command to create		

	new data sources with inputted information to system(Click Icon plus).	New data sources will be created into database. [Exception 1]
--	--	--

Alternative Scenario:N/A

Exceptions:

[Exception 1]

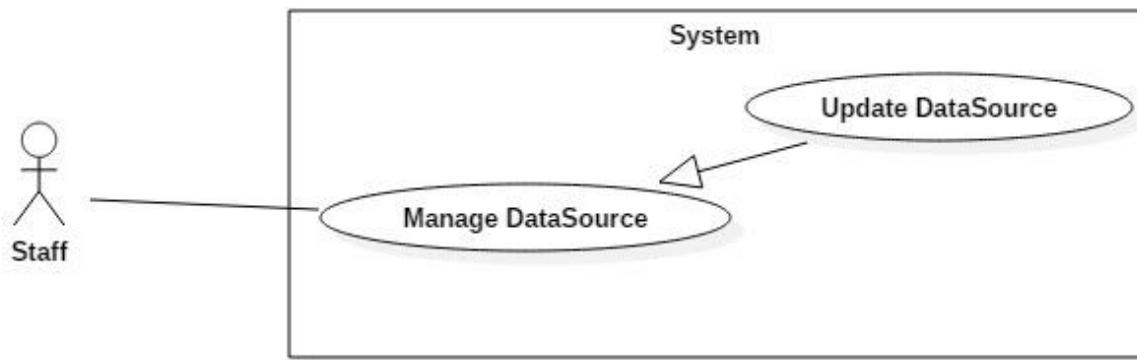
No	Cause	System Response
1	“Data source URL” is empty or doesn't contain “http” or “https”.	Show error message: “Please enter data source URL start with “http” or “https””.

Relationships: N/A

Business Rules:

- List data sources in “Manage data sources” page will be updated with new data sources, after this function finished.
- “Data source URL” must start with “http” or “https”.

2.2.3.4.2. < Staff > Update DataSource



Web Application Use Case - < Staff > Update DataSource

USE CASE - <UC number>			
Use Case No.	UC 3.11	Use Case Version	1.0
Use Case Name	Update datasource		
Author	KhuongND		
Date	3/29/2016	Priority	Normal
Actor:	<ul style="list-style-type: none"> - Staff 		
Summary:	<ul style="list-style-type: none"> - This use case allows staff to update existed datasource. 		
Goal:	<ul style="list-style-type: none"> - Update existed datasource. 		
Triggers:	<ul style="list-style-type: none"> - Staff sends the update datasource information command. 		

Preconditions:

- User logged in to the system with role staff (status in-active) or admin.
- Staff selects a datasource exist in “Manage data sources” page.

Post Conditions:

- **Success:** Datasource information will be updated and show success message
- **Fail:** Show error message

Main Success Scenario:

Step	Actor Action	System Response
1	Staff go to “Manage data sources” and click “Edit”	<p>System show modal following information:</p> <ul style="list-style-type: none"> - Data type: dropdownlist (Hotel, Flight,Tour), required. - Data source URL: textbox, contain “http” or “https”, required. - Button “Save Changes” sends update data sources command. - Butten “Close” cancels update process.
2	Staff fills in updated information of datasources.	
3	Staff click on “Save Changes” button. [Alternative 1]	Data sources will be updated to database. [Exception 1,2]

Alternative Scenario:

[Alternative 1]

Step	Actor Action	System Response
1	Staff click on “Close” button.	System close modal.

Exceptions:

[Exception 1]

No	Cause	System Response
1	“Data source URL” is empty or doesnt contain “http” or “https”.	Show error message: “Please enter data source URL start with “http” or “https””.

[Exception 2]

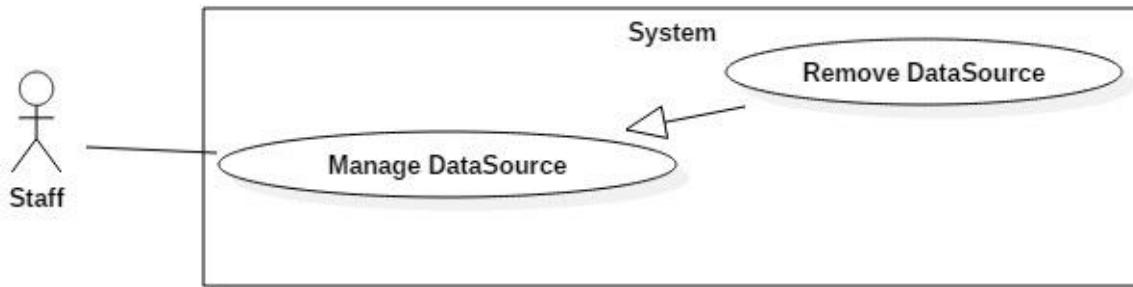
No	Cause	System Response
3	Data source which the staff chose dont exist (another staff remove when update).	Show error message: “Data source not found.”.

Relationships: N/A

Business Rules:

- “Data source URL” must start with “http” or “https”, check before updated.
- If update datasource success, a “Datasource” will be updated in database immediately.
- List “Datasource” in “Manage Data Source” page will be updated after this function finished.

2.2.3.4.3. < Staff > Remove DataSource



USE CASE - <UC number>

Use Case No.	UC 3.12	Use Case Version	1.0
Use Case Name	Remove data source		
Author	KhuongND		
Date	3/29/2016	Priority	Normal

Actor:

- Staff

Summary:

- This use case allows staff to remove datasource.

Goal:

- System removes datasource.

Triggers:

- Staff sends the remove datasource command.

Preconditions:

- User logged in to the system with role staff (status in-active) or admin.
- Staff selects a datasource exist in “Manage data sources” page.

Post Conditions:

- **Success:** Datasource will be removed and show success message.
- **Fail:** Show error message

Main Success Scenario:

Step	Actor Action	System Response
1	Staff go to “Manage data sources” and click “Remove”	-System confirm with staff to remove datasource
2	Staff confirms “Yes” to remove data source.	

	[Alternative 1]	Data source will be removed to database. [Exception 1]
--	-----------------	---

Alternative Scenario:

[Alternative 1]

Step	Actor Action	System Response
1	Staff click on “Cancel” button.	System close confirm, cancel remove process.

Exceptions:

[Exception 1]

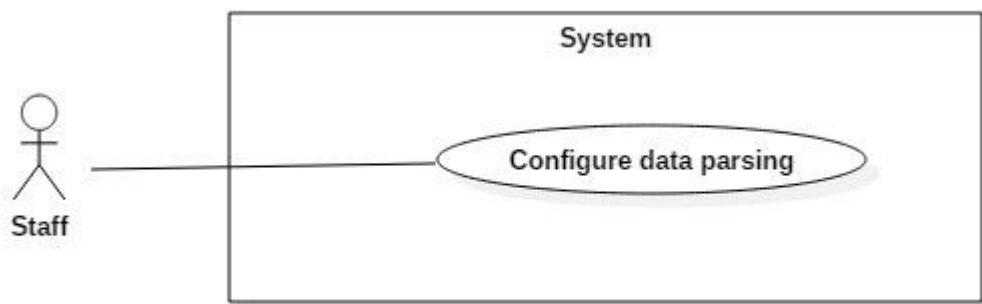
No	Cause	System Response
1	Data source which the staff chose dont exist (another staff remove when remove).	Show error message: “Data source not found.”.

Relationships: N/A

Business Rules:

- If remove datasource success, a “Datasource” will be removed in database immediately.
- List “Datasource” in “Manage Datasource” page will be updated after this function finished.
- If this “Datasource” is being used, system will not remove this “Datasource”.
- If data source have more configurations, system will remove all those configurations before remove data source.

2.2.3.5. <Staff> Configure data parsing

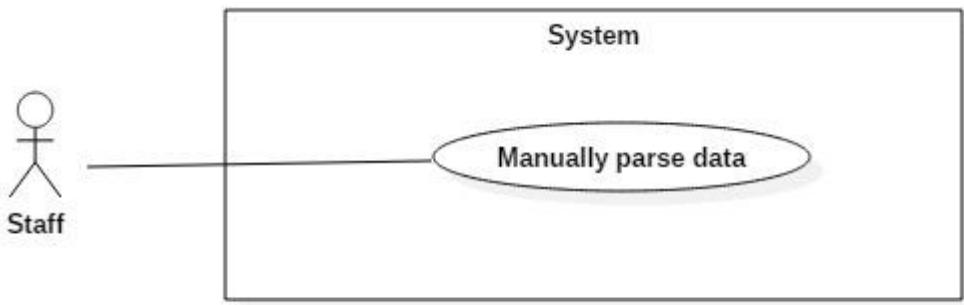


Web Application Use Case - < Staff > Configure data parsing

USE CASE - <UC number>			
Use Case No.	UC 3.13	Use Case Version	1.0
Use Case Name	Configure data parsing		
Author	TrucGS		

Date	3/29/2016	Priority	High		
Actor:					
- Staff					
Summary:					
- This use case allows staff to configure data parsing for a particular data source.					
Goal:					
- All the configuration created are used for checking html structure and parse the corresponding data from data sources.					
Triggers:					
- User sends the configure data parsing command.					
Preconditions:					
- User must login with role Staff (status "Active") or Admin.					
Post Conditions:					
- Success: The configuration are saved to database.					
- Fail: Show error message.					
Main Success Scenario:					
Step	Actor Action	System Response			
1	Choose a data source and click on button "Config" in "Manage data sources" page .	Show list of properties depending on type of the data source.			
2	Fill in configuration properties by selecting the data area in iframe that contains html of the sample page to get CSS selector of the element.				
3	Click on "Save Changes" button.	Data parsing configuration saved to the database.			
Alternative Scenario: N/A					
Exceptions: N/A					
Relationships: N/A					
Business Rules:					
<ul style="list-style-type: none"> - The configuration properties are different for each data type: <ul style="list-style-type: none"> • Hotel: Name, PreviewImgUrl, Address, Description, DescriptionHtml, Price, Rating, ProvidedServices, Comments • Tour: Name, PreviewImgUrl, Location, Agency, Description, DescriptionHtml, Route, RouteHtml, Price • Flight: MinProxyCount, MaxThreadCount, MaxResponse, MaxAttemp 					

2.2.3.6. <Staff> Manually parse data

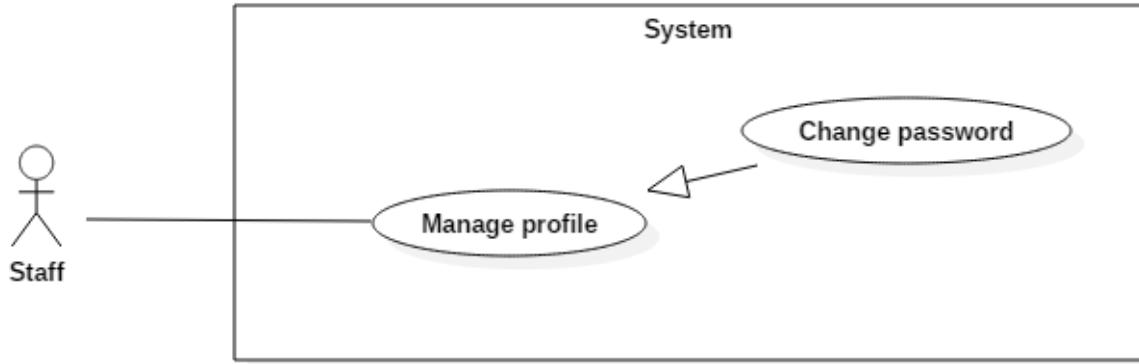


Web Application Use Case - < Staff > Manually parse data

USE CASE - <UC number>									
Use Case No.	UC 3.14	Use Case Version	1.0						
Use Case Name	Manually parse data								
Author	TrucGS								
Date	3/29/2016	Priority	Normal						
Actor:	<ul style="list-style-type: none"> - Staff 								
Summary:	<ul style="list-style-type: none"> - This use case allows staff to manually parse data. 								
Goal:	<ul style="list-style-type: none"> - User can manually parse data. 								
Triggers:	<ul style="list-style-type: none"> - User sends the manually parse data command. 								
Preconditions:	<ul style="list-style-type: none"> - User must login with role Staff (status “Active”) or Admin. 								
Post Conditions:	<ul style="list-style-type: none"> - Success: Parsing data process of the selected data source will start and run in background. - Fail: Show error message. 								
Main Success Scenario:	<table border="1"> <thead> <tr> <th>Step</th><th>Actor Action</th><th>System Response</th></tr> </thead> <tbody> <tr> <td>1</td><td>Choose a datasource and click “Force Start” button in “Manage data sources” page .</td><td>Start parsing process and show message: “Parsing progress has been started”.</td></tr> </tbody> </table>			Step	Actor Action	System Response	1	Choose a datasource and click “Force Start” button in “Manage data sources” page .	Start parsing process and show message: “Parsing progress has been started”.
Step	Actor Action	System Response							
1	Choose a datasource and click “Force Start” button in “Manage data sources” page .	Start parsing process and show message: “Parsing progress has been started”.							
Alternative Scenario:	N/A								
Exceptions:	N/A								
Relationships:	N/A								
Business Rules:	<ul style="list-style-type: none"> - There are 2 type of parsing: <ul style="list-style-type: none"> • Html parsing (for hotels and tours): used to parse those data sources that rendered fully information in html without lazy loading. 								

- Ajax parsing (for flights): used to parse those data sources that rendered partial information via Ajax request.
- The parsing process is mentioned in section D.7.1 and D.7.2.

2.2.3.7. <Staff> Change password



USE CASE - SEPTIAL05

Use Case No.	UC 3.15	Use Case Version	1.0
Use Case Name	Change Password		
Author	KhuongND		
Date	15/02/2016	Priority	Normal

Actor:

- Staff and admin

Summary:

- This use case allows staff to change password of his/her account.

Goal:

- User can update his/her password.

Triggers:

- User sends request change password.

Preconditions:

- User logged in to the system.

Post Conditions:

- Success: Account data is updated password into storage.
- Fail: Show error message.

Main Success Scenario:

Step	Actor Action	System Response
1	Admin click link "Change password" in "Manage profile" page.	System navigates to "Change password"

		<p>page:</p> <ul style="list-style-type: none"> - Old Password: textbox, length: 1 - 10 character, required. - New Password: textbox, length: 1 - 10 character, required . - Confirm New Password: textbox, length: 1 - 10 character, required . - Change password: button.
2	Input field.	
3	Click on “Change password” button.	<p>Password of user will be updated to database. Show success message: “Account change password successfully”.</p> <p>[Exception 1,2,3,4,5]</p>

Alternative Scenario: N/A

Exceptions:

No	Cause	System Response
1	Click on “Change password” button.	New password and confirm password don’t match. Show error message: “New password and confirm password don’t match”.
2	Click on “Change password” button	Old Password is blank. Show error message: “Please fill old password”.
3	Click on “Change password” button	New Password is blank. Show error message: “Please fill new password”.
4	Click on “Change password” button	Confirm New Password is blank. Show error message: “Please fill confirm new password”.
5	Click on “Change password” button	Old Password isn’t match with old password of account in database. Show error message: “Please fill old password, this password is wrong”.

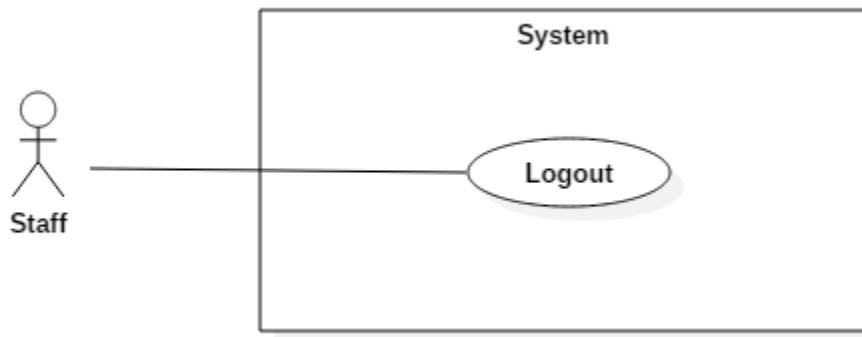
Relationships: N/A

Business Rules:

- If account is logged on a different session and change personal info success, account will be logout.
- If user fill wrong old password 3 times, account will be blocked in 30 minutes.

- New password will be prompted security level (strength,weak).

2.2.3.8. <Staff> Logout



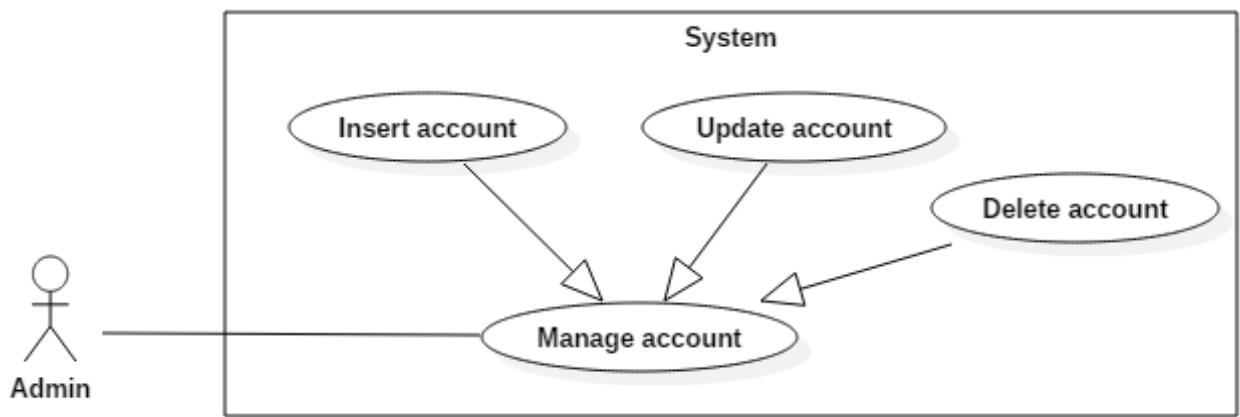
USE CASE - SEPTIAL04					
Use Case No.	UC 3.16	Use Case Version	1.0		
Use Case Name	Logout				
Author	KhuongND				
Date	15/02/2016	Priority	Normal		
Actor:					
- Staff and admin					
Summary:					
- This use case allows staff to log out of system.					
Goal:					
- User can log out of system.					
Triggers:					
- User sends request logout.					
Preconditions:					
- User logged in to the system.					
Post Conditions:					
- Success: Display login page. - Fail: N/A.					
Main Success Scenario:					
Step	Actor Action	System Response			
1	User click button "Logout" in menu bar.	System takes user out of system and display login page.			
Alternative Scenario: N/A					
Exceptions: N/A					

Relationships: N/A

Business Rules:

- If user logout while send request to system, all request will be delete.
- If user is not available longer than 30 minutes, they will see the login view when they be back.

2.2.4. <Admin> Overview Use Case



2.2.4.1. <Admin> Create account

USE CASE - SEPTIAL01

Use Case No.	UC 3.17	Use Case Version	1.0
Use Case Name	Create Account		
Author	CuongHH		
Date	15/02/2016	Priority	Normal

Actor:

- Admin

Summary:

- Create a new account

Goal:

- A account is created with role Staff

Triggers:

- Admin sends request to create account.

Preconditions:

- User logged in the system with role "Admin".

Post Conditions:

- **Success:** New account will be created for staff.
- **Fail:** Show error message.

Main Success Scenario:

Step	Actor Action	System Response
1	Admin click button "Create Account" in menu bar.	<p>System navigates to "Insert Account" page.</p> <ul style="list-style-type: none"> - Email: textbox, length: 1 - 50 character, required. - Password: textbox, length: 1 - 10 character, required. - Confirm Password: textbox, length: 1 - 10 character, required. - Full name: textbox, length: 1 - 50 character, required. - Gender: dropdownlist (male, female), required. - Birthdate: datepicker, required, value from 1900 to current year. - Register: button.
2	Input field.	
3	Click on "Insert" button.	<p>New account will be added to database. Show success message: "A account staff created successfully".</p> <p>[Exception 1,2,3,4]</p>

Alternative Scenario: N/A

Exceptions:

No	Cause	System Response
1	Password and confirming password don't match.	Show error message: "Password and confirm password don't match".
2	Password is blank.	Show error message: "Please fill password".
3	Email is blank or wrong format.	Show error message: "Please fill email".
4	Duplicated email.	Show error message: "Please choose another email. Email already in use".

Relationships: N/A

Business Rules:

- New Account will be created with status in-active.
- System must ensure has not duplicated email.

2.2.4.2. <Admin> Update account

USE CASE - SEPTIAL02			
Use Case No.	UC 3.18	Use Case Version	1.0
Use Case Name	Update Account		
Author	CuongHH		
Date	15/02/2016	Priority	Normal

Actor:

- Admin

Summary:

- Update account information

Goal:

- Update account information

Triggers:

- Admin sends request to update account.

Preconditions:

- User logged in to the system as Admin role.

Post Conditions:

- **Success:** Account data is updated into storage.
- **Fail:** Show error message.

Main Success Scenario:

Step	Actor Action	System Response
1	Admin click link "Update" in list account.	<p>System navigates to "Update account" page and fill information of this account to form:</p> <ul style="list-style-type: none"> - Email: textbox, length: 1 - 50 character, read-only. - Full name: textbox, length: 1 - 50 character. - New Password: textbox, length: 1 - 10 character . - Confirm New Password: textbox, length: 1 - 10 character . - Gender: dropdownlist (male, female). - Birthdate: datepicker, value from 1900 to current year. - Status: dropdownlist (in-active,

		blocked). - Update: button.
2	Input field.	
3	Click on "Update" button.	Account will be updated to database. Show success message: "Account updated successfully". [Exception 1]

Alternative Scenario: N/A

Exceptions:

No	Cause	System Response
1	New password and confirm password don't match.	Show error message: "New password and confirm password don't match".

Relationships: N/A

Business Rules:

- If account's status is changed to "blocked", the account won't be able to access the system.

2.2.4.3. <Admin> Delete account

USE CASE – SEPTIAL03			
Use Case No.	UC 3.19	Use Case Version	1.0
Use Case Name	Delete Account		
Author	CuongHH		
Date	15/02/2016	Priority	Normal
Actor:			
- Admin			
Summary:			
- Delete an account in system.			
Goal:			
- Remove a staff account.			
Triggers:			
- Admin sends request delete account.			

Preconditions:

- User has to log in to the system as Admin role.

Post Conditions:

- **Success:** The account is deleted.
- **Fail:** Show error message.

Main Success Scenario:

Step	Actor Action	System Response
1	Admin click link "Delete" in list account.	- System shows message options to confirm delete account: "Do you want to delete this account ?".
2	Admin confirms "Yes" to delete account. [Alternative 1]	Account will be deleted to database. Show success message: "Account deleted successfully".

Alternative Scenario:

Step	Actor Action	System Response
1	Admin confirm "Cancel" to cancel account	System abort to delete account.

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

- Admin cannot delete himself/herself.

3. Software System Attribute

3.1. Usability

- The system usability is easy to use that users generally don't need to look at the document in order to use them.

3.2. Reliability

- The data provided by the system needs to be updated within 30 minutes – 2 hours.
- The data should be backed up everyday.

3.3. Availability

- The web app must be available 24/7.
- The mobile application needs to connect to the internet.

3.4. Security

- Users can use the software without having to register or provide their private information.
- The mobile application does not install anything else beside itself to user's phone.

3.5. Maintainability

- The software is organized into modules.

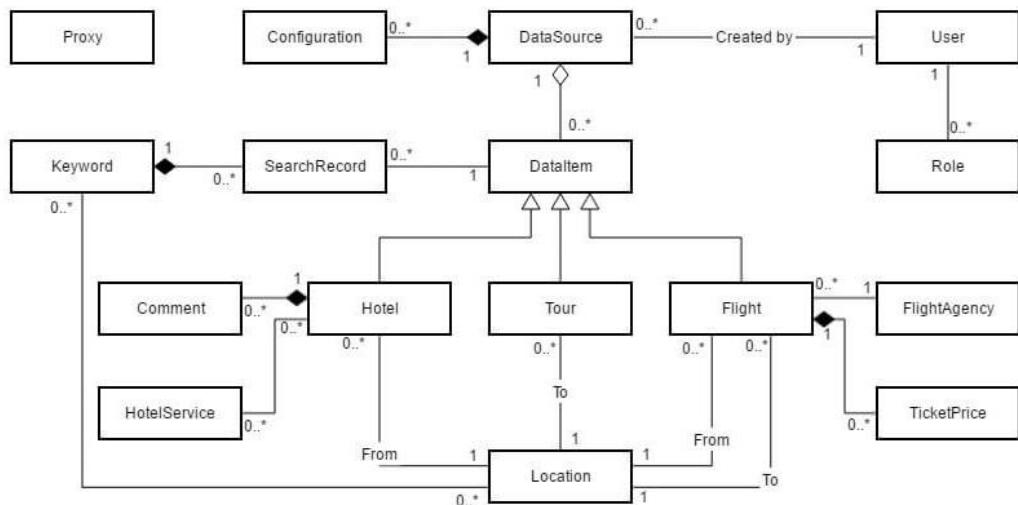
3.6. Portability

- The software itself is a web application, therefore it can be used in any platform that has a web browser and can connect to the internet.
- Users can use mobile app on every Android smart phone that supports Android API 4.0 or above.

3.7. Performance

- Requests from users are responded in less than 10 seconds at 5 Mbps bandwidth speed.

4. Conceptual Diagram



Data Dictionary:

Entity Name	Description
DataSource	Contains all the information about websites to be parsed.
Configuration	Specifies needed properties for parsing data.
DataItem	Has 3 types: Hotel, Tour, Flight
Hotel	Contains all the properties about hotels.
Flight	Contains all the properties about tours.
Tour	Contains all the properties about flights.
Keyword	Recorded from search keywords of users.
SearchRecord	Used to update click count & view count from user behaviors.
Location	Used to categorize data by cities/provinces.
Comment	Parsed comments associated with hotels.
HotelService	Parsed services associated with hotels.
FlightAgency	Parsed agencies associated with hotels.
TicketPrice	Categorize flight data into 3 type of tickets: Adult, Children, Orphan
Proxy	Contains the list of proxies used to parse data without being banned by the data sources.
User	Store information of admin and staff of the system.
Role	Has 2 types: Admin, Staff

Table 11: Conceptual Diagram Data Dictionary

D. Report No. 4 Software Design Description

1. Design Overview

This document describes the technical and user interface design of SEPTIAL. It includes the architectural design, the detailed design of common functions and business functions and the design of database model.

The architectural design describes the overall architecture of the system and the architecture of each main component and subsystem.

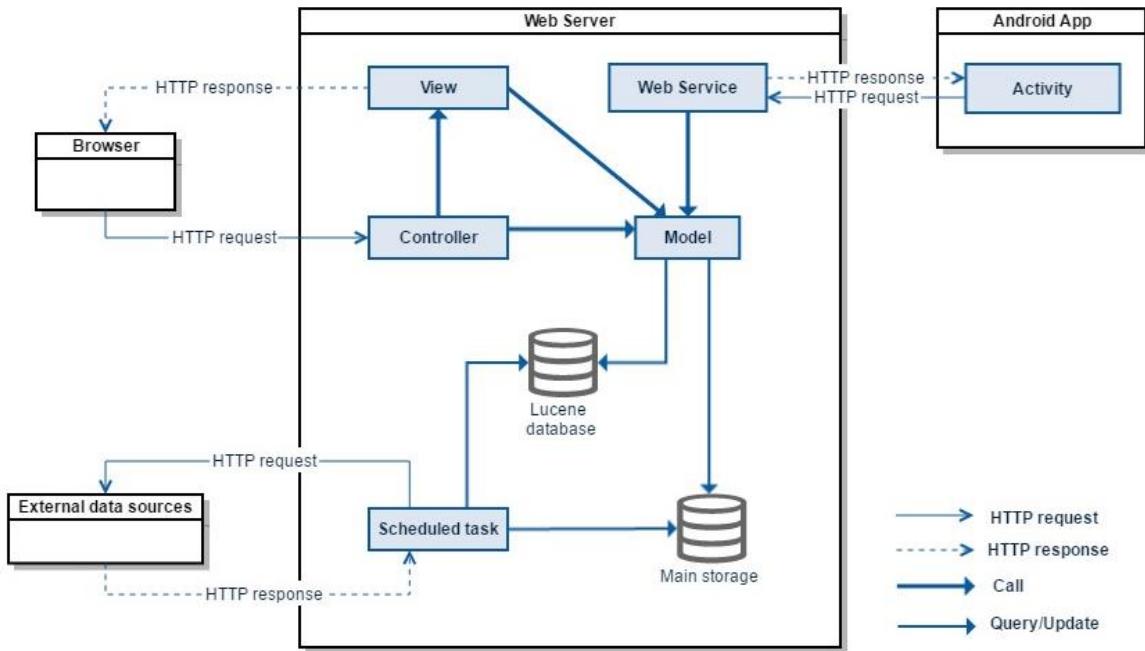
The detailed design describes static and dynamic structure for each component and functions. It includes class diagrams, class explanations and sequence diagrams for each use cases.

The database design describes the relationships between entities and details of each entity.

Document overview:

- Section 2: Overall description of the system architecture design.
- Section 3: Component diagrams that describe the connection and integration of the system.
- Section 4: Detail design description which includes class diagram, class explanation, and sequence diagram to details the application functions.
- Section 5: Interface design for web, android applications.
- Section 6: ERD with logical diagram.

2. System Architecture Design



1.1. Web Application architecture description

The web app is developed under ASP .NET MVC architecture with the following components:

- Model: Model objects are the parts of the application that implement the logic for the application's data domain.
- View: Views are the components that display the application's user interface.
- Controller: Controllers are the components that handle user interaction, work with the model, and ultimately select a view to render that displays UI.

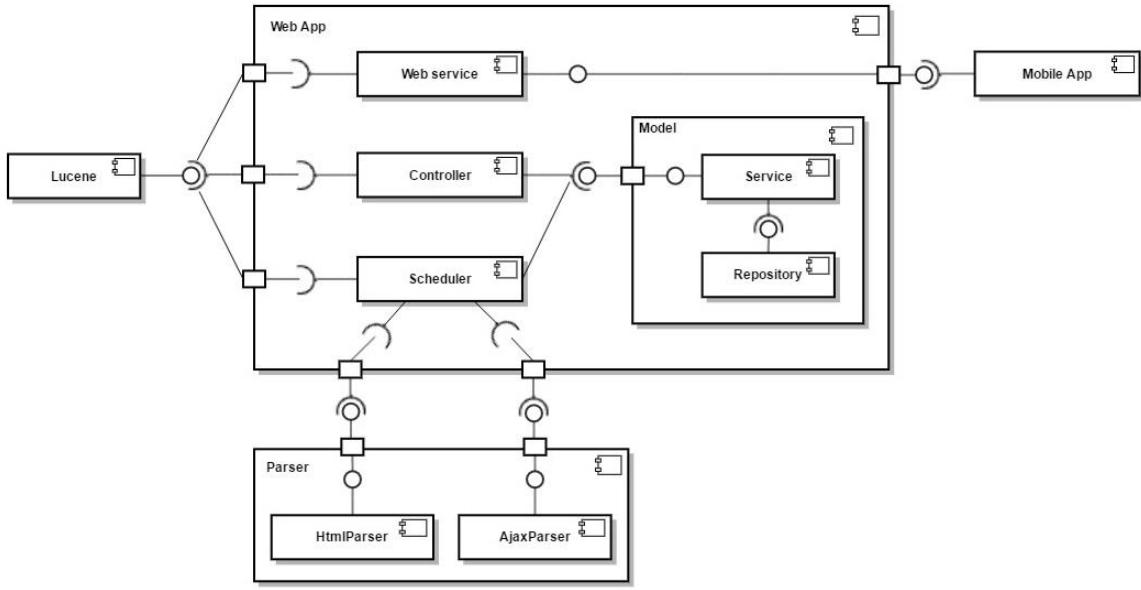
Besides, the system includes:

- Lucene server: Lucene server has a No-SQL database that contains all the data used for retrieving search results.
- Scheduler: Scheduled tasks parse data from external data sources, then categorize and update data from main storage to Lucene database.

1.2. Mobile Application architecture description

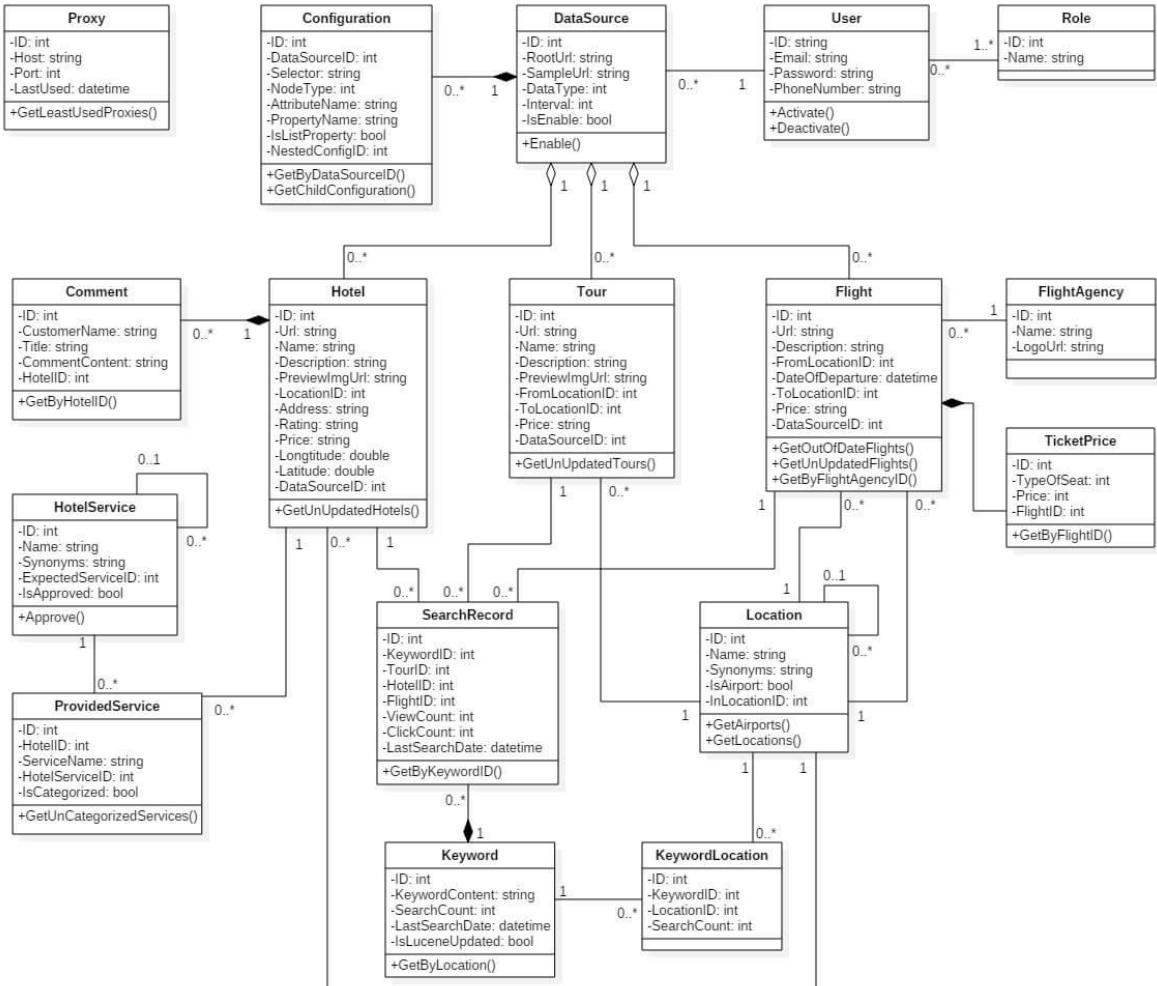
The application is developed as an Android native application. In general, the application architecture conforms to Android architecture: Activity is the basic core of an android application that handles user input, create thread to run AsyncTask, send request and receive data from server via web sevices.

3. Component Diagram



4. Detail Description

4.1. Class Diagram



4.2. Class Diagram Explanation

Entity Name	Description
DataSource	Contains all the information about websites to be parsed.
Configuration	Specifies needed properties for parsing data.
Hotel	Contains all the properties about hotels.
Flight	Contains all the properties about tours.
Tour	Contains all the properties about flights.
Keyword	Recorded from search keywords of users.
SearchRecord	Used to update click count & view count from user behaviors.
Location	Used to categorize data by cities/provinces.
Comment	Parsed comments associated with hotels.
HotelService	Official hotel services used for filtering search results.
ProvidedService	Parsed services associated with hotels.
FlightAgency	Parsed agencies associated with hotels.
TicketPrice	Categorize flight data into 3 type of tickets: Adult, Children, Orphan
Proxy	Contains the list of proxies used to parse data without being banned by the data sources.
User	Store information of admin and staff of the system.
Role	Has 2 types: Admin, Staff

4.2.1. *DataSource*

Attribute	Description
ID	Unique ID for each data source
RootUrl	The root url of the data source used for limit the crawled urls when parsing data

SampleUrl	The url to the web page with the standard html structure
DataType	Can be 3 types: <ul style="list-style-type: none">• Hotel = 0• Tour = 1• Flight = 2
Interval	The pending time (in minute) after the scheduler complete parsing data from the data source. When the pending time is up, the scheduler start to run again.
.IsEnabled	Can be True or False. If .IsEnabled = True, the scheduler automatically run to parse data after the interval of pending.

4.2.2. Configuration

Attribute	Description
ID	Unique ID for each configuration
DataSourceID	ID of the data source that contains the configuration
PropertyName	The property name of hotel, tour, or flight that will be parsed
Selector	The CSS selector of the element containing data for the specified property
NodeType	Can be 3 type: <ul style="list-style-type: none">• Text = 0• Attribute = 1• Html = 2
AttributeName	<ul style="list-style-type: none">• If the NodeType = 1, AttributeName will contain the name of the attribute that contain data for the specified property.• Else, AttributeName = null.
IsListProperty	Can be True or False. <ul style="list-style-type: none">• True: The specified property is a list of objects (Ex: list of comments on a hotel)• False: The specified property is a single property (Ex:

	Hotel name, description)
NestedConfigID	If the specified property is the property of the object type in a list-property, the NestedConfigID is the ID of the Configuration of the list-property. Ex: Hotel has the list-property "Comments", Comment has the property "CommentContent". The ID of the configuration corresponding to property "Comments" is the NestedConfigID of the configuration corresponding to the property "CommentContent".

4.2.3. Keyword

Attribute	Description
ID	Unique ID for each keyword
KeywordContent	The whole input keyword.
SearchCount	By default, SearchCount = 1. Whenever users enter a keyword, increase SearchCount by 1.
LastSearchDate	The last time this keyword is searched.
IsLuceneUpdated	Can be True or False. If the KeywordContent and SearchCount is updated to Lucene database, IsLuceneUpdated = true.

4.2.4. SearchRecord

Attribute	Description
ClickCount	By default, ClickCount = 1. Whenever users select a hotel/tour/flight to view detail, increase ClickCount by 1.
ViewCount	By default, ViewCount = 1. Whenever a search result is display to users, increase ViewCount by 1.
LastSearchDate	The last time users clicks on a search result for the input keyword.

4.2.5. HotelService

Attribute	Description
-----------	-------------

ID	Unique ID for each hotel service
Name	Name of the group of similar services
Synonyms	All the similar service names in the group
ExpectedServiceID	ID of the service that is supposed to be in the same group with this service.
IsApproved	Can be True or False. If Admin or Staff approve this service, IsApproved = true.

4.2.6. ProvidedService

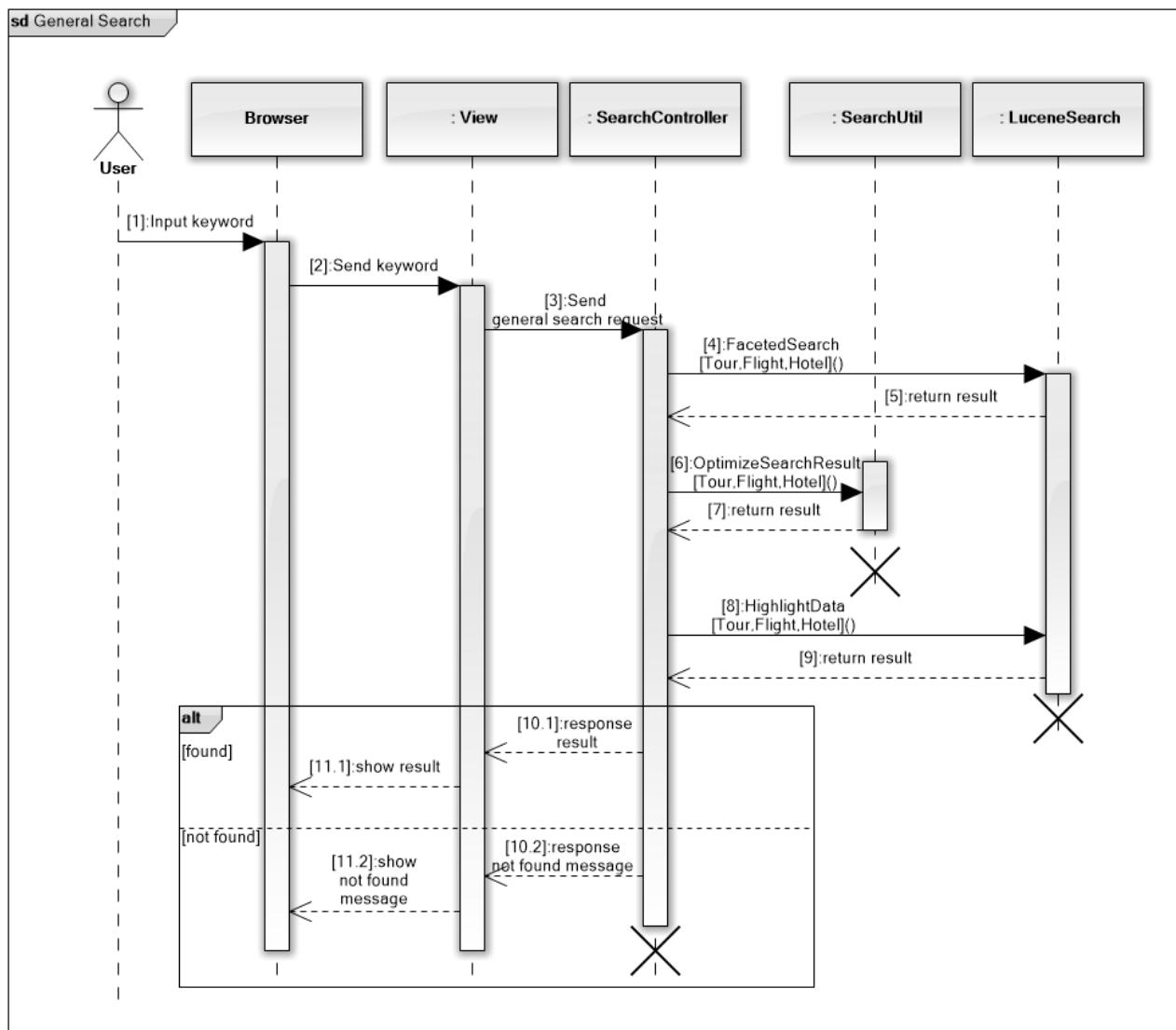
Attribute	Description
ID	Unique ID for each provided service.
ServiceName	The parsed service name from the data source
HotelServiceID	ID of the service group that is supposed to contain this service. If system cannot find any relevant hotel service, HotelServiceID = null.
IsCategorized	Can be True or False. Used to mark if this service is already categorized to a service group or not.

4.3. Interaction Diagram

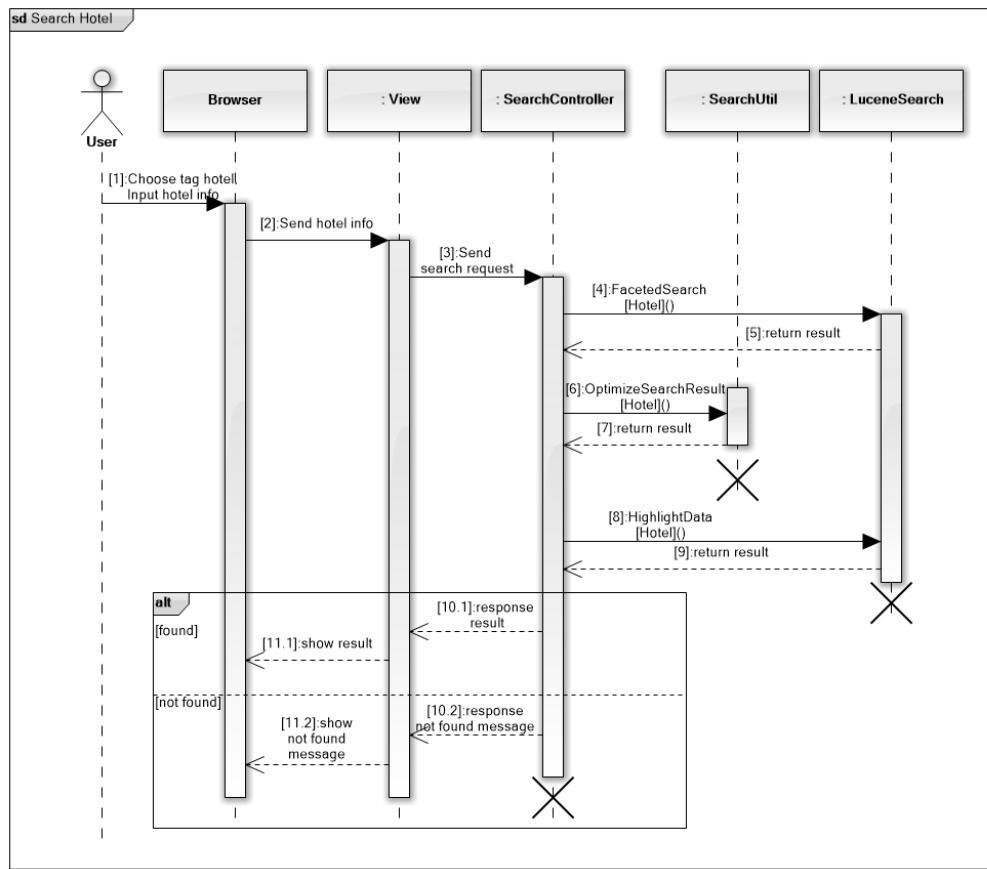
4.3.1. Web App

4.3.1.1. User

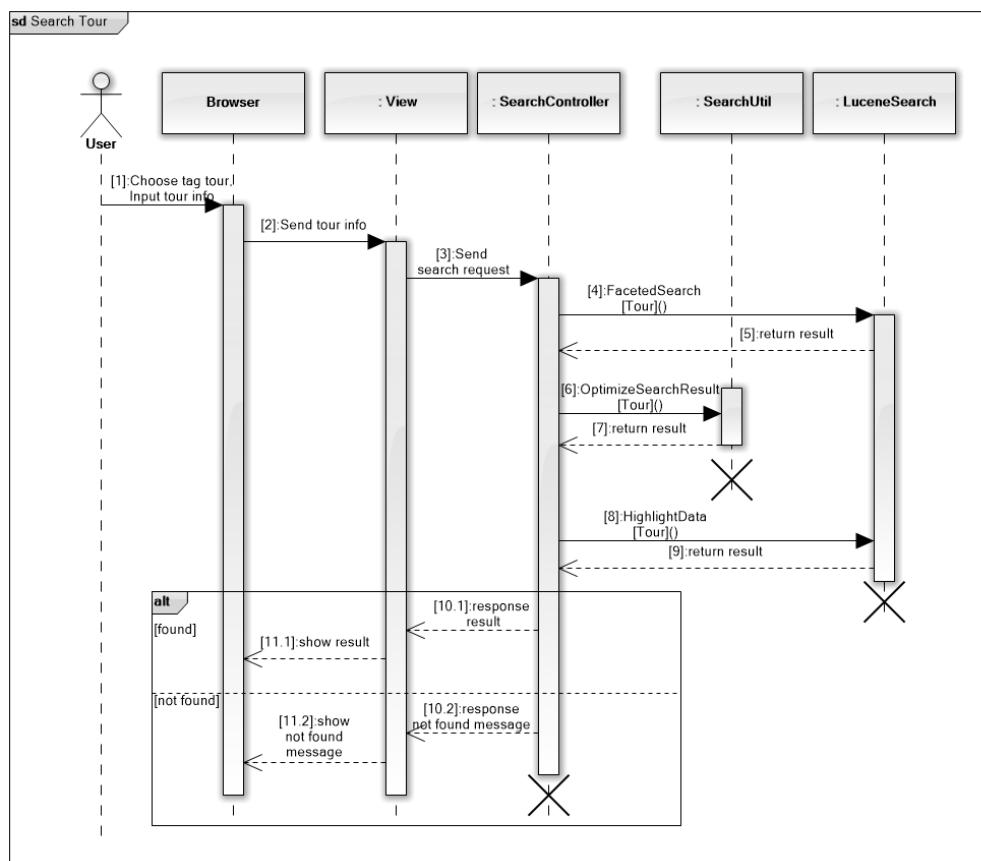
4.3.1.1.1. General search



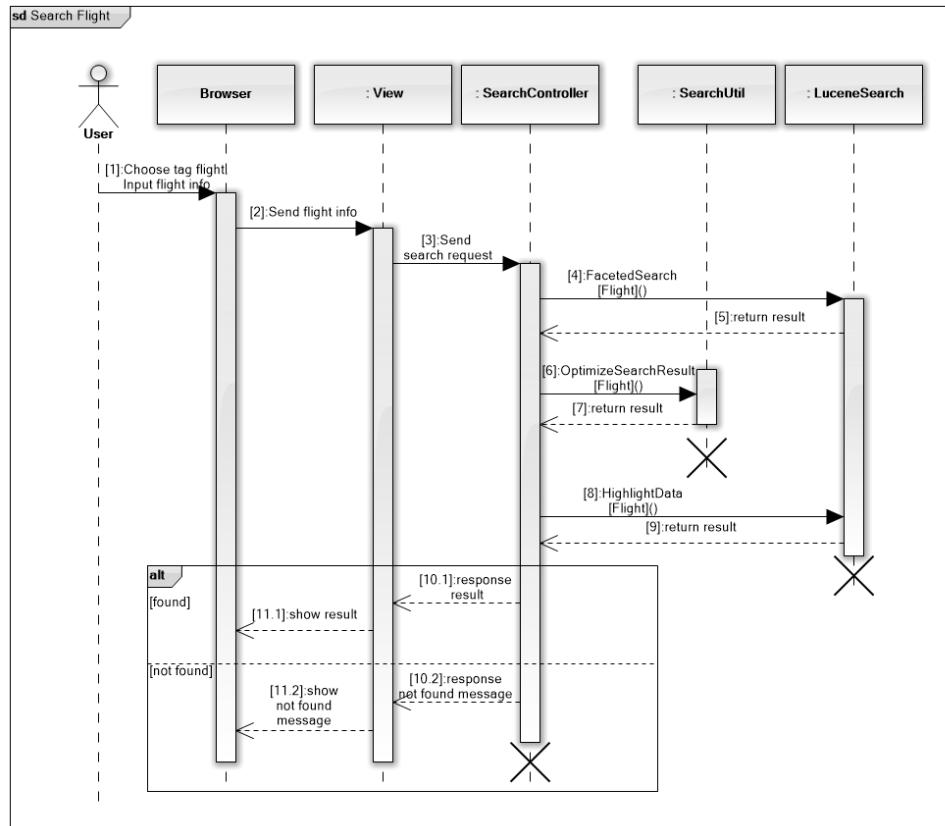
4.3.1.1.2. Search hotel



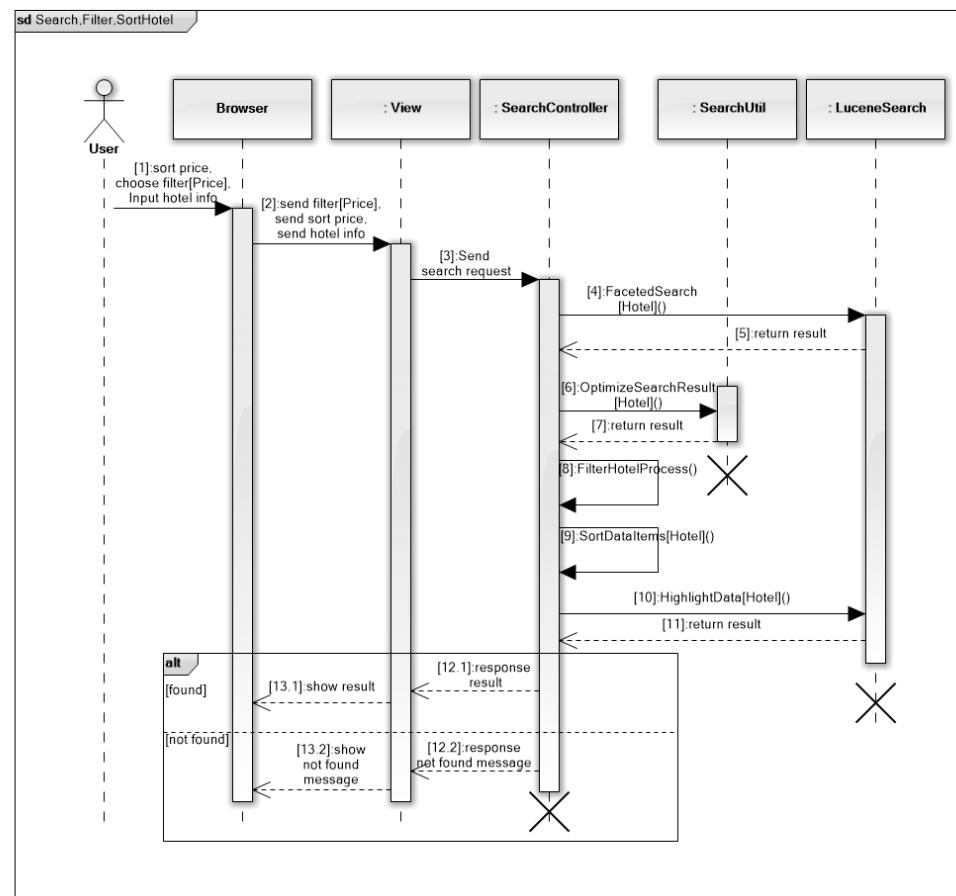
4.3.1.1.3. Search tour



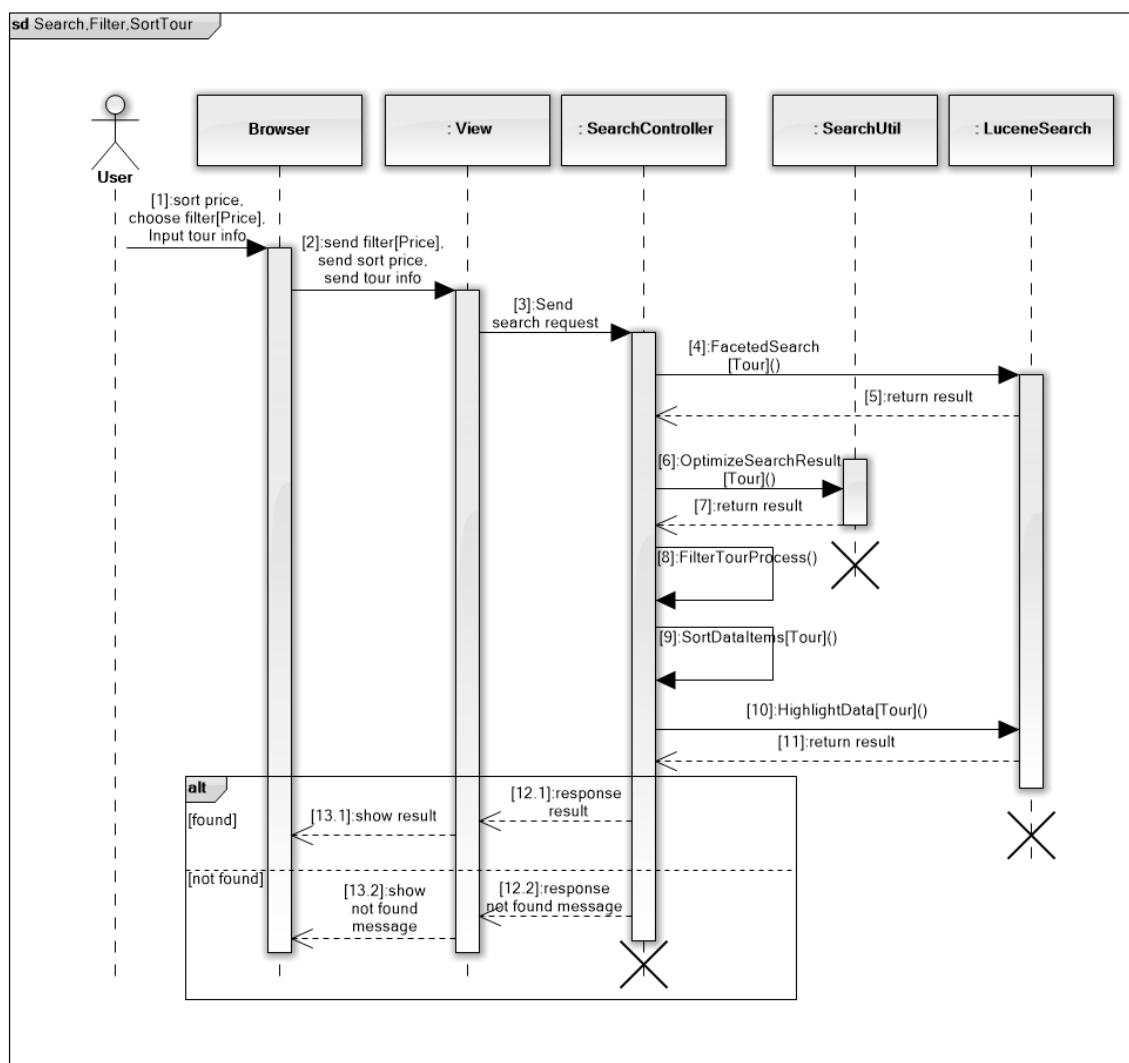
4.3.1.1.4. Search flight



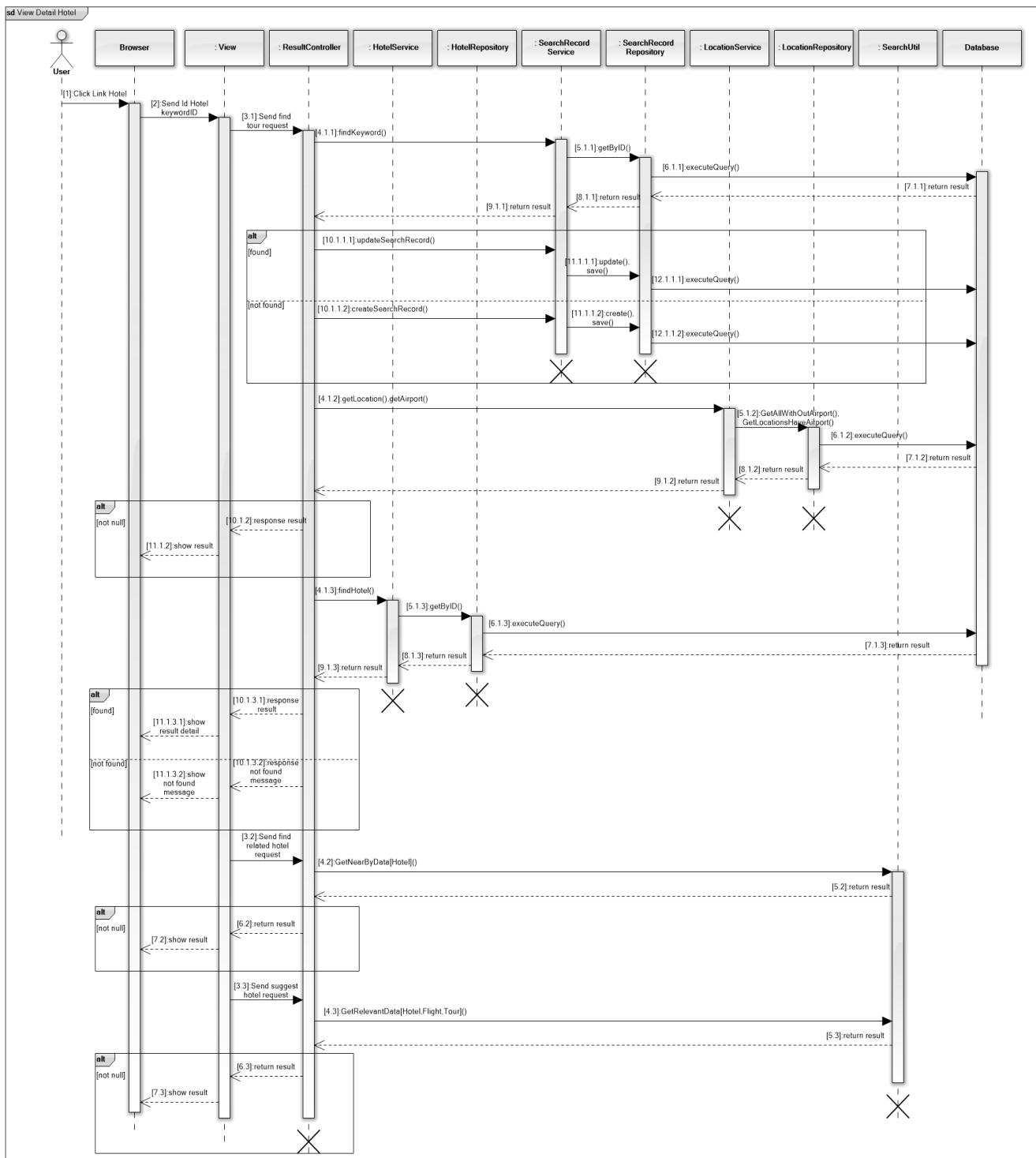
4.3.1.1.5. Filter & sort hotel search result



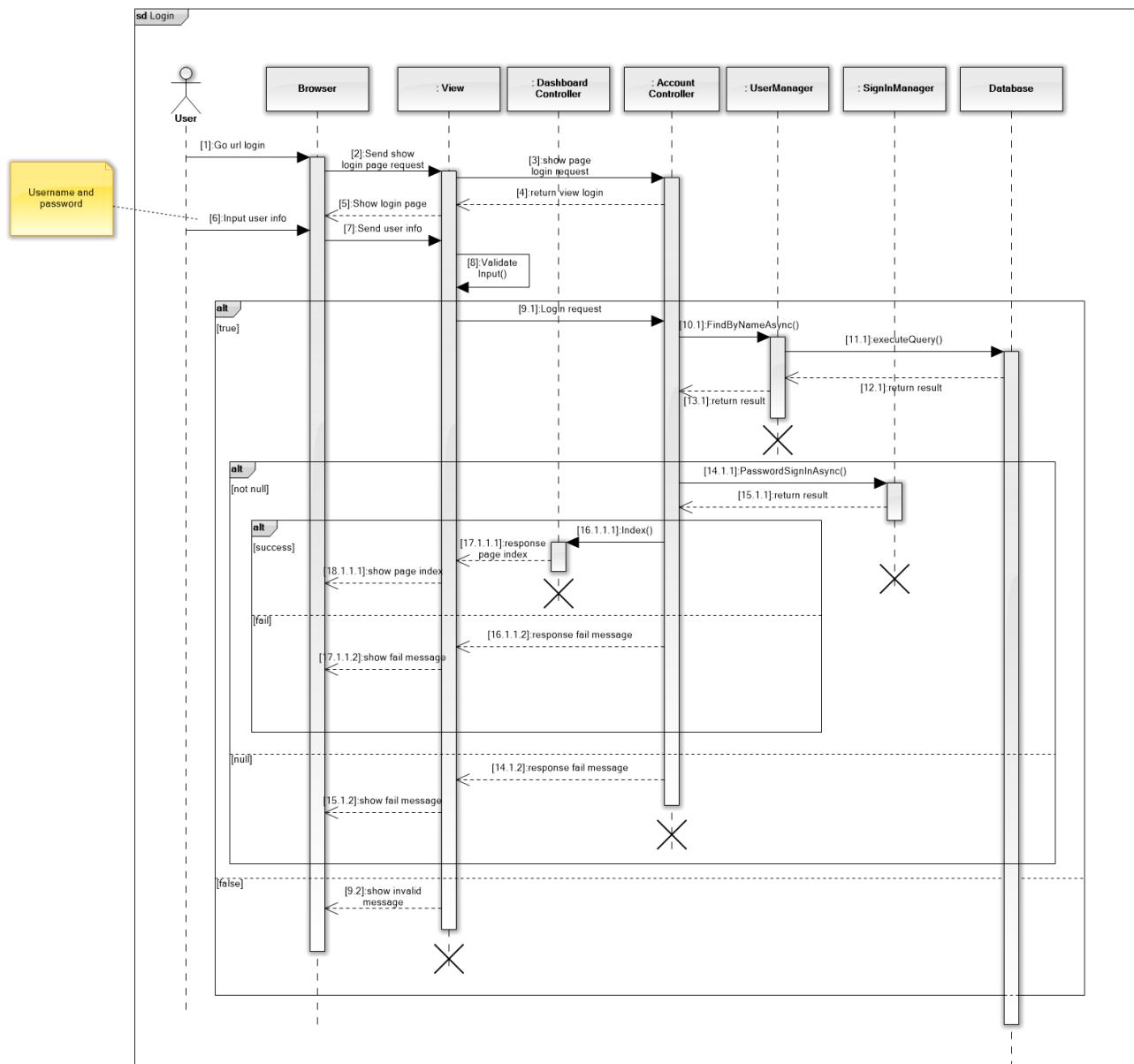
4.3.1.1.6. Filter & sort tour search result



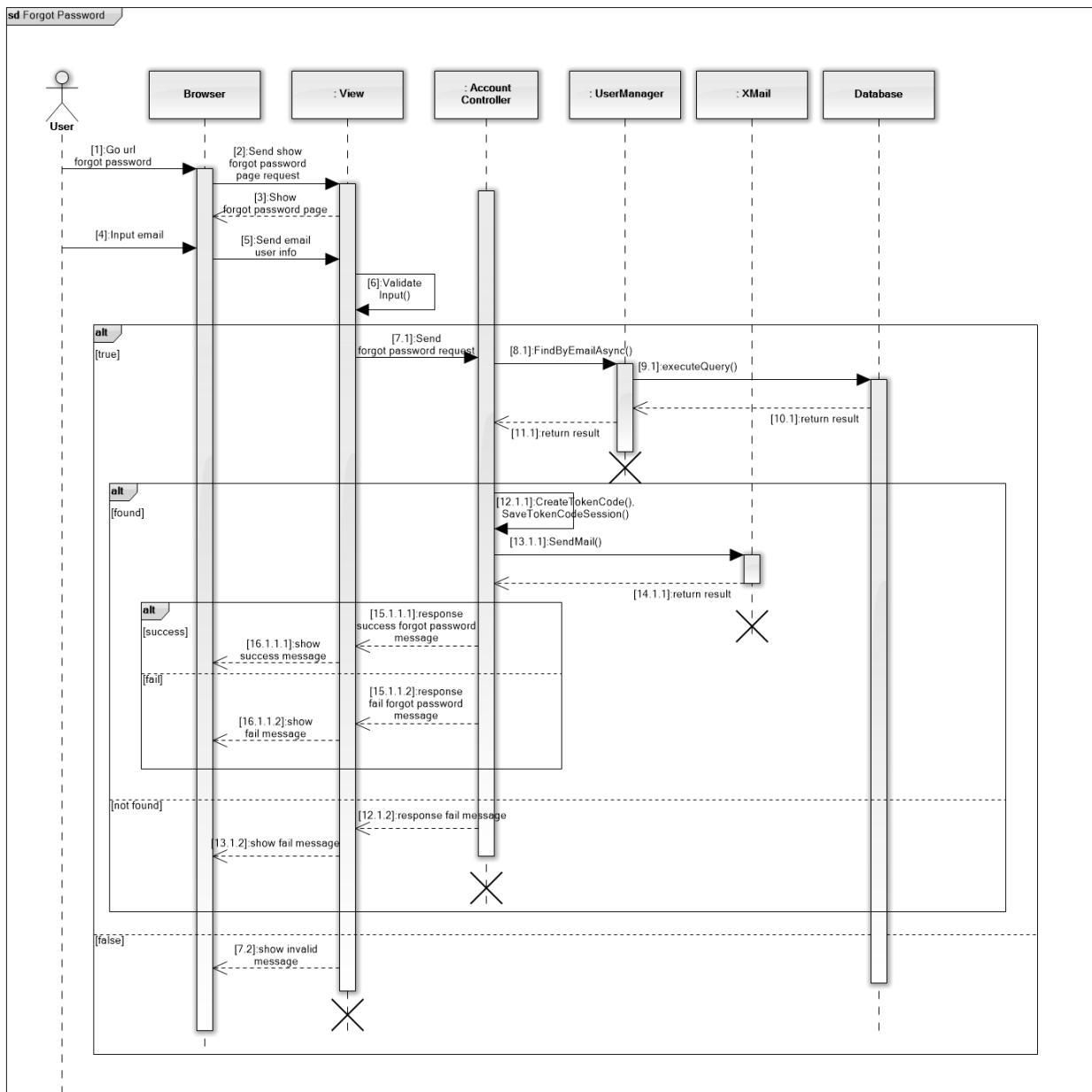
4.3.1.1.7. View hotel detail



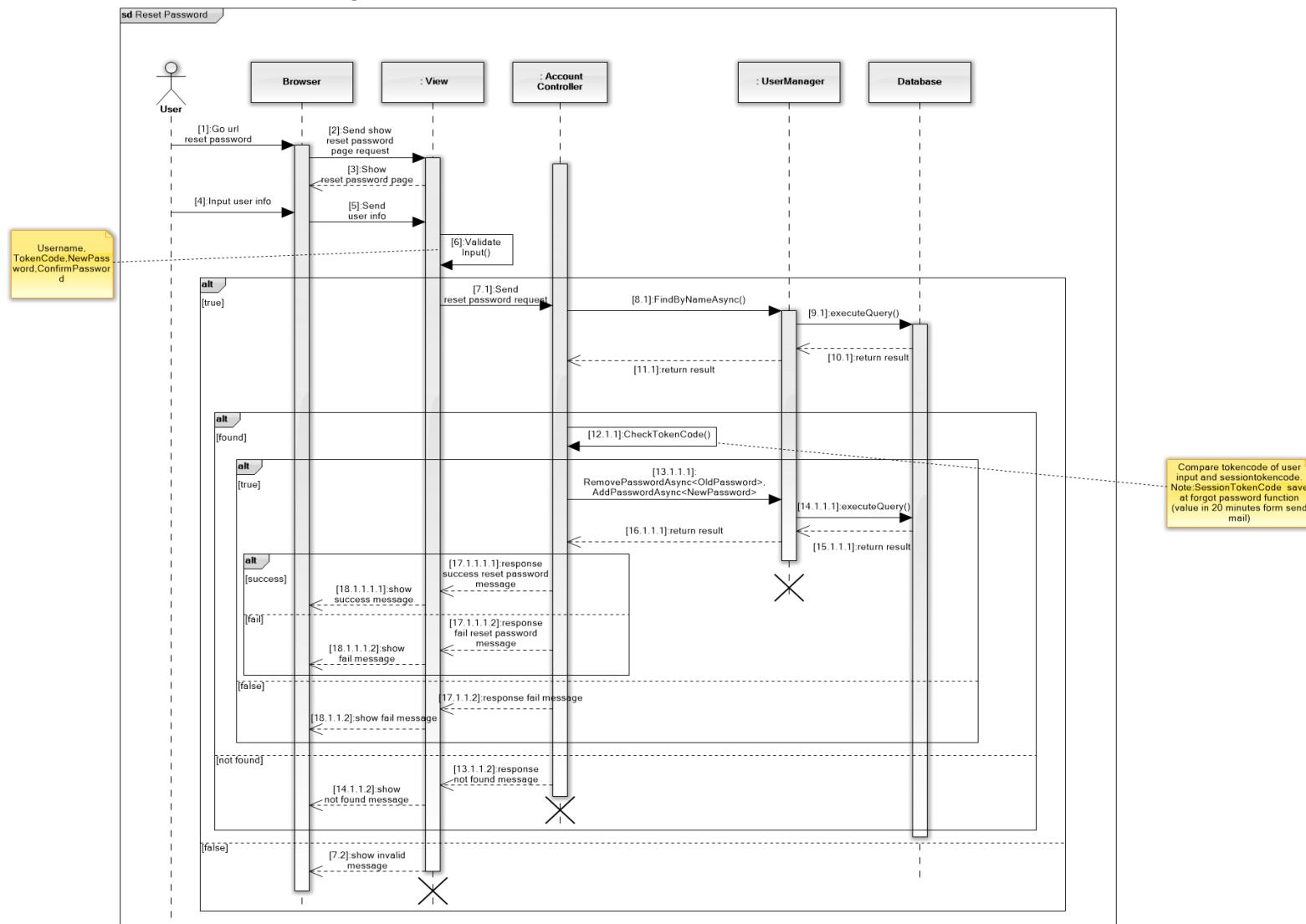
4.3.1.1.8. Login



4.3.1.1.9. Forget password

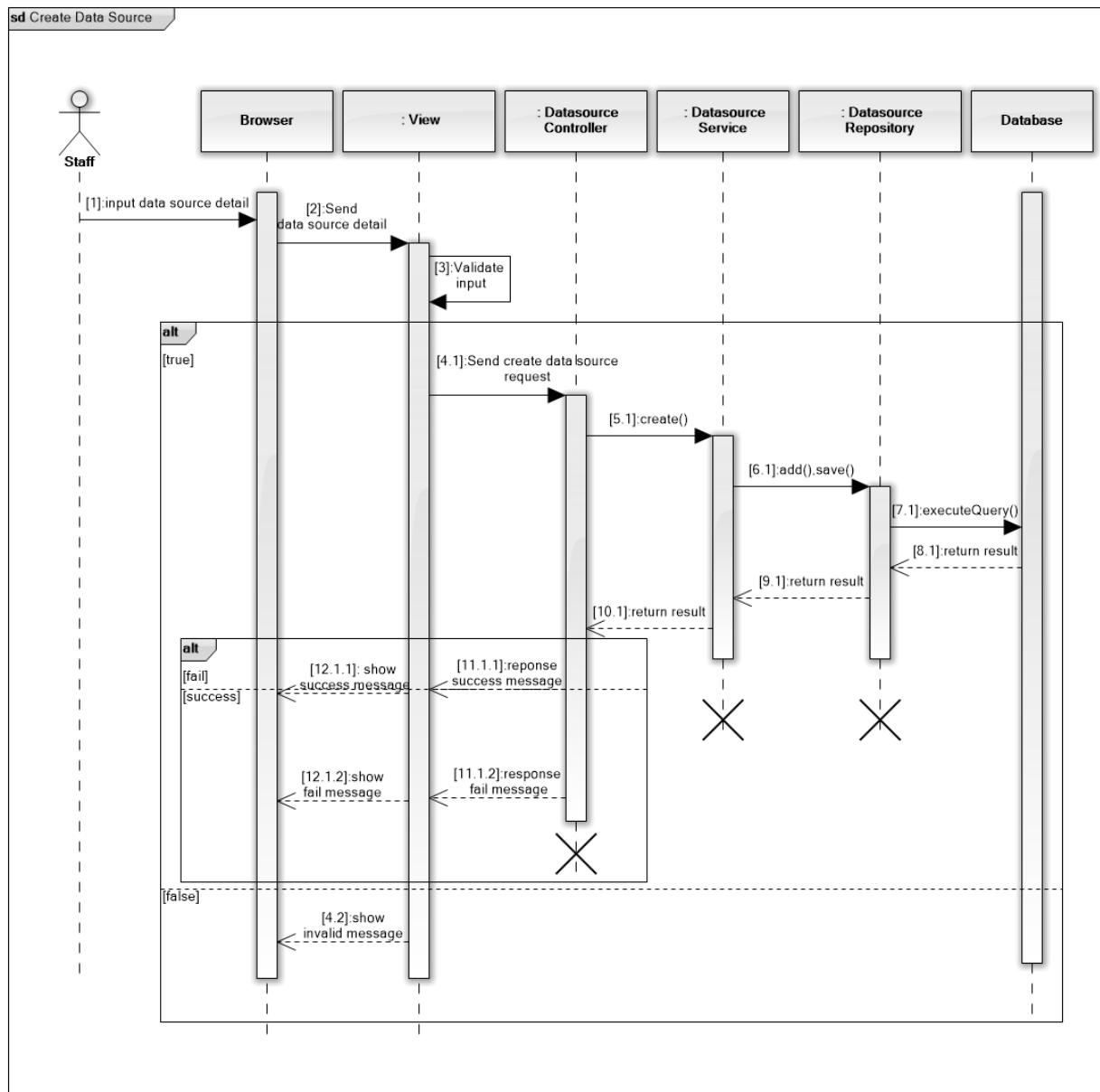


4.3.1.10. Reset password

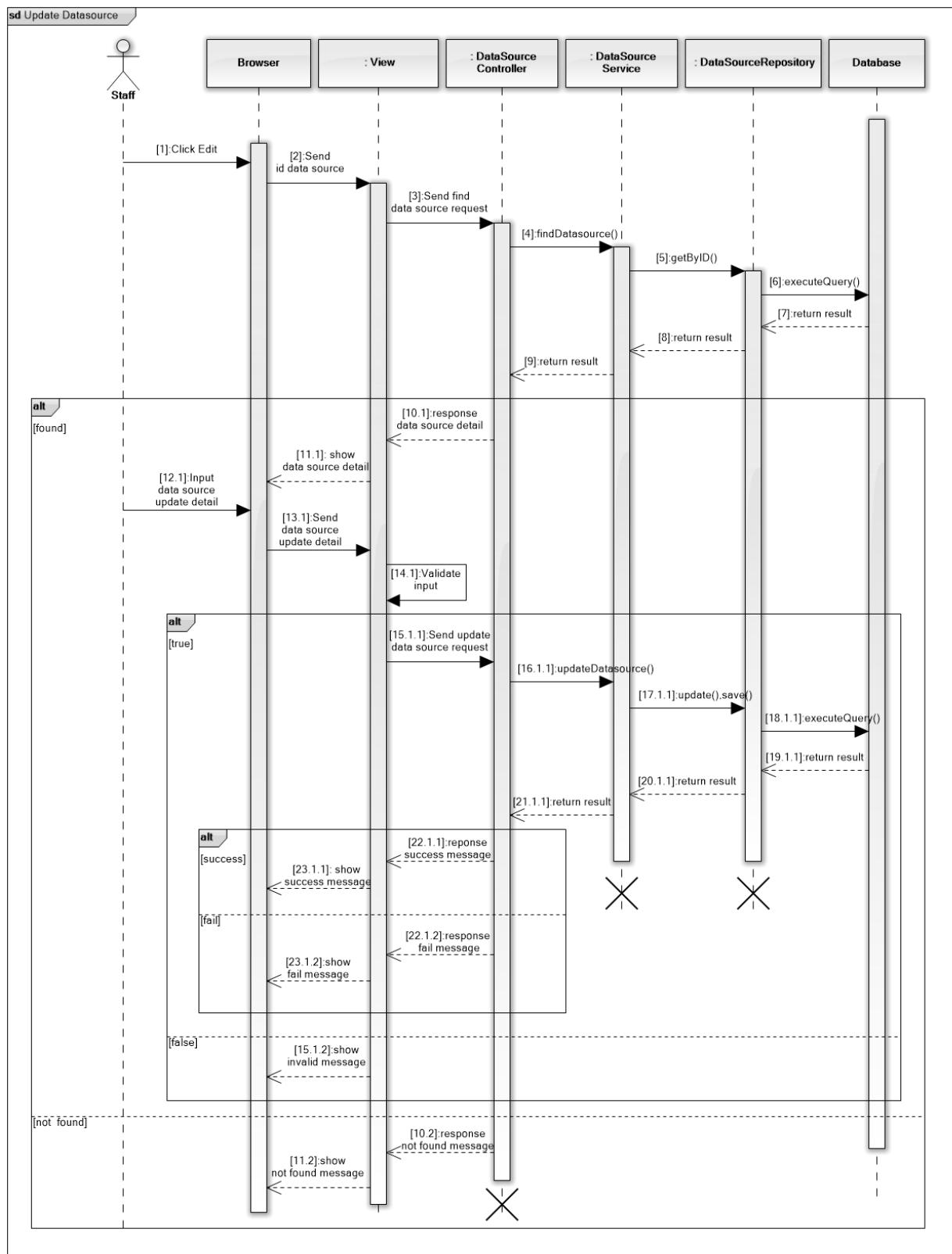


4.3.1.2. Admin/Staff

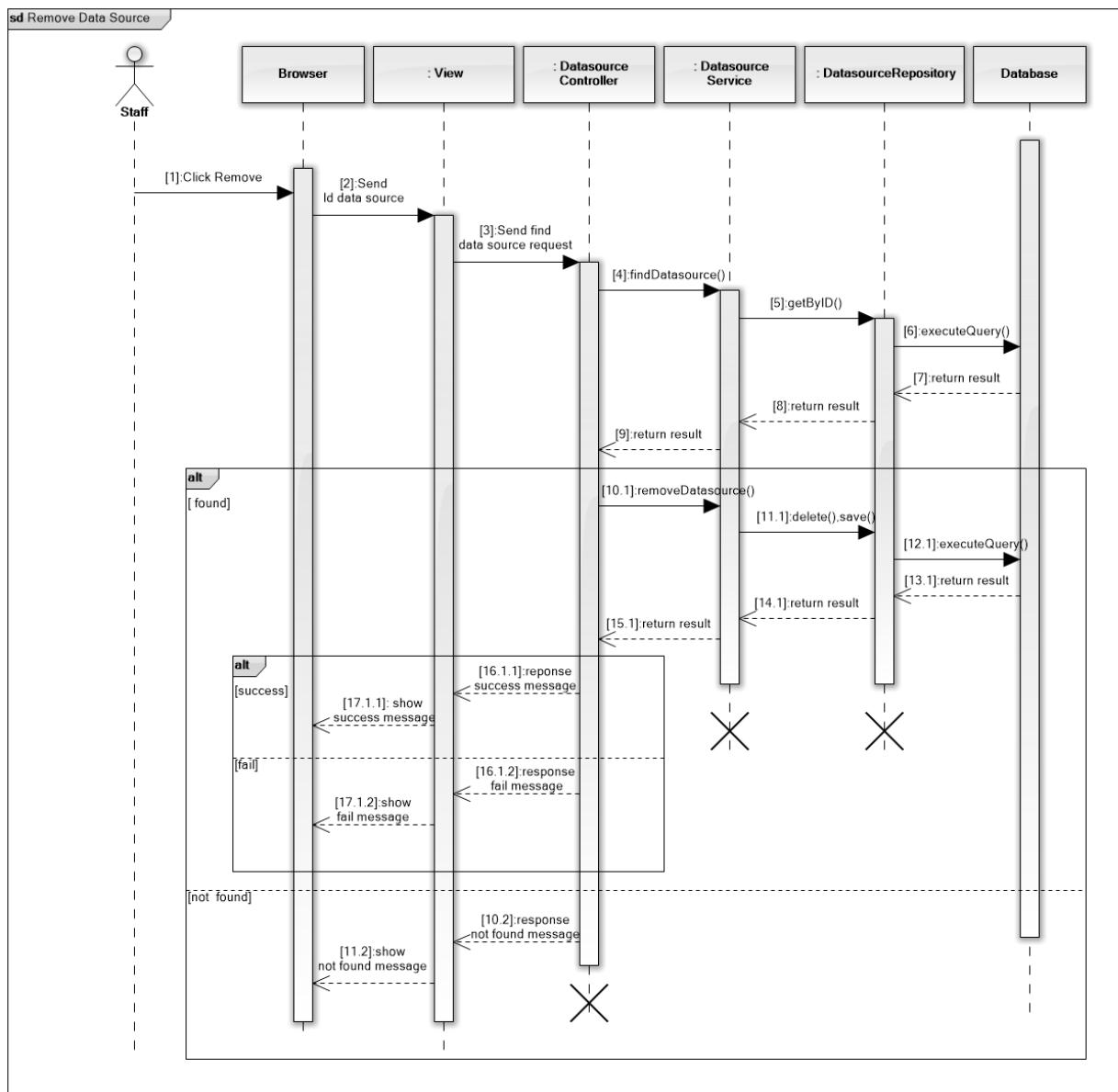
4.3.1.2.1. Create data source



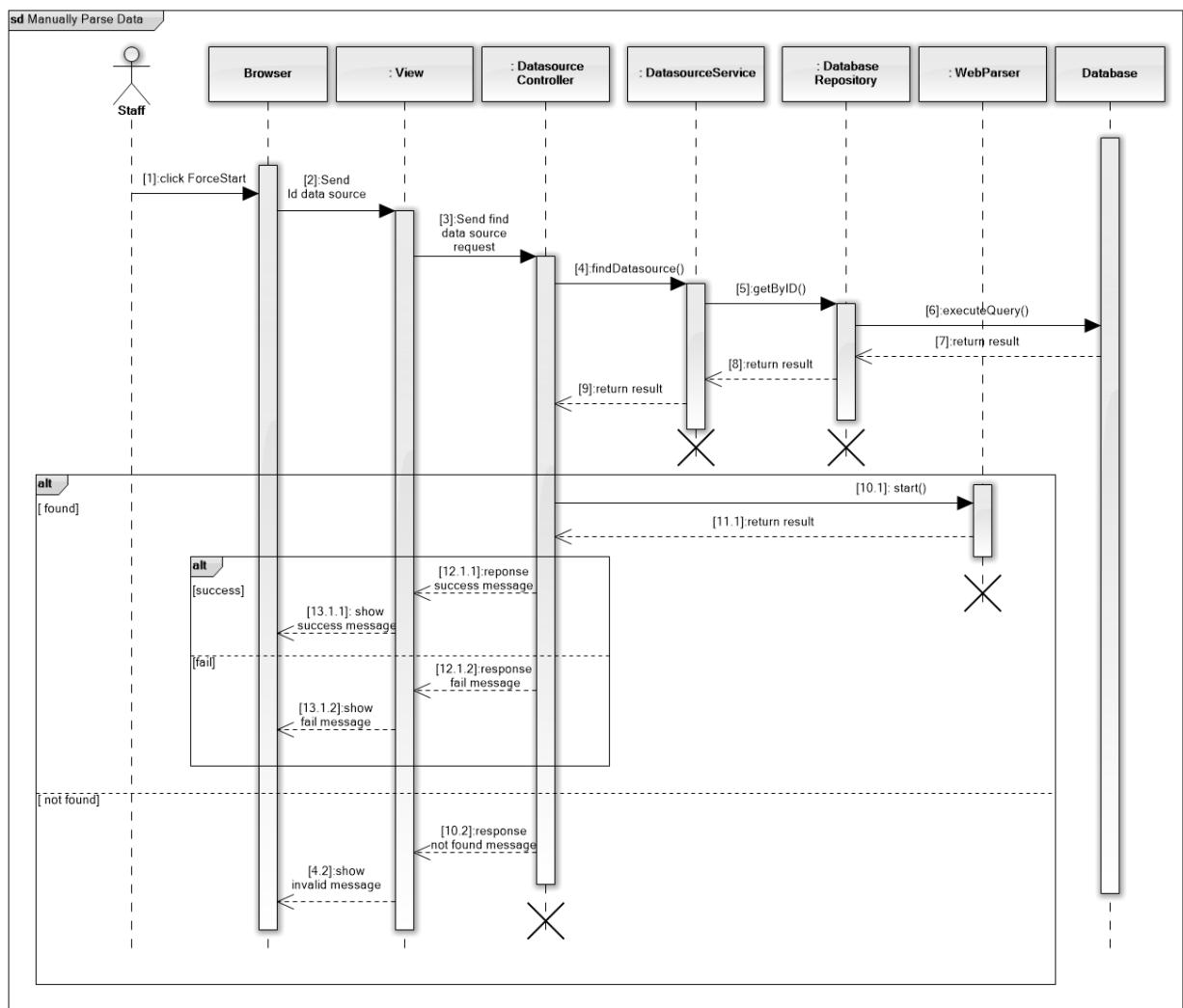
4.3.1.2.2. Edit data source



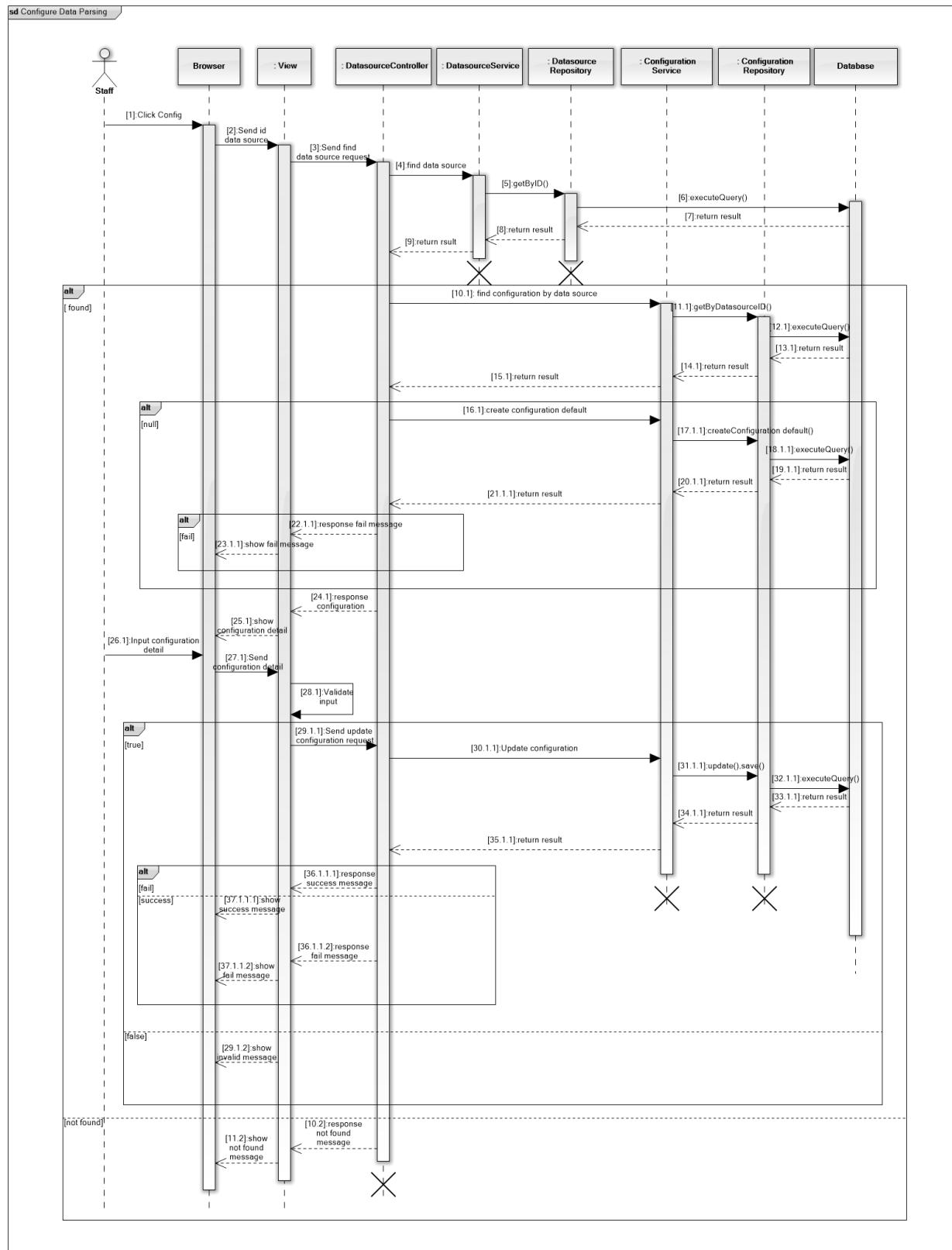
4.3.1.2.3. Remove data source



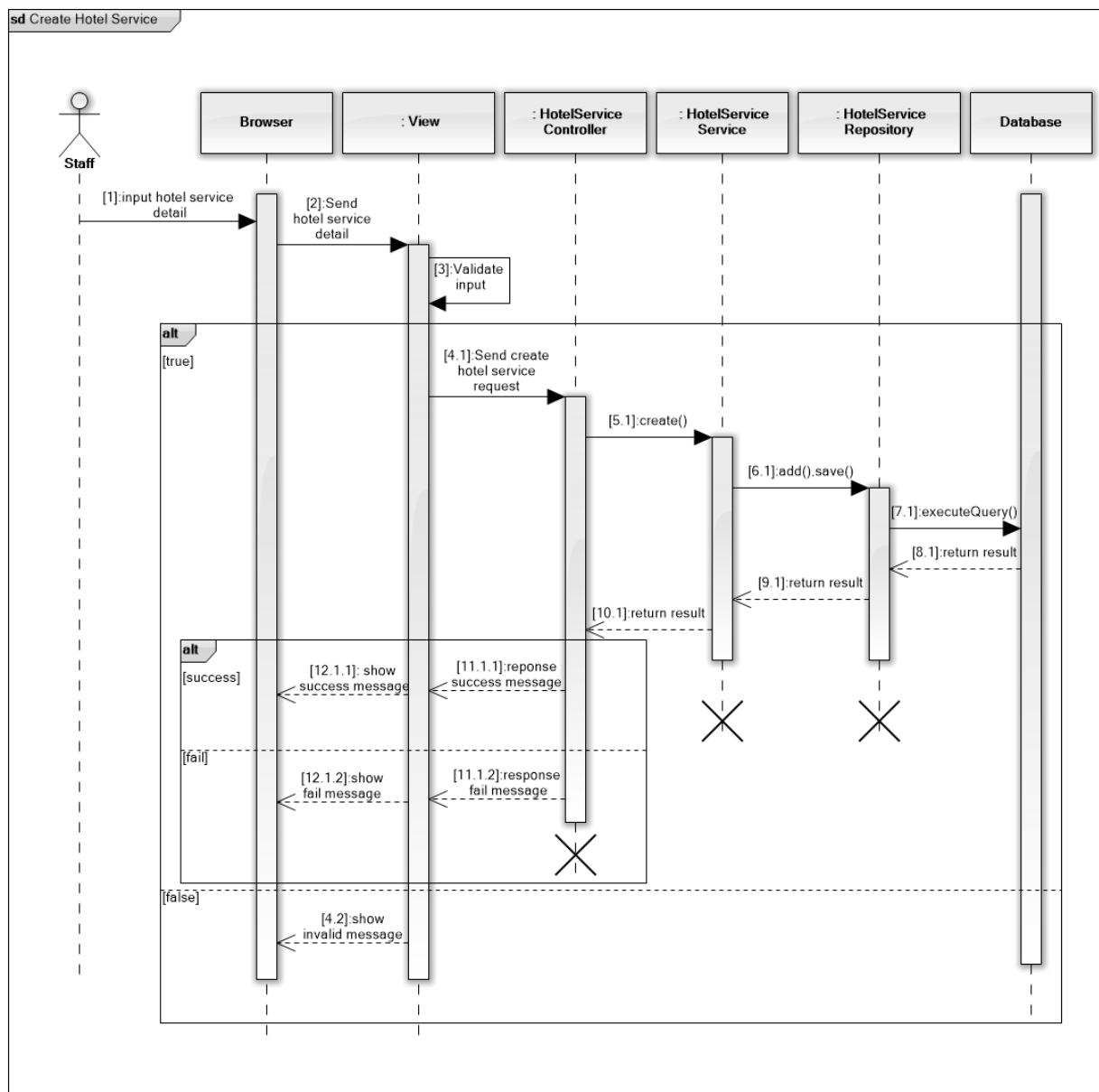
4.3.1.2.4. Manually start parsing scheduler



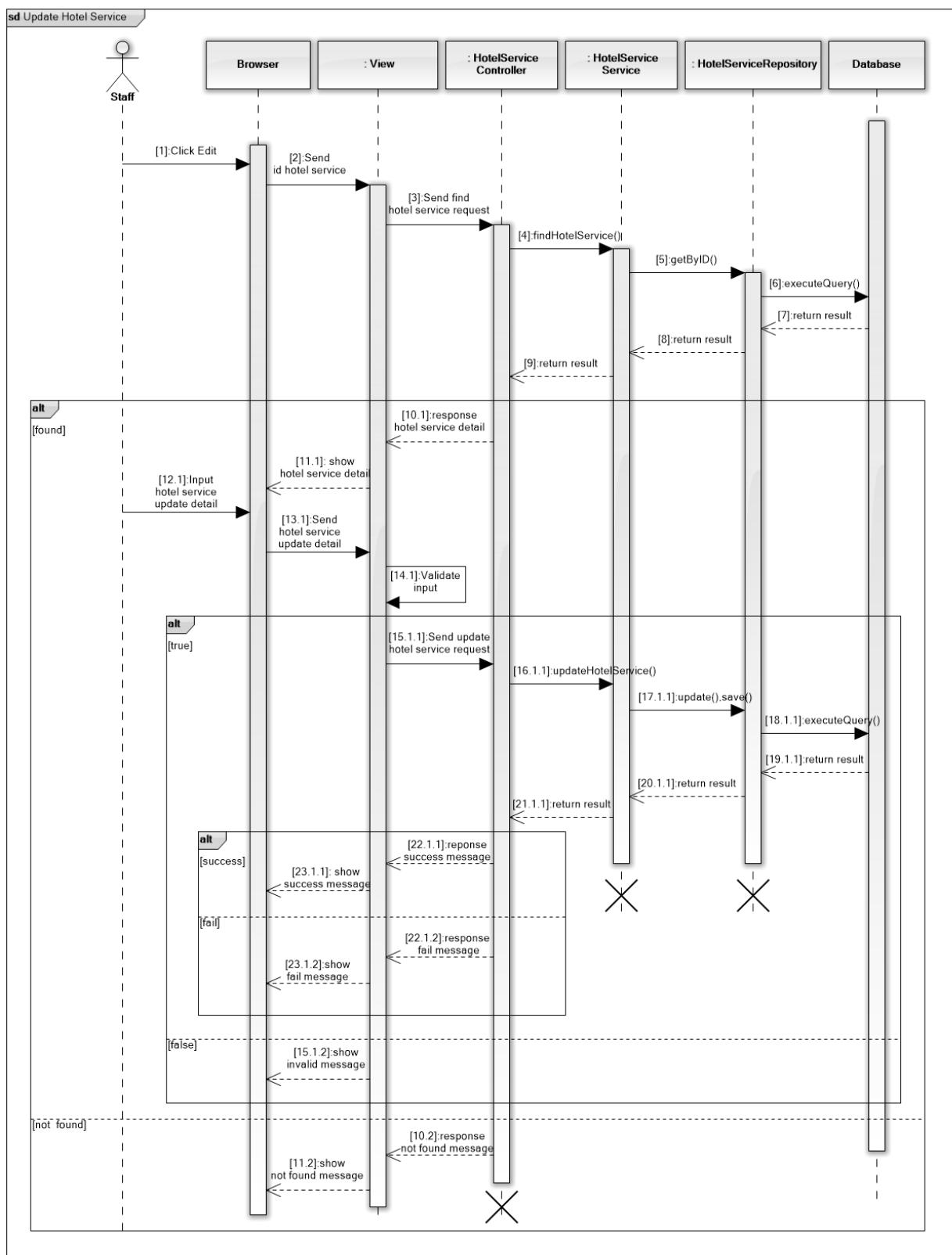
4.3.1.2.5. Configure parsing



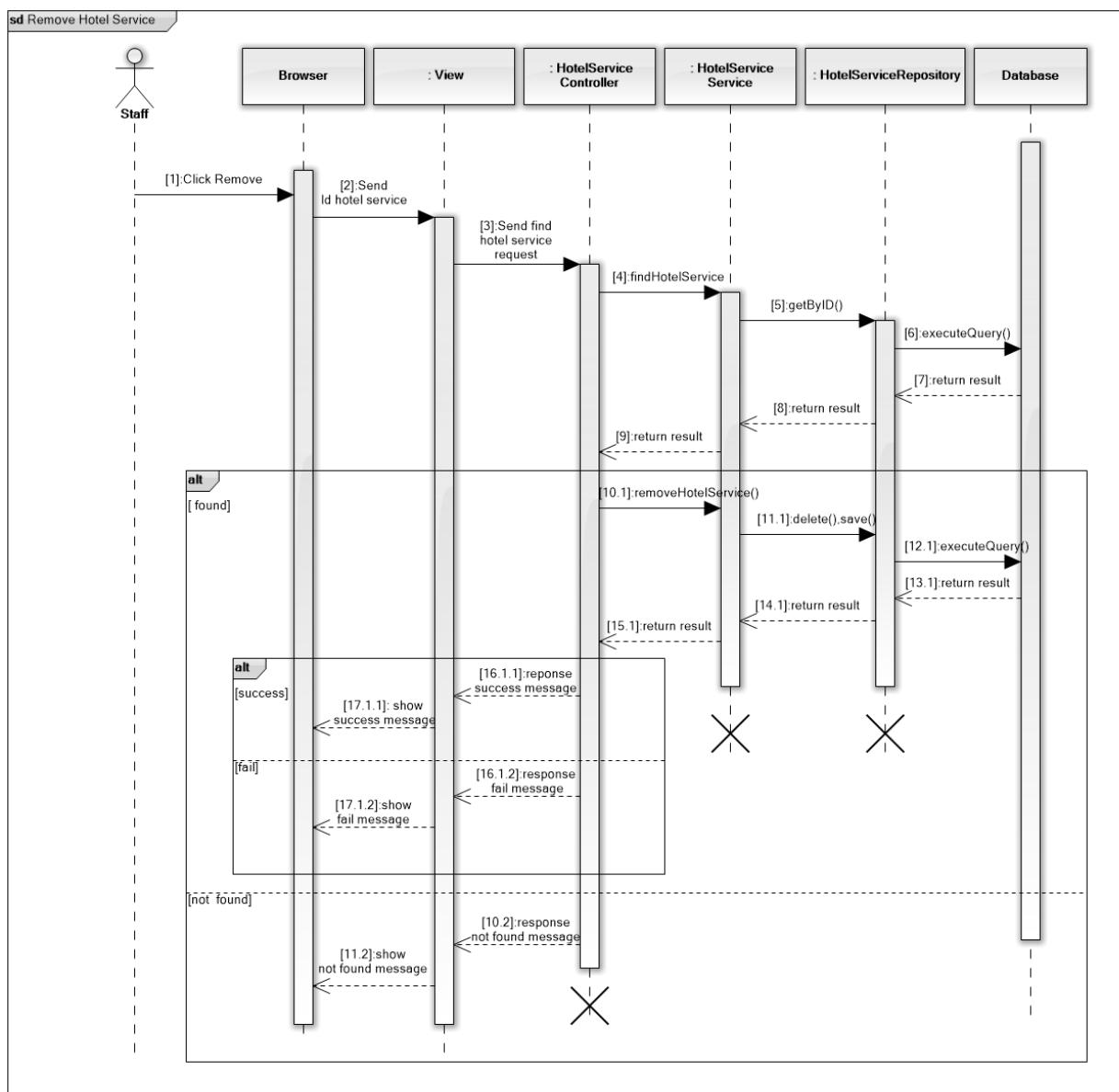
4.3.1.2.6. Create hotel service



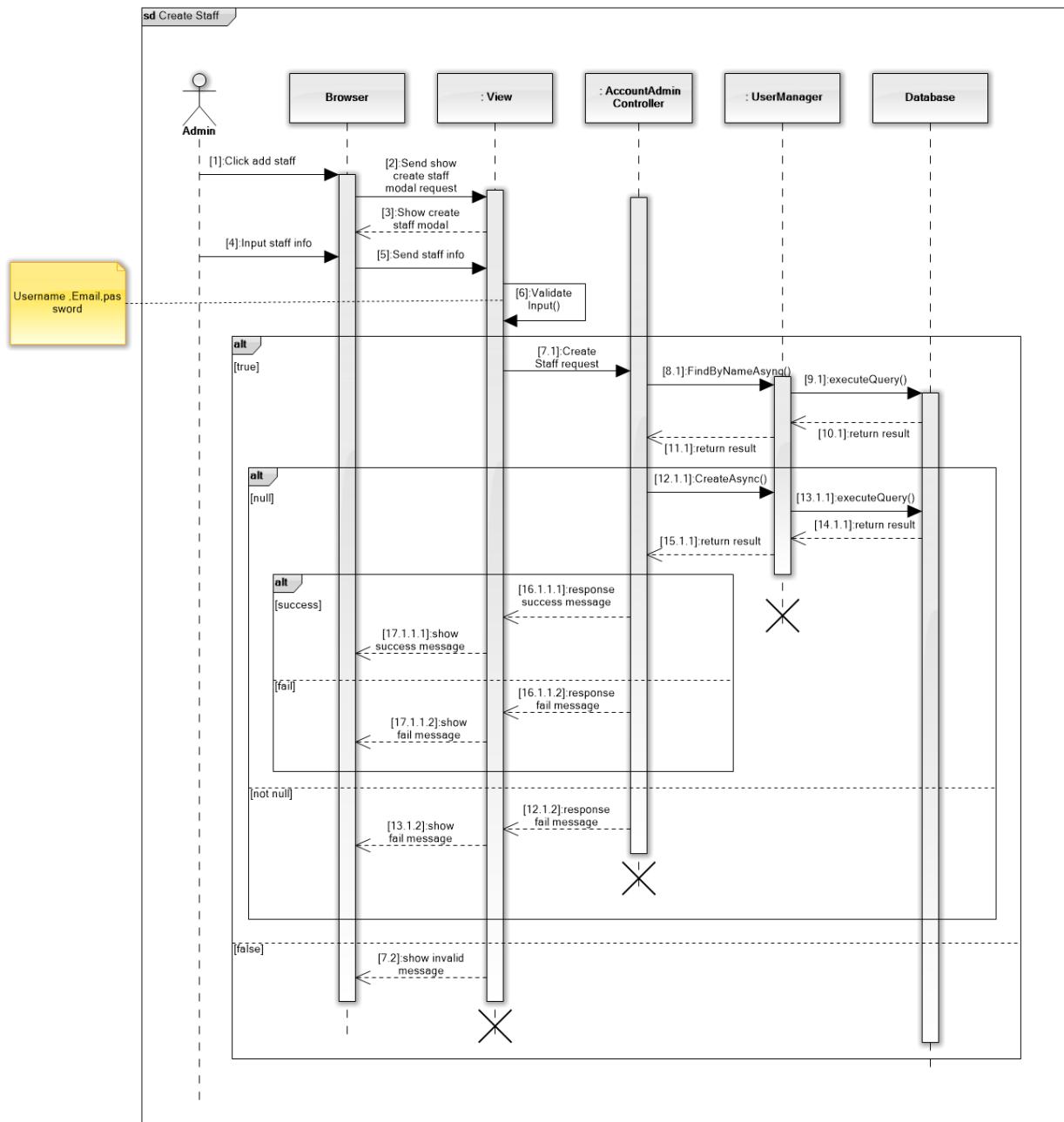
4.3.1.2.7. Edit hotel service



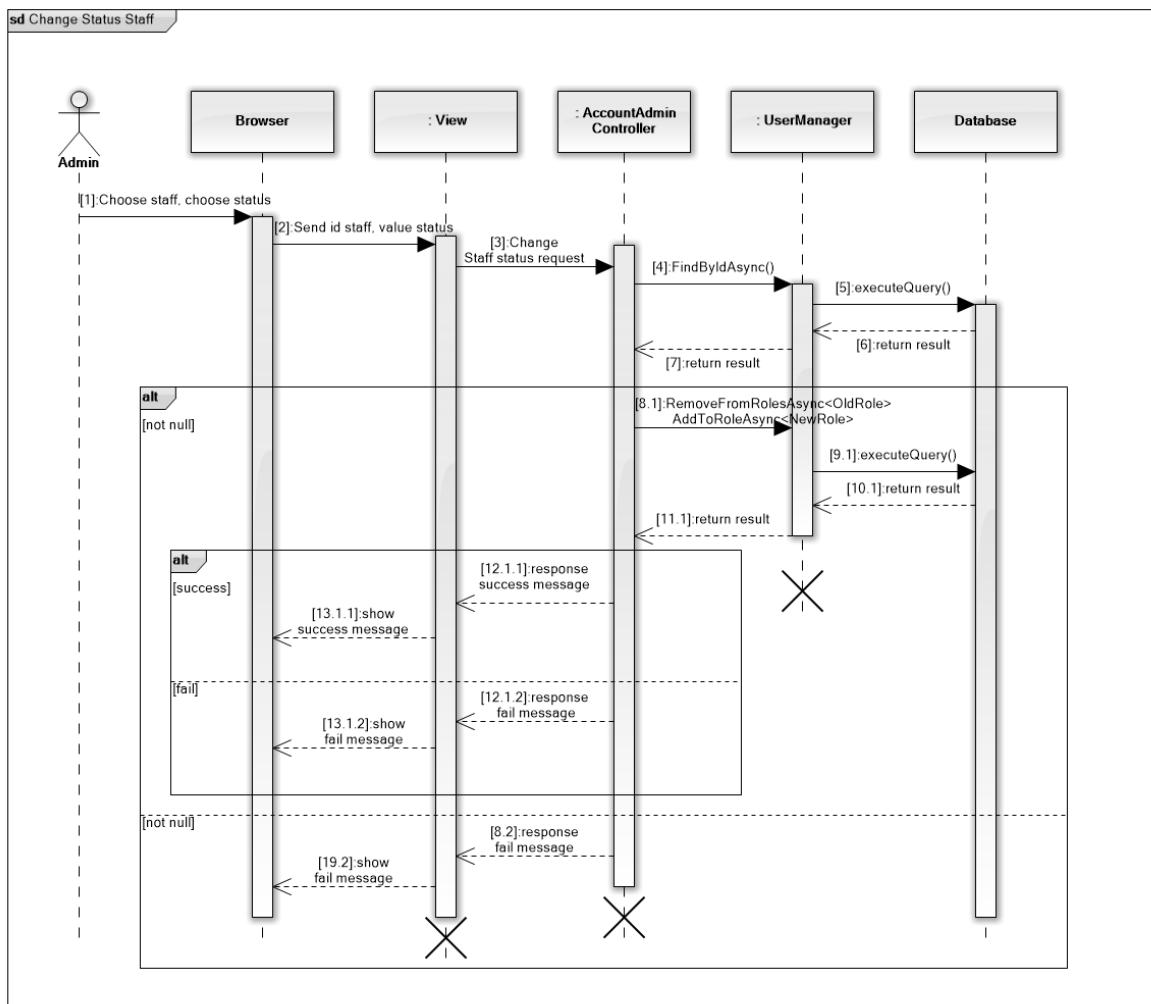
4.3.1.2.8. Remove hotel service



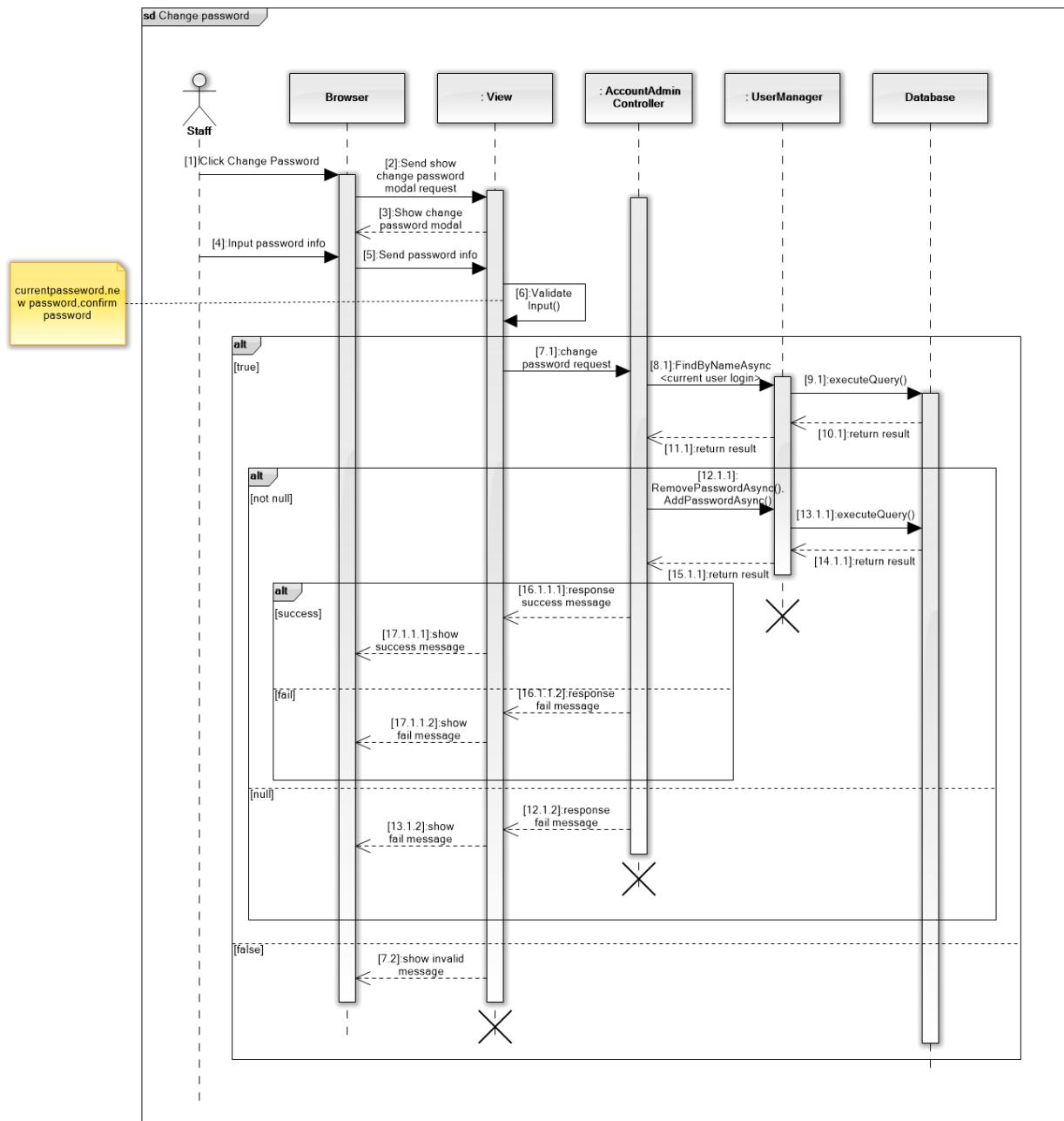
4.3.1.2.9. Create account



4.3.1.2.10. Activate/Deactivate account

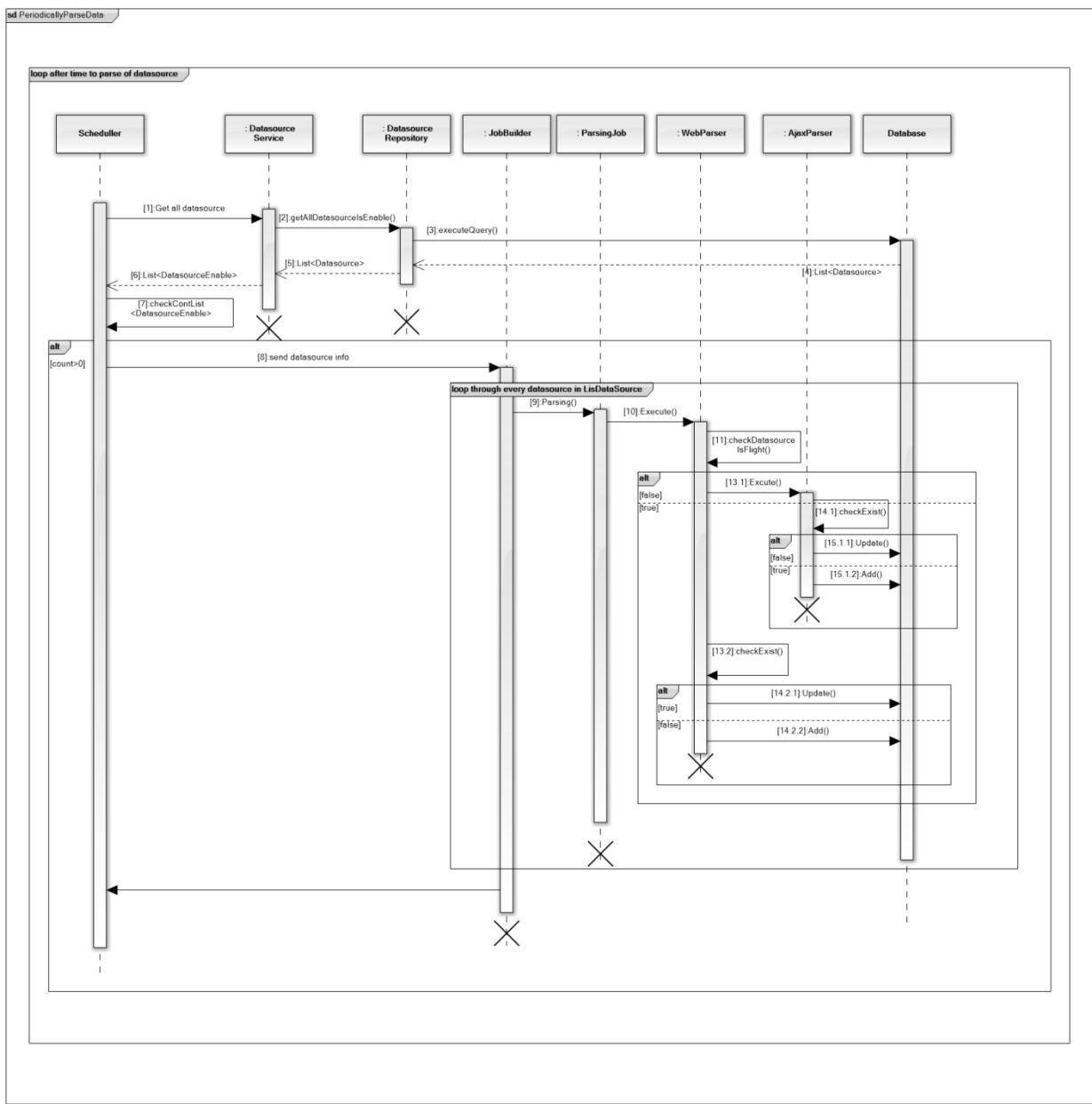


4.3.1.2.11. Change password

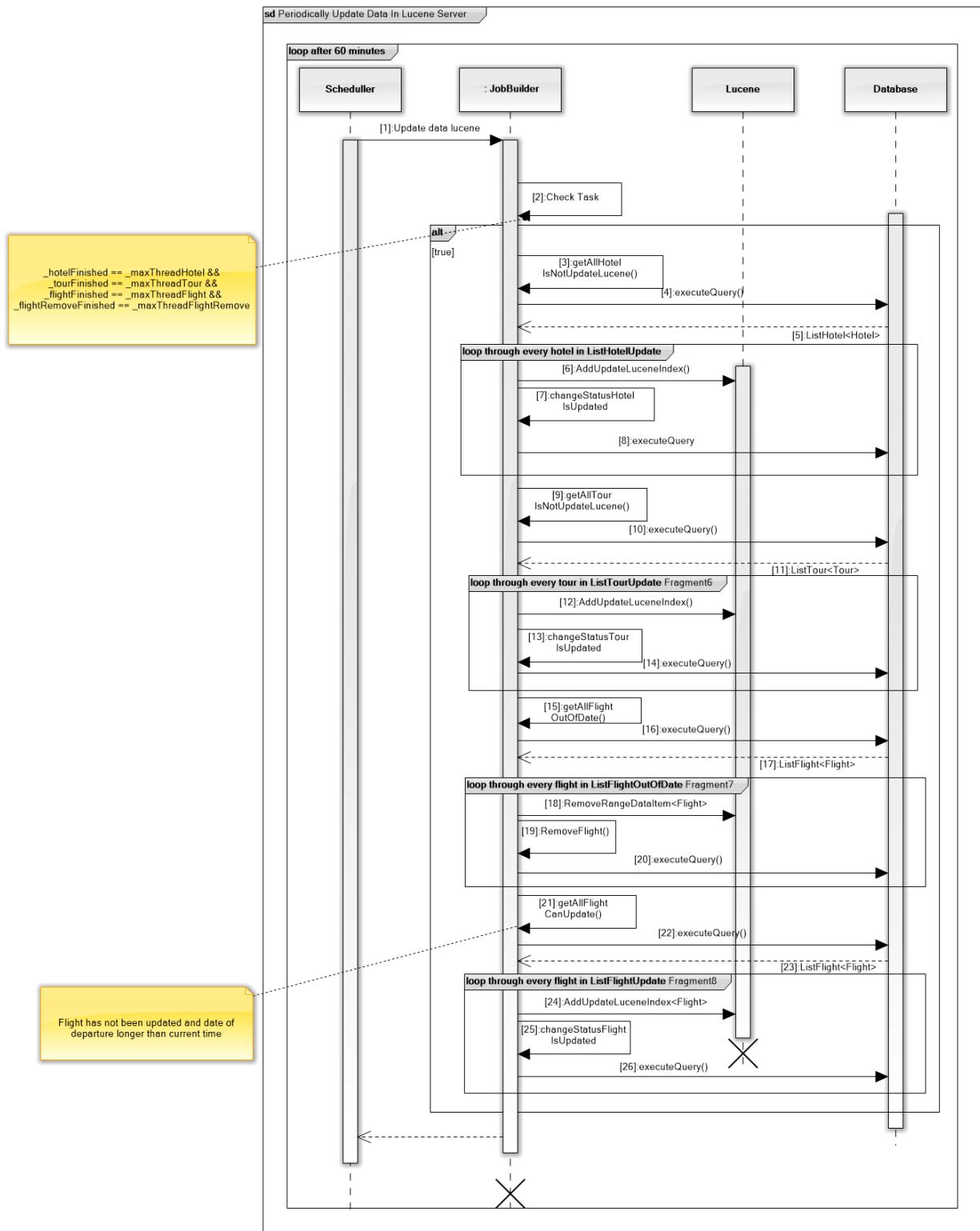


4.3.1.3. Scheduler

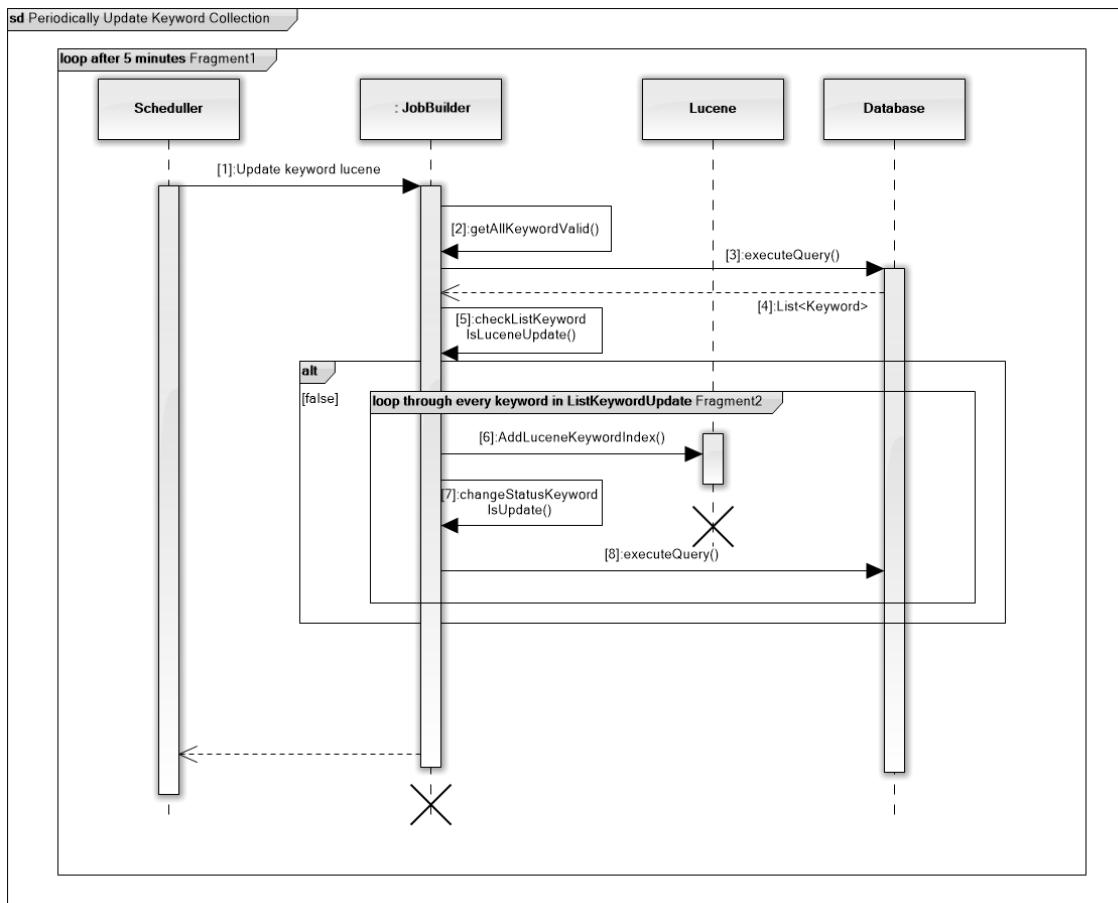
4.3.1.3.1. Parse data



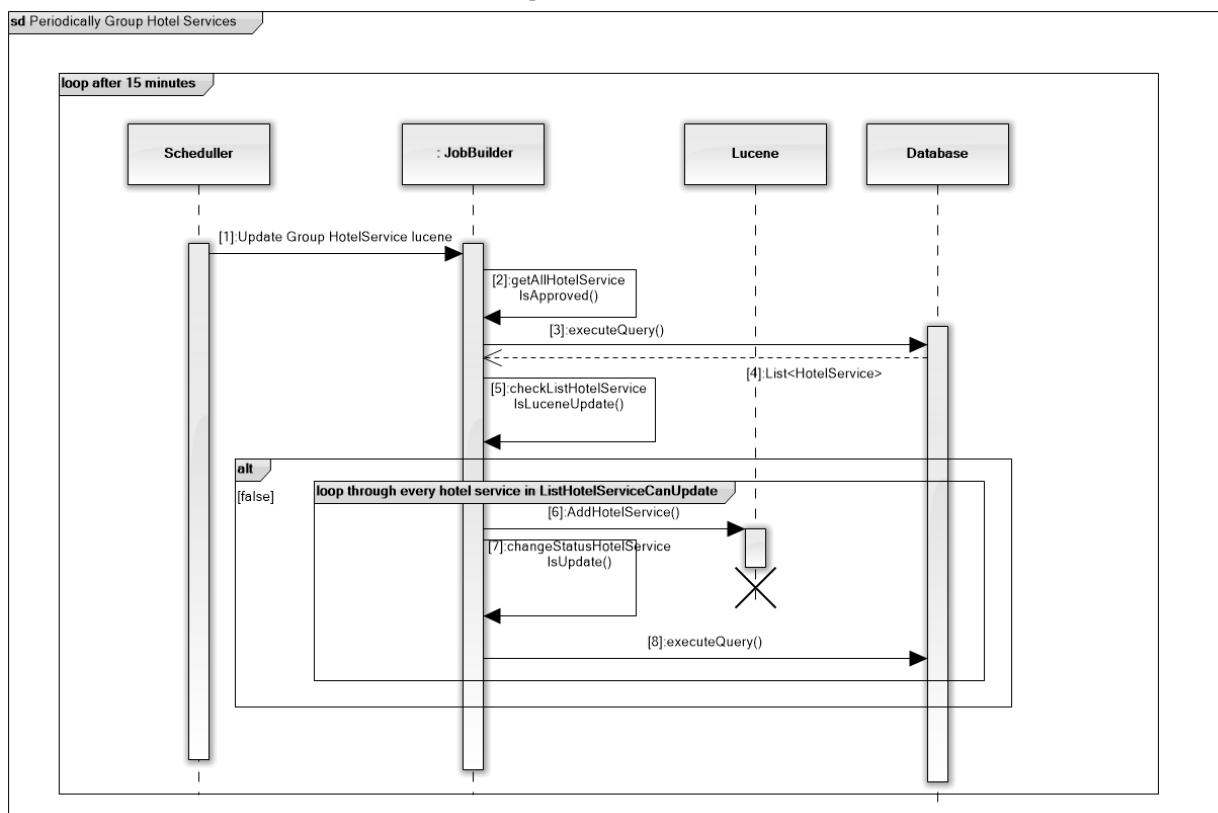
4.3.1.3.2. Update data to Lucene



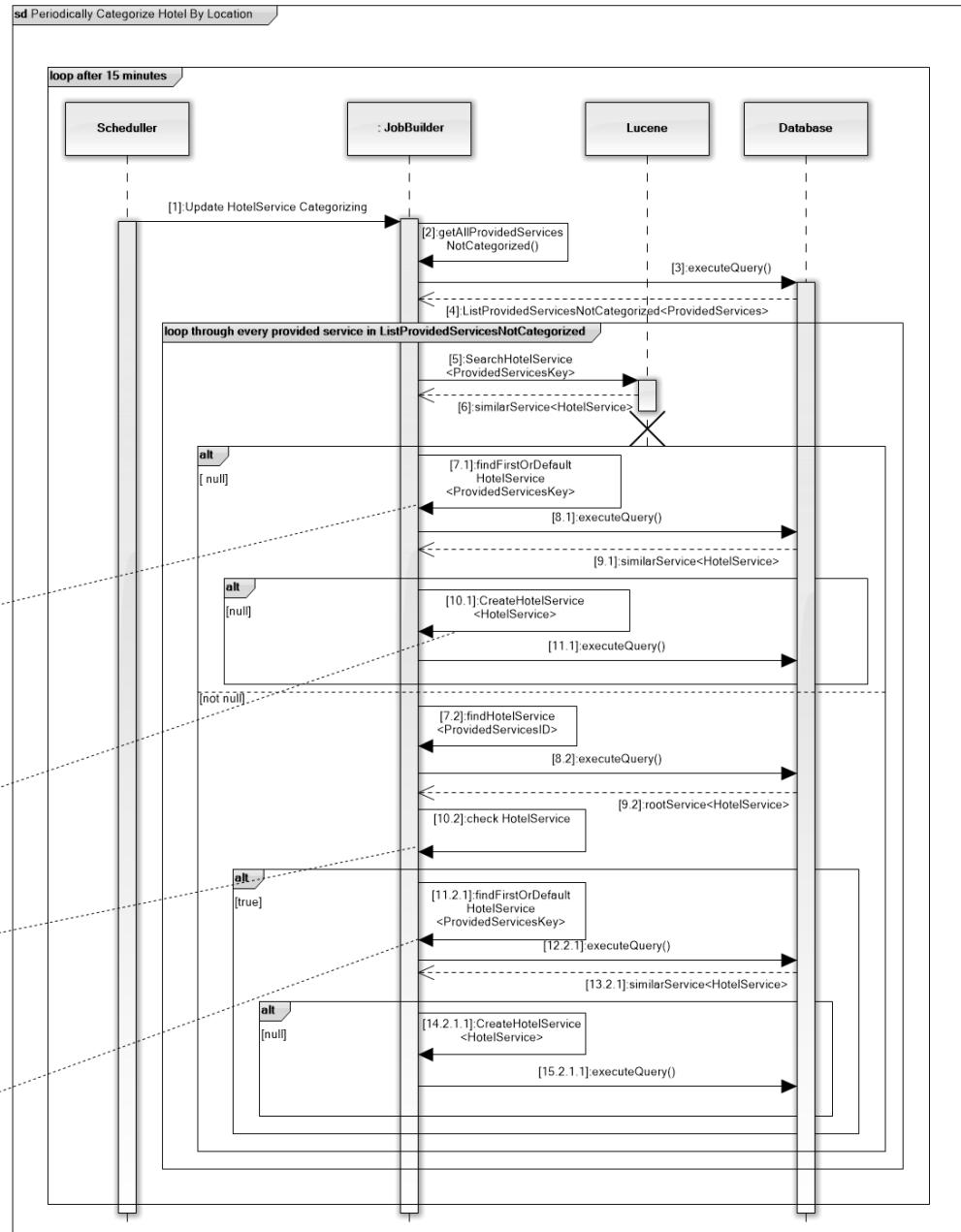
4.3.1.3.3. Remove redundant keywords



4.3.1.3.4. Categorize hotel services

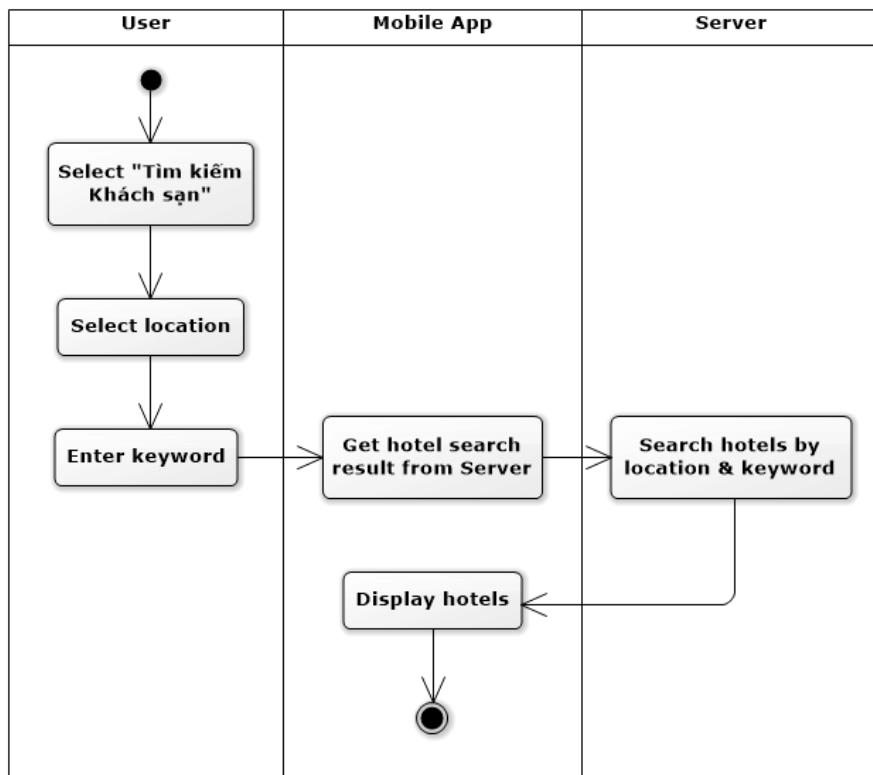


4.3.1.3.5. Categorize hotel by location

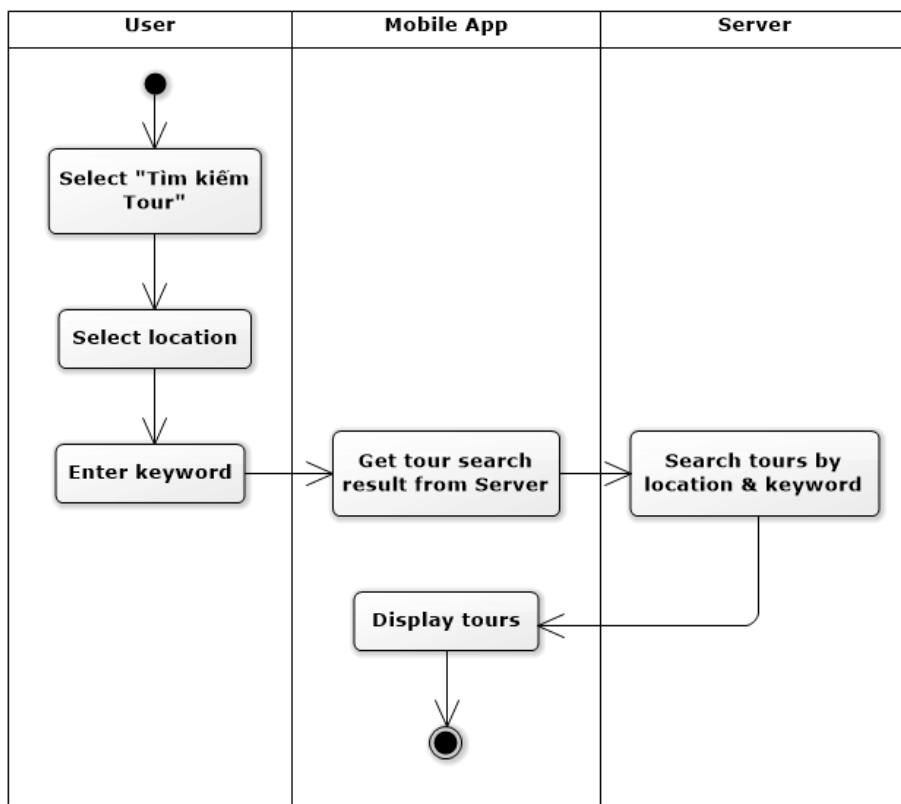


4.3.2. Mobile App

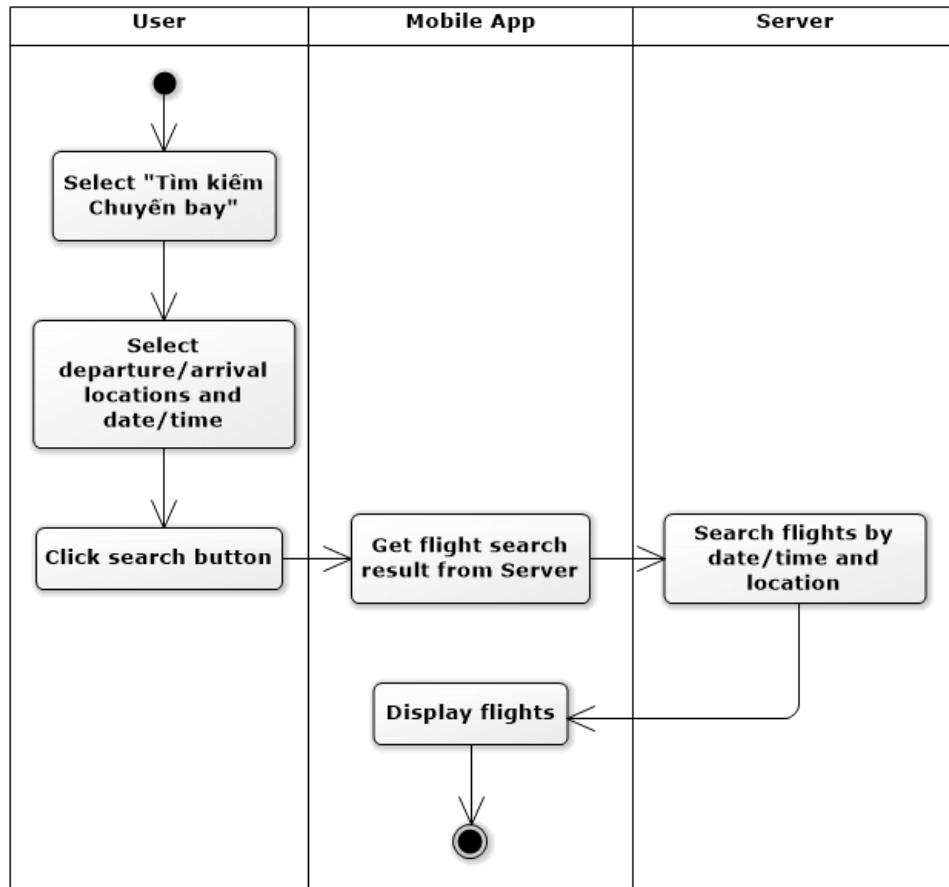
4.3.2.1. Search hotels



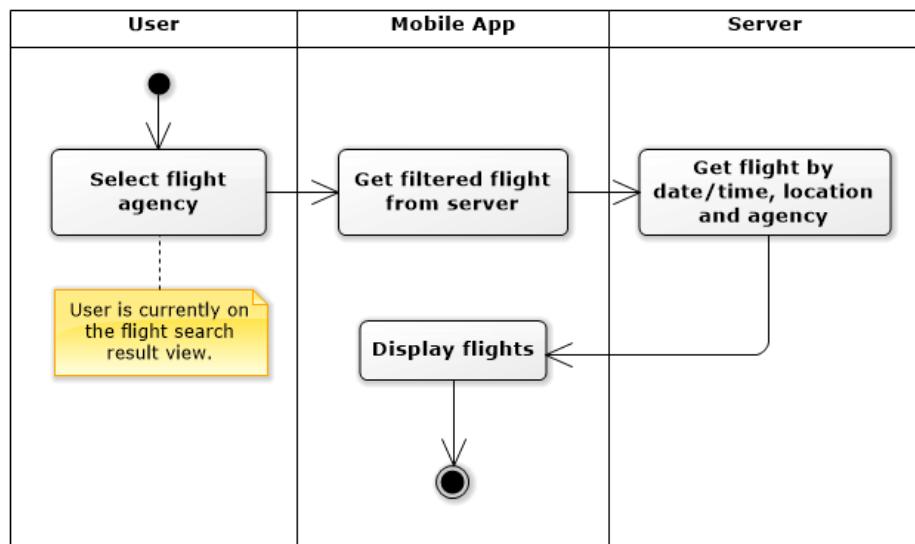
4.3.2.2. Search tours



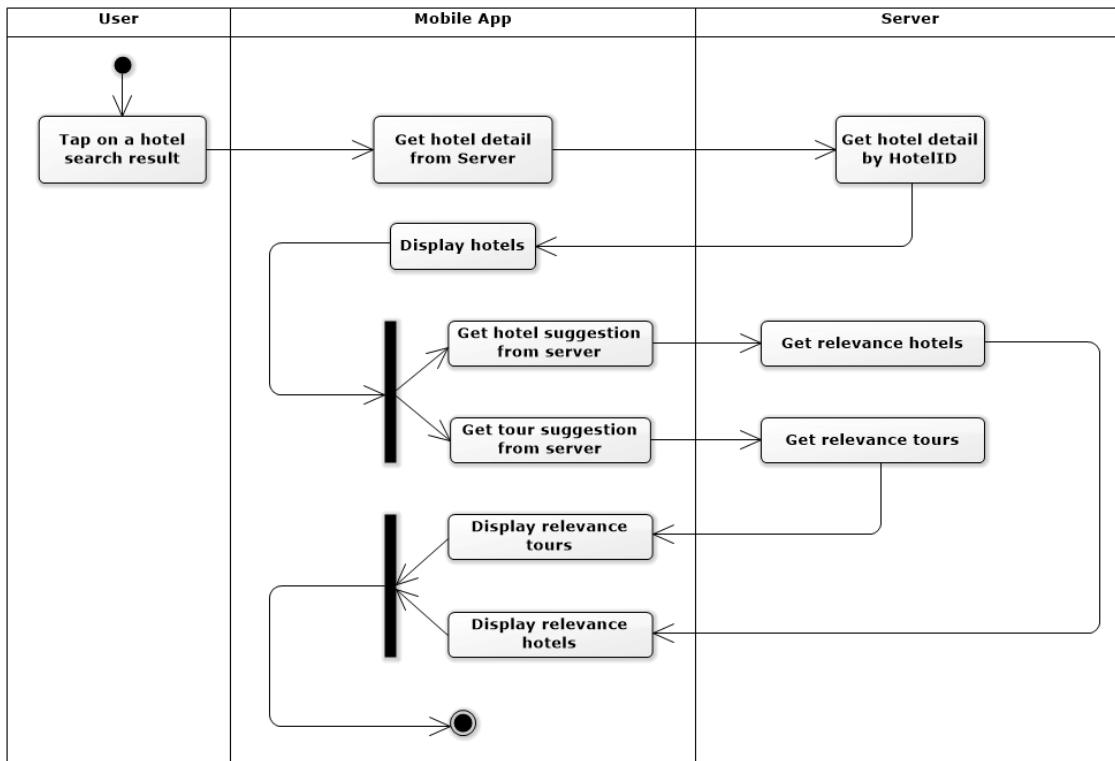
4.3.2.3. Search flights



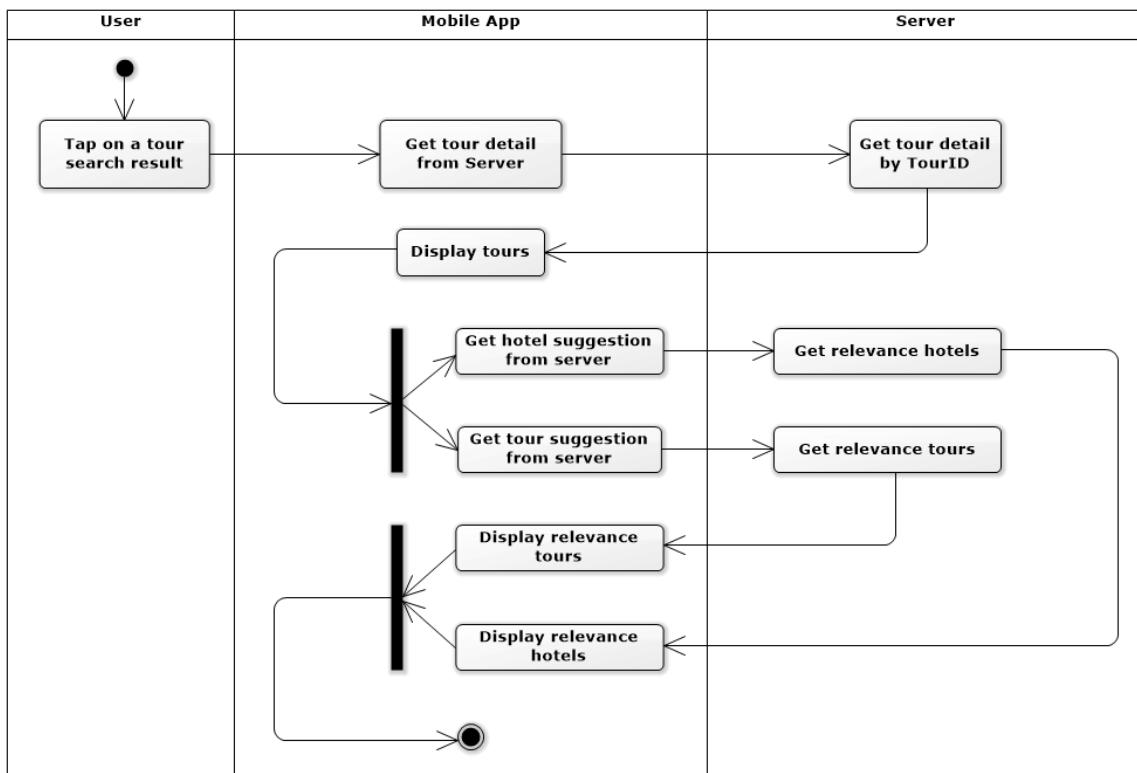
4.3.2.4. Filter flights



4.3.2.5. View hotel detail



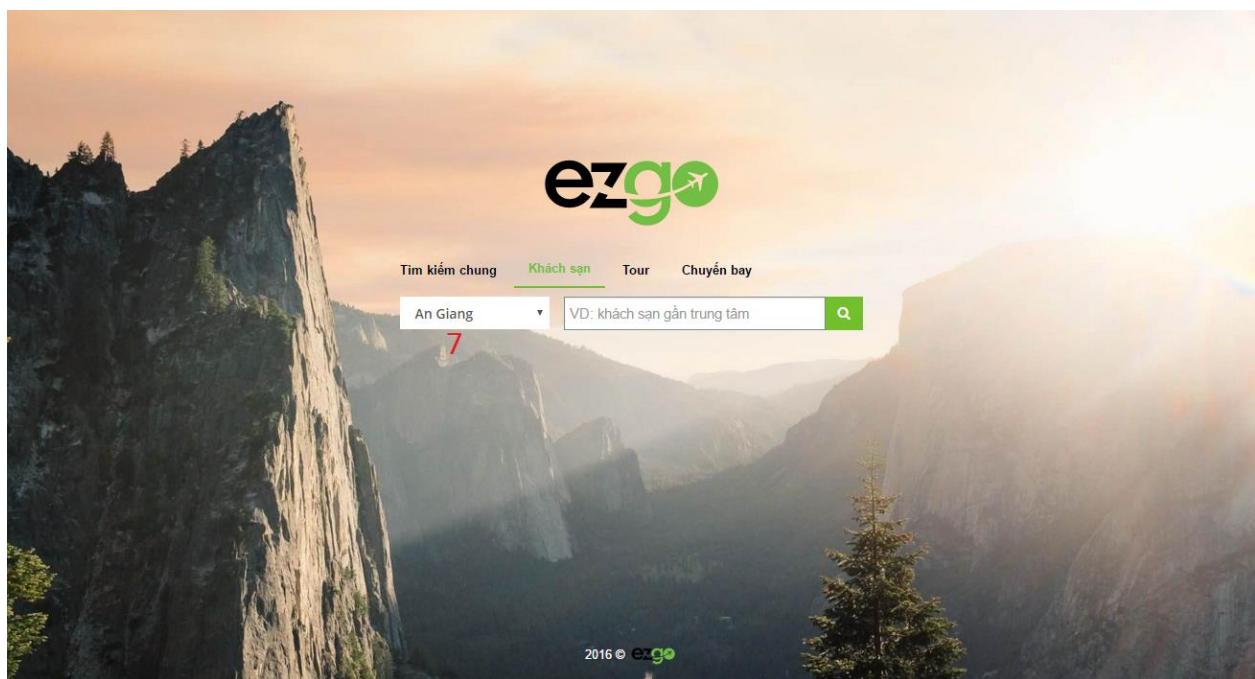
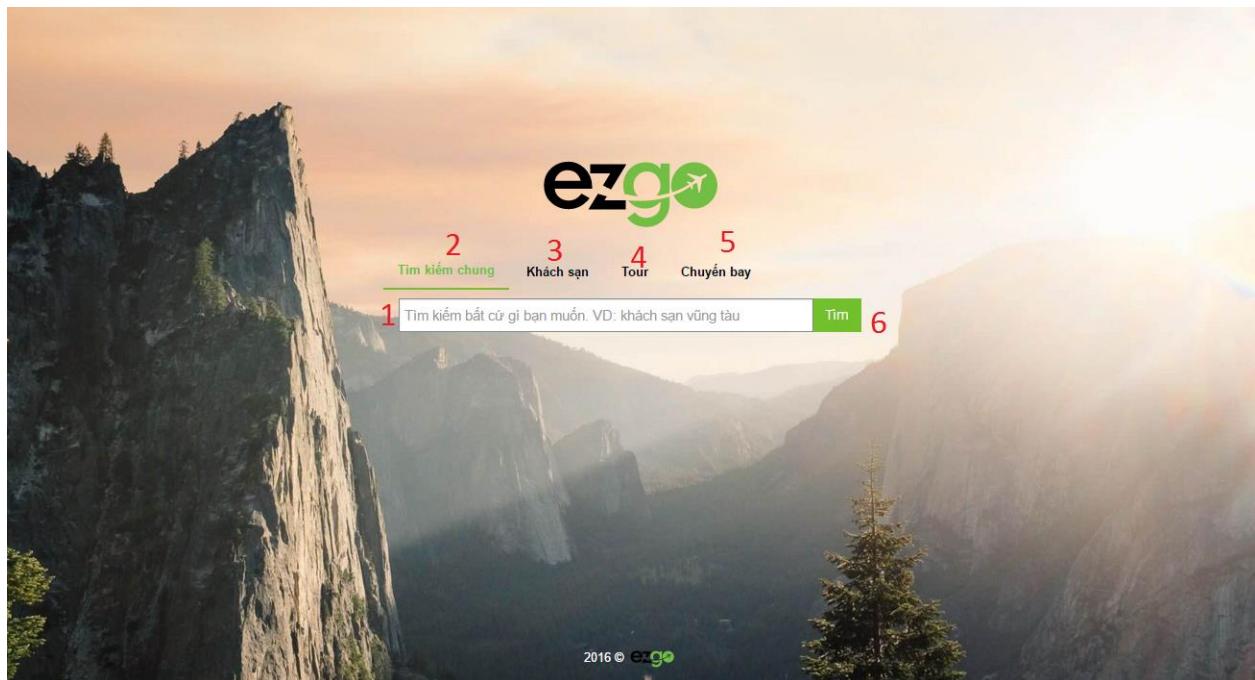
4.3.2.6. View tour detail



5. User Interface Design

5.1. Web app

5.1.1. Search



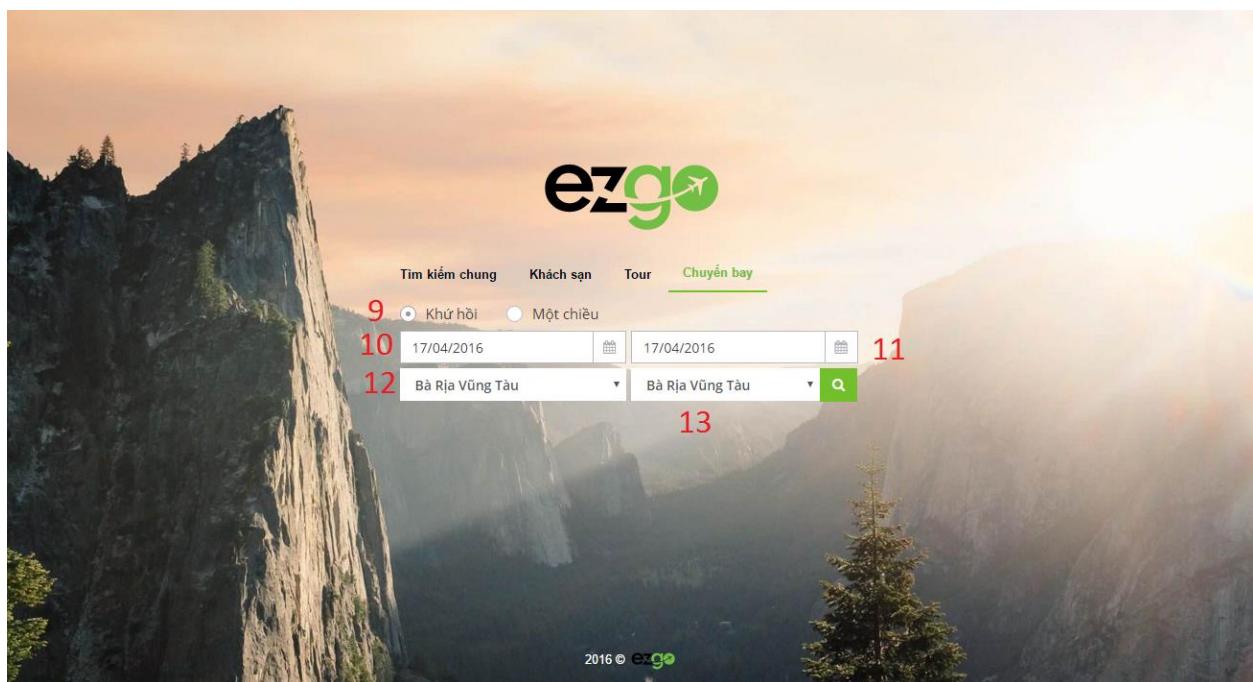
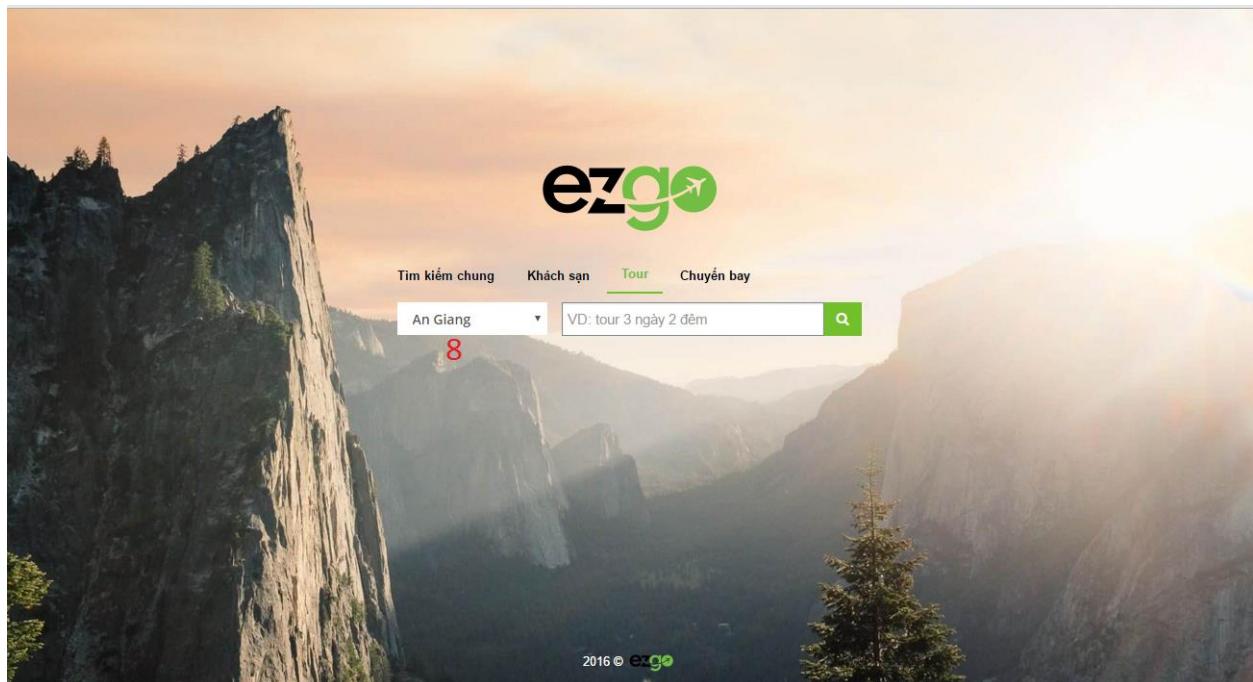


Figure: Web Application Design - <User>Search

Fields

No	Field Name	Description	Read Only	Mandatory	Control Type	Data Type
1	Search	Keyword to search	No	Yes	TextBox	String
7	Location	Location of hotel	No	Yes	Dropdown	Int

					n	
8	Location	Location of tour	No	Yes	Dropdown	Int
9	cbTypeFlight	Type of flight	No	Yes	Combobox	String
10	dpStart	Date of start flight	No	Yes	Datetime picker	Date
11	dpEnd	Date of end flight	No	Yes	Datetime picker	Date
12	LocationStart	Location of start flight	No	Yes	Dropdown	Int
13	LocationEnd	Location of end flight	No	Yes	Dropdown	Int

Buttons/Hyperlinks

No	Function	Description	Validation	Outcome
2	linkSearchGeneral	Show tab search general	N/A	Show tab search general
3	linkSearchHotel	Show tab search hotel	N/A	Show tab search hotel
4	linkSearchTour	Show tab search tour	N/A	Show tab search tour
5	linkSearchFlight	Show tab search flight	N/A	Show tab search flight
6	btnSearch	Send search command	Validate all required fields before submit.	Submit search.

5.1.2. Search result

Tìm được 16605 kết quả (khoảng 2.79 giây)

1 [Tất cả](#) 2 [Khách sạn \(1857\)](#) 3 [Tour \(373\)](#) 4 [Chuyến bay \(14375\)](#)
Kết quả tour cho *da nang*



Tour Du Lịch Đà Nẵng - Hội An - Bà...
5
<https://www.yesgo.vn/vi/tour/da-nang/tour-du-lich-da-nang...>
Giá vé: 3,990,000 VNĐ
vui lòng mang theo CMND/Passport... khách Việt Kiều
phải mang theo... : TP.HCM - ĐÀ NẴNG - HỘI AN Số...
chuyển bay đi Đà Nẵng. Tại sân bay Đà Nẵng, xe và
hướng dẫn...



Tour du lịch Đà Nẵng - Sơn Trà - H...
<http://mytour.vn/tour/8980-tour-du-lich-da-nang-son-tra-hoi...>
Giá vé: 3,550,000 VNĐ
sân nổi tiếng Đà Nẵng "Bánh tráng thịt heo 2 đầu da & M... 3 sao Đà Nẵng + 4 sao Huế... 1:ĐÀ NẴNG – BÁN ĐẢO SON TRÀ.... Đón quý khách tại Đà Nẵng (Sân...



Tour Du Lịch Đà Nẵng - Huế - Đà Nẵ...
<http://mytour.vn/tour/8190-tour-du-lich-da-nang-hue-da-nan...>

Từ khóa liên quan

da nang hoi an **6**
da nang . hoi an
A La Carte Da Nang Beach
da nag
da nng

Lọc kết quả khách sạn

Theo mức giá

- Dưới 500.000 **7**
- 500.000 - 1.000.000
- 1.000.000 - 1.500.000
- 1.500.000 - 2.000.000
- 2.000.000 - 2.500.000
- Trên 2.500.000

Theo dịch vụ khách sạn

- Bữa sáng miễn phí
- Bãi đỗ xe
- Bể bơi ngoài trời
- Giặt là
- Hỗ trợ đặt tour
- Internet
- Massage
- Nhà hàng
- Quán cafe
- Tắm hơi

Tìm được 1857 kết quả (khoảng 0.81 giây)

1 [Tất cả](#) 2 [Khách sạn \(1857\)](#) 3 [Tour \(373\)](#) 4 [Chuyến bay \(14375\)](#)



9.0 Khách sạn A La Carte Đà Nẵng

<https://www.ivivu.com/khach-san-da-nang/a-la-carte-da-na...>
200 Võ Nguyên Giáp, Quận Sơn Trà, Tp. Đà Nẵng, Việt Nam
Giá phòng: 2,030,000 VNĐ
Delight Plus tại A La Carte Da Nang Beach. 2...
A La Carte Da Nang... nổi tiếng của thành phố.
Tại A La Carte Da Nang Beach Hotel, mọi sự cố gắng
đều... muốn khám phá Đà Nẵng. Khách sạn A La...



9.0 Khách sạn A La Carte Đà Nẵng

<https://www.ivivu.com/khach-san-da-nang/a-la-carte-da-na...>
200 Võ Nguyên Giáp, Quận Sơn Trà, Tp. Đà Nẵng, Việt Nam
Giá phòng: 2,030,000 VNĐ
Delight Plus tại A La Carte Da Nang Beach. 2...
A La Carte Da Nang... nổi tiếng của thành phố.
Tại A La Carte Da Nang Beach Hotel, mọi sự cố gắng
đều... muốn khám phá Đà Nẵng. Khách sạn A La...

Từ khóa liên quan

da nang hoi an
da nang . hoi an
A La Carte Da Nang Beach
da nag
da nng

Lọc kết quả vé máy bay

Theo hãng

- VietNam Airline
- JetStar
- Vietjet

Theo giờ bay

- Giờ bay
- Giờ đến

- 00:00 - 06:00
- 06:00 - 12:00
- 12:00 - 18:00
- 18:00 - 00:00

Sắp xếp

- Giá tăng dần
- Giá giảm dần
- Giờ bay tăng dần
- Giờ bay giảm dần

Tìm được 14375 kết quả (khoảng 1.41 giây)

Tất cả Khách sạn (1857) Tour (373) Chuyến bay (14375)

Chuyến bay	Giờ bay - Giờ đến	Giá vé	
Lâm Đồng - Đà Nẵng (VN 1954)	08:30 - 09:50	2,515,000	9 Xem chi tiết
Hải Phòng - Đà Nẵng (VN 1673)	12:30 - 13:40	2,405,000	Xem chi tiết
Đà Nẵng - Lâm Đồng (VN 1957)	14:20 - 15:45	2,020,000	Xem chi tiết
Đà Nẵng - Lâm Đồng (VN 1955)	06:30 - 07:55	2,020,000	Xem chi tiết
Lâm Đồng - Đà Nẵng (VN 1956)	12:25 - 13:45	2,515,000	Xem chi tiết
Đà Nẵng - Hà Nội (VN 716)	08:00 - 09:20	2,790,000	Xem chi tiết
Đà Nẵng - Hà Nội (VN 16)	07:35 - 08:55	2,515,000	Xem chi tiết
Đăk Lăk - Đà Nẵng (VN 1912)	08:25 - 09:30	1,910,000	Xem chi tiết
Hồ Chí Minh - Đà Nẵng (VN 13)	19:40 - 21:00	2,515,000	Xem chi tiết
Hồ Chí Minh - Đà Nẵng (VN 1)	06:15 - 07:35	1,360,000	Xem chi tiết

Từ khóa liên quan

- da nang hoi an
- da nang , hoai an
- A La Carte Da Nang Beach
- da nag
- da nng

Fields

No	Field Name	Description	Read Only	Mandatory	Control Type	Data Type
7	Filter	Filter of search result	No	No	Checkbox	N/A

Buttons/Hyperlinks

No	Function	Description	Validation	Outcome
1	linkSearchGeneral	Show tab search general	N/A	Show tab search general
2	linkSearchHotel	Show tab search hotel	N/A	Show tab search hotel
3	linkSearchTour	Show tab search tour	N/A	Show tab search tour
4	linkSearchFlight	Show tab search flight	N/A	Show tab search flight
5	linkDetail	Open result detail	N/A	Open result detail page.
6	linkKeywordRelationship	Search with	N/A	Submit search.

		word have relationship with keyword		
8	linkHotelsNear	Detail hotel near.	N/A	Submit detail.
9	btnShowDetail	Show detail of flight	N/A	Show detail flight.

5.1.3. Search detail

Khách sạn liên quan

1 Khách Sạn A La Carte Đà Nẵng Beach
Võ Nguyên Giáp, Quận Sơn Trà, TP. Đà Nẵng, Việt Nam (Xem bản đồ)
★ ĐÁNH GIÁ: 9.0

2 Tour liên quan

Tour Free&Easy Đà Nẵng - khách sạn A La Carte trong 3 ngày 2 đêm
Giá tour: 5,300,000 VNĐ

3

MÔ TẢ

THÔNG TIN KHUYẾN MÃI:

4

GIÁ PHÒNG

Giá thấp nhất: **2,030,000 VNĐ**

Xem trang gốc

Khách sạn lân cận

Khách Sạn Misa Đà Nẵ...
11 Dương Đình Nghệ, Sơn Trà... Xem →

Khách sạn Trường Sơn...
Số 7 Hà Bồng, Phường Phước... Xem →

Khách sạn Trường Sơn...
Số 2 Hà Bồng, Phường Phước ... Xem →

Khách sạn Trường Sơn...
Số 2 Hà Bồng, Phường Phước ... Xem →

Bình luận (5)

Tôi thấy khách sạn này cũng không có gì đặc biệt tuy nhiên tôi thấy Lễ Tân làm thủ tục CI và CO Rất tốt

TOUR DU LỊCH ĐÀ NẴNG - HỘI AN - BÀ NÀ - SƠN TRÀ



[GIÁ TOUR](#)

Có thể bạn quan tâm

5 Khách Sạn Misa Đà Nẵng
11 Đường Đinh Nghệ, Sơn Trà, Đà Nẵng
Giá phòng: Đang cập nhật

Khách Sạn Đà Nẵng Bay
Số 52, Đường Võ Nguyên Giáp, Phường Mân Thái, Sơn Trà, Đà Nẵng
Giá phòng: Đang cập nhật

Mayfair Hotel & Apartment Đà Nẵng
14 Loseby, Sơn Trà, Đà Nẵng
Giá phòng: Đang cập nhật

Khách Sạn Sion Đà Nẵng
121/7 Hoàng Văn Thụ, quận Hải Châu, Đà Nẵng
Giá phòng: Đang cập nhật

Khách Sạn A La Carte Đà Nẵng Beach
Võ Nguyên Giáp, Dương Đình Nghệ, Sơn Trà, Đà Nẵng
Giá phòng: Đang cập nhật

Tour liên quan

[Tour du lịch Đà Nẵng - Bà Nà - Hội An - Sơn Trà trong 3 ngày 2 đêm](#)
Giá tour: 2,200,000 VND

[Tour Du Lịch Huế - Đà Nẵng- Sơn Trà](#)

Buttons/Hyperlinks

No	Function	Description	Validation	Outcome
1	linkHotelRelationship	Detail hotel relationship	N/A	Detail hotel relationship
2	linkTourRelationship	Detail tour relationship	N/A	Detail tour relationship
3	linkHomePage	Redirect to home page of detail	N/A	Redirect to home page of detail
4	linkHotelsNear	Detail hotel near.	N/A	Submit detail.
5	linkSuggest	Detail result maybe user care	N/A	Submit detail

5.1.4. Manage datasources

The screenshot shows the 'Data Source' section of the EzGo Admin interface. At the top, there is a dropdown for 'Data type' set to 'Hotel' (1), a 'Root URL' input field (2), and a 'Sample URL' input field (3). A green '+' button (5) is located in the top right corner. Below these fields is a search bar with a placeholder 'Search:' (4) and a dropdown for 'Show 10 entries'. The main area displays a table of five datasource entries:

#	Name	Type	Interval	Last update	Auto-run	CONFIG	FORCE START	Edit / Remove
1	http://www.abay.vn	Flight	5 minutes	26/01/2016	<input checked="" type="checkbox"/> 6	<input type="button" value="CONFIG"/> 7	<input type="button" value="FORCE START"/> 8	<input type="button" value="Edit / Remove"/> 9
2	http://mytour.vn	Hotel	120 minutes	08/04/2016	<input checked="" type="checkbox"/>	<input type="button" value="CONFIG"/>	<input type="button" value="FORCE START"/>	<input type="button" value="Edit / Remove"/>
3	http://mytour.vn/tour	Tour	30 minutes	17/04/2016	<input checked="" type="checkbox"/>	<input type="button" value="CONFIG"/>	<input type="button" value="FORCE START"/>	<input type="button" value="Edit / Remove"/>
4	https://avia.vn	Flight	5 minutes	13/06/1905	<input checked="" type="checkbox"/>	<input type="button" value="CONFIG"/>	<input type="button" value="FORCE START"/>	<input type="button" value="Edit / Remove"/>
5	http://www.skytour.vn/Khach-san	Hotel	120 minutes	17/04/2016	<input checked="" type="checkbox"/>	<input type="button" value="CONFIG"/>	<input type="button" value="FORCE START"/>	<input type="button" value="Edit / Remove"/>

Fields

No	Field Name	Description	Read Only	Mandatory	Control Type	Data Type
1	Data type	Type of datasource	No	Yes	DropDown	Int
2	Root URL	URL of datasource	No	Yes	TextBox	String
3	Sample URL	Sample URL of datasource	No	Yes	TextBox	String
4	Search datasource	Search datasource	No	No	TextBox	String

Buttons/Hyperlinks

No	Function	Description	Validation	Outcome
5	btnAdd	Send add new datasource command	Validate all required fields before submit.	Submit add new datasource.
6	btnAutoRun	Send auto run command	N/A	Change status auto-run datasource.

7	btnConfig	Open config	N/A	Open config page
8	btnForceStart	Send force start parsing data.	N/A	Submit force start parsing data.
9	btnEdit	Edit datasource	N/A	Open datasource edit modal.
10	btnRemove	Remove datasource	N/A	Show delete datasource pop-up

5.1.5. Edit datasource

The screenshot shows a modal dialog titled '+ Edit data source'. It has a dropdown menu for 'Data type' set to 'Flight' (marked with red number 1) and a text input for 'Data source URL' containing 'http://wwwabay.vn/' (marked with red number 2). At the bottom right are three buttons: 'CLOSE' (gray), 'SAVE CHANGES' (green, marked with red number 4), and a small red number 3.

Fields

No	Field Name	Description	Read Only	Mandatory	Control Type	Data Type
1	Datatype	Type of datasource	No	Yes	DropDown	Int
2	Data source URL	URL of datasource	No	Yes	TextBox	String

Buttons/Hyperlinks

No	Function	Description	Validation	Outcome
3	btnClose	Close edit datasource modal	N/A	Close edit datasource modal
4	btnSave	Send change datasource command	Validate all required fields before submit.	Submit save change datasource.

5.1.6. Config datasource

Parsing Configuration

Data source
http://www.abay.vn/ **1**

MinProxyCount
40 **2**

MaxThreadCount
1111 **3**

MaxResponse
30000 **4**

MaxAttemp
3 **5**

SAVE CHANGES **6**

Fields

No	Field Name	Description	Read Only	Mandatory	Control Type	Data Type
1	Data source	URL data source	Yes	No	TextBox	String
2	MinProxyCount	Min proxy count	No	Yes	TextBox	Int
3	MaxThreadCount	Max thread count	No	Yes	TextBox	Int
4	MaxResponse	Max Response	No	Yes	TextBox	Int
5	MaxAttemp	Max attemp	No	Yes	TextBox	Int

Buttons/Hyperlinks

No	Function	Description	Validation	Outcome
6	btnChange	Send change config datasource command	Validate all required fields before submit.	Submit change config datasource.

Screenshot of the EZGO application interface showing a data source configuration page.

The interface includes a sidebar with navigation links and a main content area for managing a "Hotel" data type from a "Data source".

Key elements in the main area:

- Data type:** Hotel 1
- Data source:** http://mytour.vn 2
- Buttons:** SAVE CHANGES 14, LOAD 13
- Form Fields:**
 - Name 3
 - PreviewImgUrl 4
 - Address 5
 - Description 6
 - DescriptionHtml 7
 - Price 8
 - Rating 9
 - ProvidedServices
 - ServiceName 10
 - Comments
 - CustomerName 11
- Sample URL:** http://mytour.vn/10505-khach-san-hoa-binh-1-an-giang.html 12
- Rating and Image Preview:** A large image of a modern hotel building with a rating of 8.1 (26 đánh giá) 15.
- Footer:** ez-go.net/Admin/DataSource/GetPage?url=http://mytour.vn/10505-khach-san-hoa-binh-1-an-giang.html#

Fields

No	Field Name	Description	Read Only	Mandatory	Control Type	Data Type
1	Data type	Type datasource	Yes	No	TextBox	String
2	Data source	URL of datasource	Yes	No	TextBox	String
3	Name	Rating of data	No	Yes	TextBox/ DropDownList	N/A
4	PreviewImgUrl	PreviewImgUrl of data	No	No	TextBox/ DropDownList	N/A
5	Address	Address of data	No	No	TextBox/ DropDownList	N/A
6	Description	Description of data	No	No	TextBox/ DropDownList	N/A
7	DescriptionHtml	DescriptionHtml of data	No	No	TextBox/ DropDownList	N/A
8	Price	Price of data	No	No	TextBox/ DropDownList	N/A

					n	
9	Rating	Rating of data	No	No	TextBox/ Dropdown n	N/A
10	ServiceName	Name of service	No	No	TextBox/ Dropdown n	N/A
11	CustomerName	Name of customer who commented.	No	No	TextBox/ Dropdown n	N/A
12	Sample datasource	Sample URL datasource	Yes	No	TextBox	String
15	IframSample	Load page sample	No	No	Iframe	N/A

Buttons/Hyperlinks

No	Function	Description	Validation	Outcome
13	btnLoad	Load sample page into Iframe	N/A	Load sample page into Iframe
14	btnSave	Send change config command	N/A	Submit change config.

5.1.7. Manage hotelservices

SERVICES **NEW SERVICES**

Name **1** **2** **3**

Synonyms

Show 10 entries

Search: **4**

#	Name	Synonyms	Action
1	Bữa sáng miễn phí	Bữa sáng miễn phí	5 Edit / Remove 6
2	Bãi đỗ xe	Có đỗ xe Có bãi Giữ xe miễn phí Bãi đậu xe ô tô tại khách sạn	Edit / Remove
3	Bể bơi ngoài trời	Hồ bơi	Edit / Remove
4	Giặt là	Dịch vụ giặt úi	Edit / Remove
5	Hỗ trợ đặt tour	Dịch vụ du lịch	Edit / Remove

SERVICES **NEW SERVICES**

Show 10 entries

Search:

#	Name	Synonyms	Expected Service	Action
1	Dịch vụ trông trẻ	Dont have data	Trông trẻ	7 APPROVE ALL 8 Edit / Remove
2	Dịch vụ tự giặt úi	Dont have data	Giặt là	APPROVE Edit / Remove
3	Hồ bơi ngoài trời	Dont have data	Bể bơi ngoài trời	APPROVE Edit / Remove
4	Hồ bơi trong nhà	Dont have data	Bể bơi trong nhà	APPROVE Edit / Remove
5	Kết sắt an toàn	Dont have data	Dont have data	APPROVE Edit / Remove
6	Nhà hàng, Khu ăn uống	Dont have data	Nhà hàng	APPROVE Edit / Remove

Fields

No	Field Name	Description	Read Only	Mandatory	Control Type	Data Type
1	Name	Name of hotelservice	No	Yes	TextBox	String
2	Synonyms	Synonyms of hotelservice	No	Yes	TagsInput	String

4	Search	Search hotelservice	No	No	TextBox	String
---	--------	---------------------	----	----	---------	--------

Buttons/Hyperlinks

No	Function	Description	Validation	Outcome
3	btnAdd	Send add new hotelservice command	Validate all required fields before submit.	Submit add new hotelservice.
5	btnEdit	Edit hotelservice	N/A	Open hotelservice edit modal.
6	btnRemove	Remove hotelservice	N/A	Show delete hotelservice pop-up
7	btnApproveAll	Send approve all hotelservice command	N/A	Submit approve all hotelservice.
8	btnApprove	Send approve one hotelservice command	N/A	Submit approve one hotelservice.

5.1.8. Edit hotelservice

Edit Hotel Service ×

Name
Bữa sáng miễn phí 1

Synonyms
Bữa sáng miễn phí x 2

3 CLOSE SAVE CHANGES 4

Fields

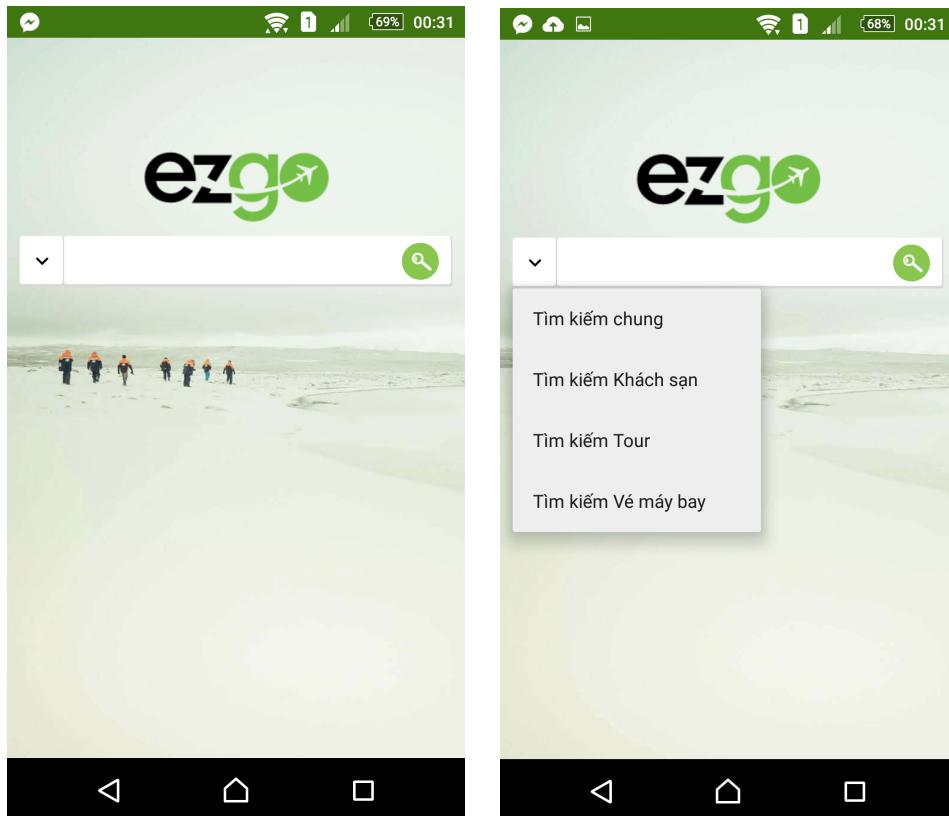
No	Field Name	Description	Read Only	Mandatory	Control Type	Data Type
1	Name	Name of hotelservice	No	Yes	TextBox	String
2	Synonyms	Synonyms of hotelservice	No	No	TagsInput	String

Buttons/Hyperlinks

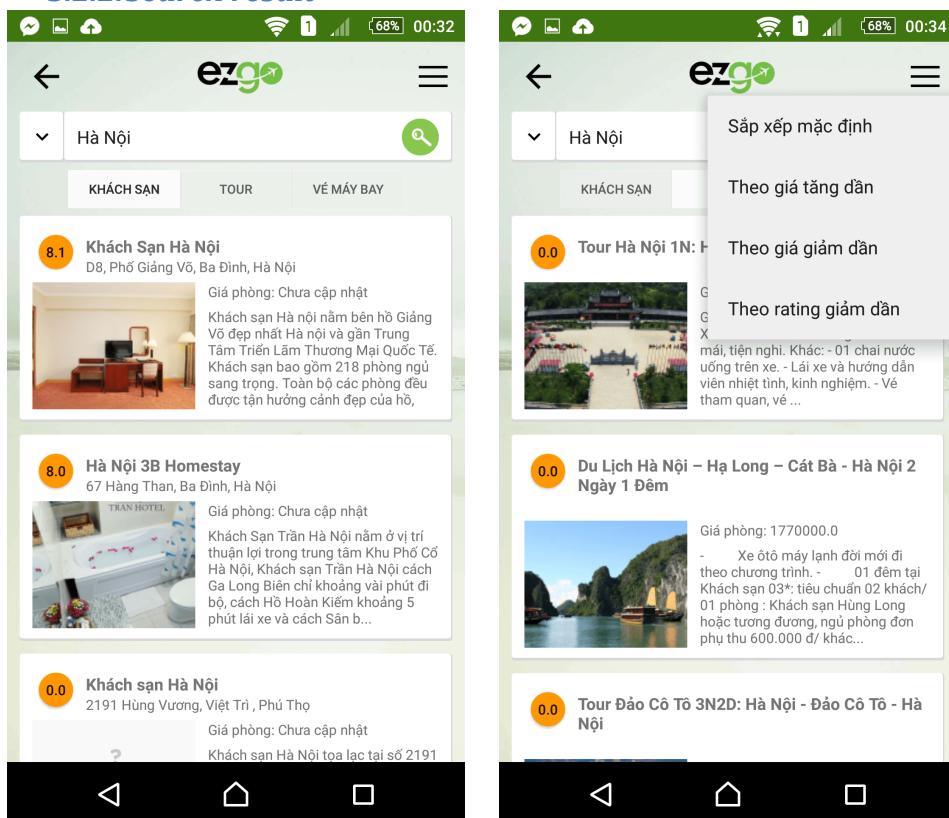
No	Function	Description	Validation	Outcome
3	btnClose	Close edit hotelservice modal	N/A	Close edit hotelservice modal
4	btnSave	Send change hotelservice command	Validate required fields before submit.	Submit save change hotelservice .

5.2. Mobile app

5.2.1. Home



5.2.2. Search result



5.2.3. Detail

The image consists of two side-by-side screenshots from a mobile application.

Screenshot 1 (Left): Hotel Review

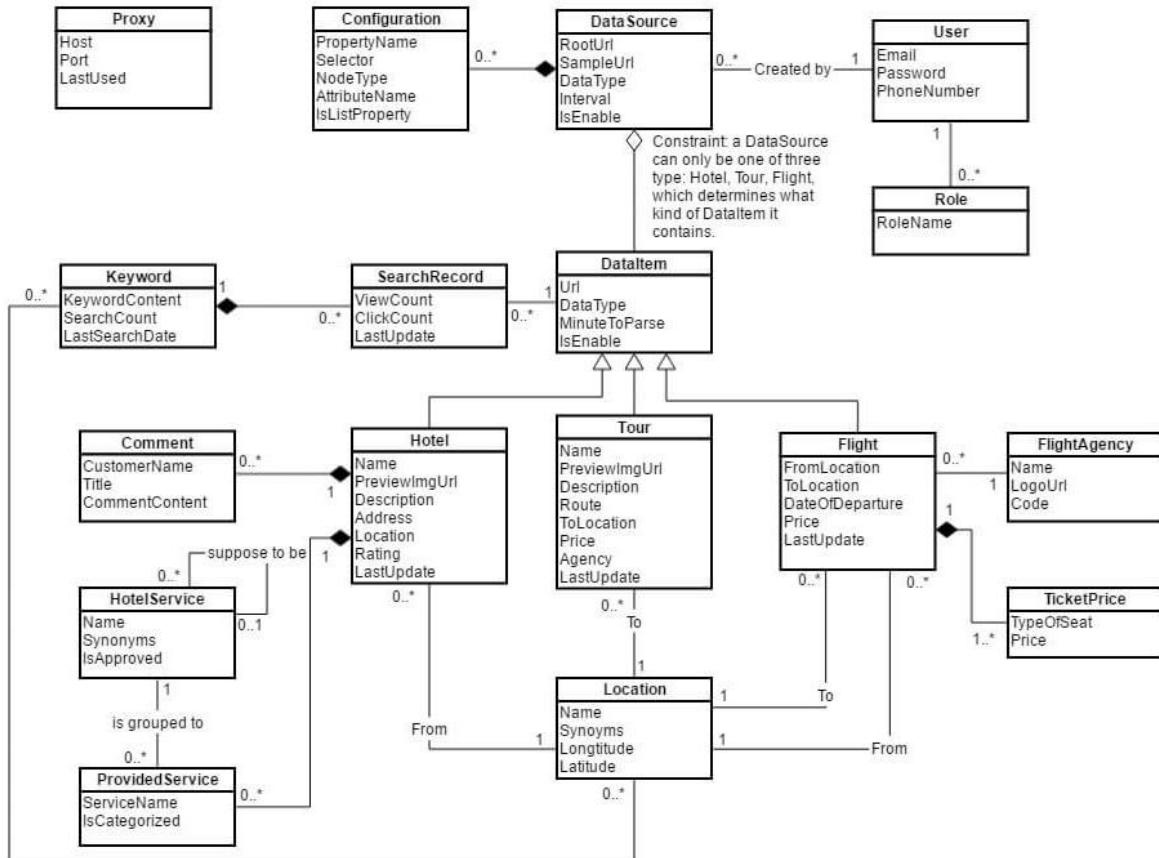
- Header:** Shows signal strength, battery level (68%), and time (00:34).
- Image:** A photograph of a hotel room interior featuring a desk with a computer monitor, a lamp, and a chair.
- Title:** Khách Sạn Hà Nội (Hanoi Hotel).
- Rating:** 8.1 (highlighted in orange).
- Address:** D8, Phố Giảng Võ, Ba Đình, Hà Nội.
- Section:** Giới thiệu chung (Introduction).
- Text:** Khách sạn Hà nội nằm bên hồ Giảng Võ đẹp nhất Hà nội và gần Trung Tâm Triển Lãm Thương Mại Quốc Tế. Khách sạn bao gồm 218 phòng ngủ sang trọng. Toàn bộ các phòng đều được tận hưởng cảnh đẹp của hồ, thành phố. Các phòng được trang bị tiện nghi hiện đại đem đến cho quý khách sự hài lòng và thoải mái....
- Section:** Bình luận (Comments).
- Comment by Nguyễn Hằng Nga:** "Hài lòng".
"Thật ra lần này tôi có đặt phòng hộ cho khách của nhà tôi, mỗi lần đến Hà Nội họ đều muốn nghỉ ngơi tại đây. Họ bảo phòng ốc tiện nghi, điều kiện vệ sinh sạch sẽ. Vị trí khách sạn thuận tiện đi lại mà giá cả lại phải."

Screenshot 2 (Right): Tour Recommendations

- Header:** Khách Sạn Hà Nội.
- Comment by HUANG JIE:** "Phòng sạch sẽ"
"Trong chuyến đi vừa qua thì tôi có đặt phòng tại khách sạn Hà Nội qua công ty mytour. Căn phòng mà tôi ở tại khách sạn khá thoải mái và sạch sẽ.Thêm vào đó cũng khá đủ tiện nghi cho tôi sử dụng. Nhân viên cũng khá ok."
- Section:** Có thể bạn quan tâm (You might be interested in).
- Options:** TRAN HOTEL, ? (camera icon), Khách Sạn Ngọc, Khách Sạn V...
- Section:** Tour liên quan (Related tours).
- Tours:** Tour Hà Nội 1N: Hà, Tour Đảo Cô Tô, Du Lịch Hà Ni.

6. Database Design

6.1. Entity Relationship Diagram



6.2. Entity dictionary

Entity Name	Description
DataSource	Contains all the information about websites to be parsed.
Configuration	Specifies needed properties for parsing data.
Hotel	Contains all the properties about hotels.
Flight	Contains all the properties about tours.
Tour	Contains all the properties about flights.
Keyword	Recorded from search keywords of users.
SearchRecord	Used to update click count & view count from user behaviors.
Location	Used to categorize data by cities/provinces.
Comment	Parsed comments associated with hotels.

HotelService	Official hotel services used for filtering search results.
ProvidedService	Parsed services associated with hotels.
FlightAgency	Parsed agencies associated with hotels.
TicketPrice	Categorize flight data into 3 type of tickets: Adult, Children, Orphan
Proxy	Contains the list of proxies used to parse data without being banned by the data sources.
User	Store information of admin and staff of the system.
Role	Has 2 types: Admin, Staff

7. Algorithms

7.1. Parse data from Html

Definition

- Automatically parse all the appropriate data from html of a given website.

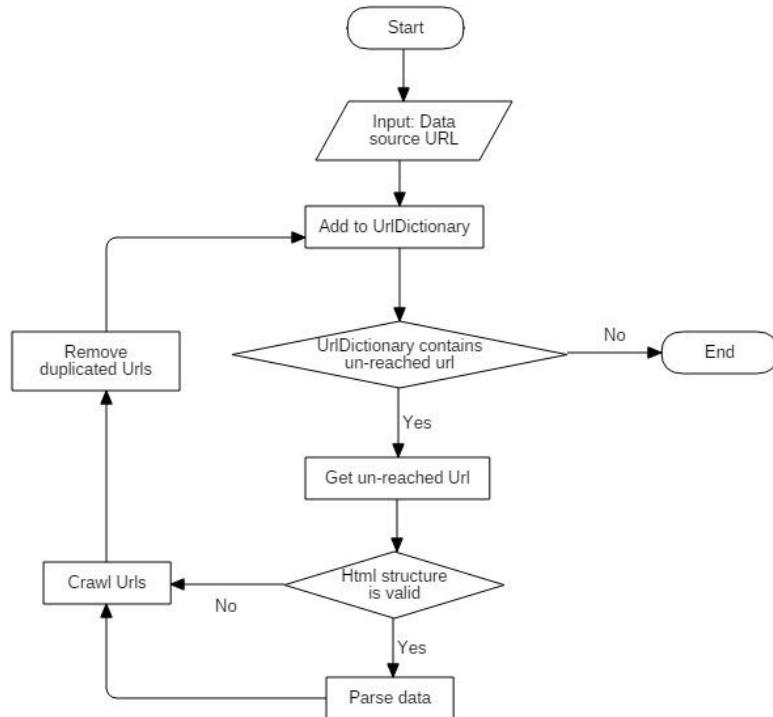
Problem Definition

- Each website has a different html structure.
- There are various types of data to parse.
- There are many valid and unvalid urls to parse.

Solution

- The table “Configuration” is used to set parsing configuration for all data types (hotel/tour/flight).
- Admin/Staff can specify a sample page on the website and pre-define all the html elements for every data property.
- Process to parse data with the following steps:
 - Crawl all the urls in the same domain as the given website
 - With each url, check if the html structure matches the defined configurations
 - If match, parse the specified data and update to database.
 - Mark the url as “done”

Flow chart



7.2. Parse data from Ajax

Definition

- Automatically parse all the appropriate data using the available ajax functions of a given website.
- Is mainly used for parsing flight information.

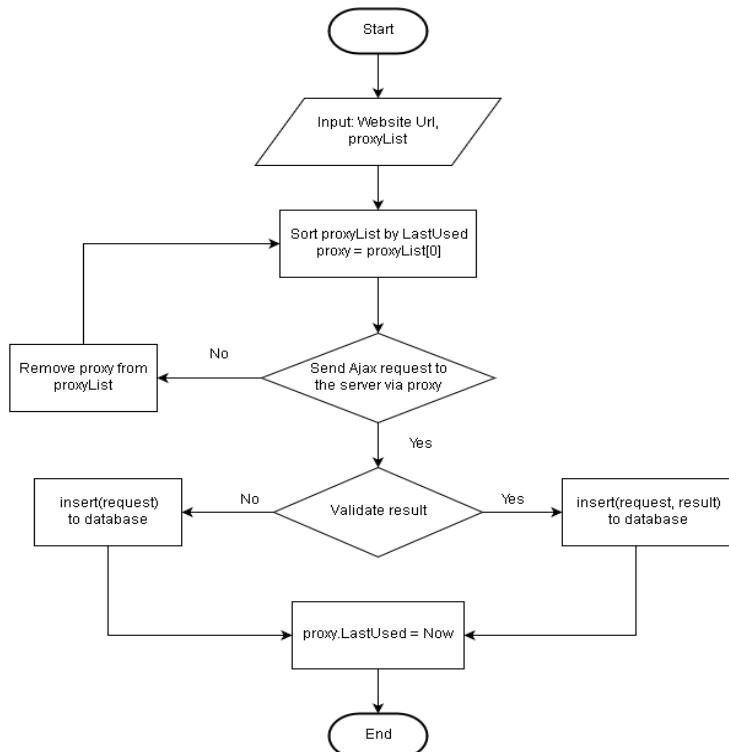
Problem Definition

- Too much requests can cause to IP address being banned by the website.

Solution

- Create a configuration file containing all the properties for the ajax function.
- To prevent IP address being banned by the website, a scheduled task is created to collect a list of available proxies and check if those proxies are dead or alive.
- Proxies are prioritized by Round Robin algorithm.
- Using those alive proxies:
 - Collect data of flight routes and airports
 - Send ajax request to get data of flight tickets
 - Parse data from ajax response and update to database

Flow chart



7.3. Remove low-valuable search keywords

Definition

- Remove low-valuable keyword from the keyword collection

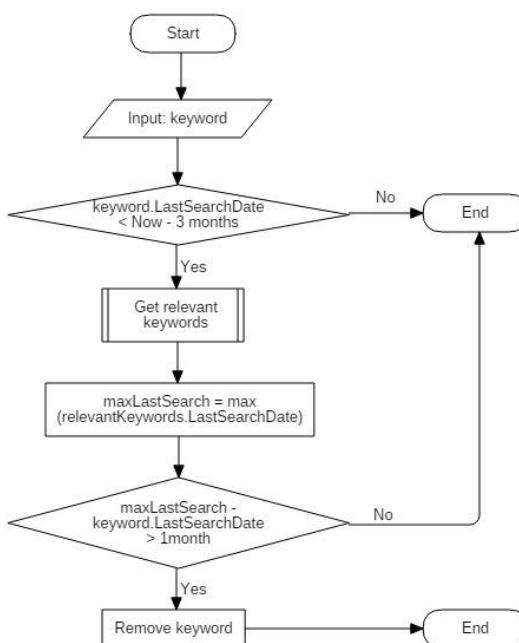
Problem definition

- Keyword collection must be updated periodically to keep usable keywords and remove low-valuable keywords.
- The recorded keywords are used for suggestion.

Solution

- All the keywords searched by users are recorded and stored to database.
- After a period of time, every keyword is checked if it will be removed. Keywords will be removed when meeting both of the conditions below:
 - Condition 1: Hasn't been search in over 3 months
(A keyword hasn't been search for such a long time may be considered as possibly useless. However, the last-search date can be affected by other factors such as seasons or trends.)
 - Condition 2: Max last search of relevant keywords – lastSearch of the given keyword > 1 month
(Compare to the others in the group, if the last-search date of the given keyword seems quite far away, the keyword can be considered as low-value. However, it takes time to make sure if a keyword is really low-value.)
- The process to determine the group of relevant keyword is mentioned in the section 7.5 Suggest relevant keywords.

Flow chart



7.4. Search data by keyword

Definition

- Retrieve a list of data corresponding to the inputted keyword.

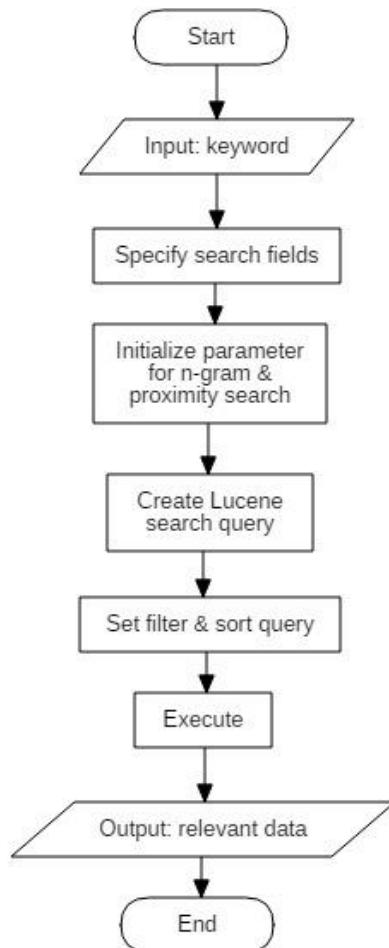
Problem definition

- The search result content must be relevant to the keyword.
- The search result must be prioritized by users' interest and optimized time by time.

Solution

- Each pair of a keyword and a data item must have a RelevantScore to be evaluated for search result.
- RelevantScore of keyword K and data item D is a combination of SimilarityScore and Click-through rate (CTR):
$$RelScore(K, D) = SimScore(K, D) * CTR(K, D)$$
- SimScore: SimScore is a measure of how well the keyword matches the content of the data item.
- CTR: CTR is the number of clicks that a result receives divided by the number of times it is shown expressed as a percentage.

Flow chart



- Initialize parameters for:
 - N-gram: specify acceptable matching percentage of two strings
 - Levenshtein distance: specify acceptable maximum distance of two strings
- Create Lucene query:
 - Customize scoring factors: TF(keyword, data), IDF(keyword, data), Coord(data)
 - Set boost value for important search terms
 - Add value of "SearchCount" field to scoring factors

7.5. Suggest relevant keyword

Definition

- Retrieve a list of relevant keywords to a the keyword searched by users.

Problem definition

- When user search a particular keyword, the system will suggest relevant keywords which are often searched by other users.

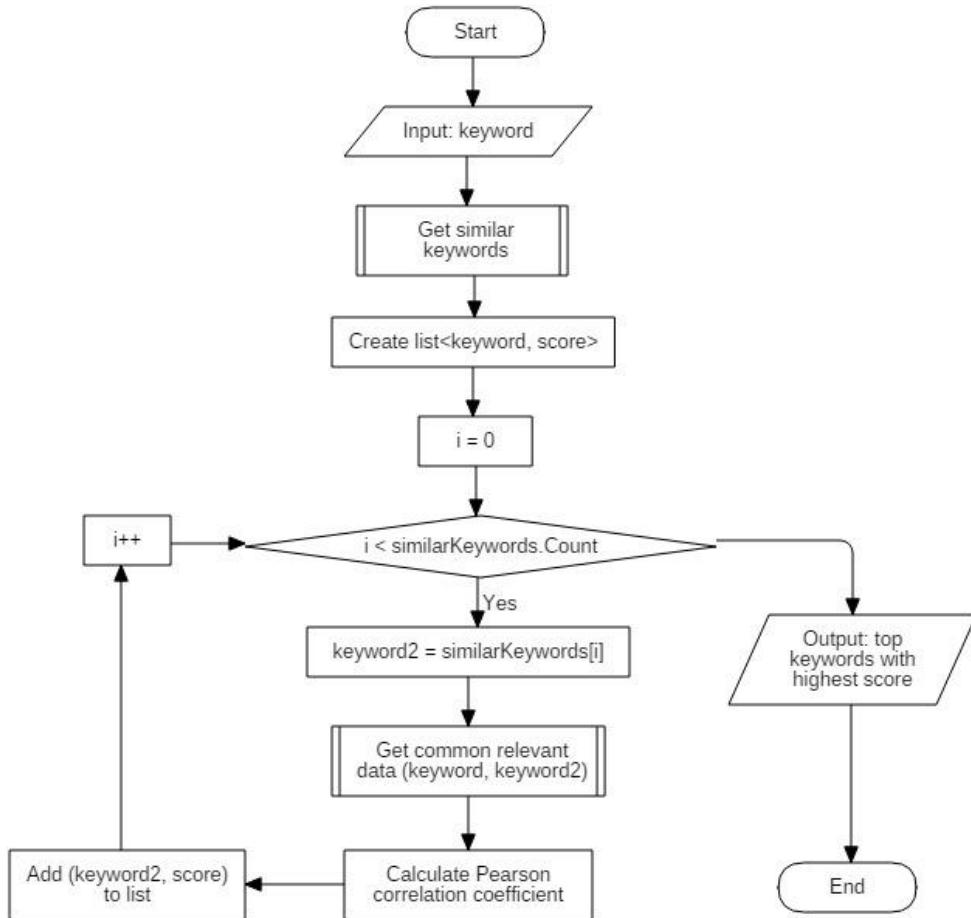
Solution: Pearson Correlation

- Giving two keywords K1 and K2, retrieve a list of common relevant data of both K1 and K2.
- Consider each data item d in data collection D as a point in the coordinate system with:
 - $x = \text{RelScore}(K1, d)$
 - $y = \text{RelScore}(K2, d)$
- The correlation coefficient is a measure of how well that set of data items fit on a straight line.
- The Pearson score of two keywords K1 and K2 is calculated as below:

$$\text{PearsonScore}(K1, K2) = \frac{\sum_{d \in D} (x * y) - \frac{\sum x * \sum y}{|D|}}{\sqrt{\left(\sum(x^2) - \frac{(\sum x)^2}{|D|}\right) * \left(\sum(y^2) - \frac{(\sum y)^2}{|D|}\right)}}$$

- PearsonScore is between -1 and 1, where 1 is total positive correlation, 0 is no correlation, and -1 is total negative correlation.
- The final results are those keywords in the top highest PearsonScore to the given keyword.

Flow chart



7.6. Suggest relevant data

Definition

- Retrieve a list of relevant data to a the hotel/tour/flight viewed by users.

Problem definition

- When user view a particular data item, the system will suggest the relevant data which are often viewed by other users.

Solution: Pearson Correlation

- Giving two data items D1 and D2, calculate PearsonScore(D1, D2) the same way as the keyword suggestion algorithm:
 - Consider each keyword as a point in the coordinate system.
 - D1 and D2 are the two axis.
- The final results are those data items in the top highest PearsonScore to the given data item.

Flow chart

(Similar as section 7.5 Suggest relevant keyword)

7.7. Suggest near-by hotels

Definition

- Retrieve a list of near-by hotels to the hotel viewed by users.

Problem definition

- Suggest hotels in the same area as the chosen hotel.

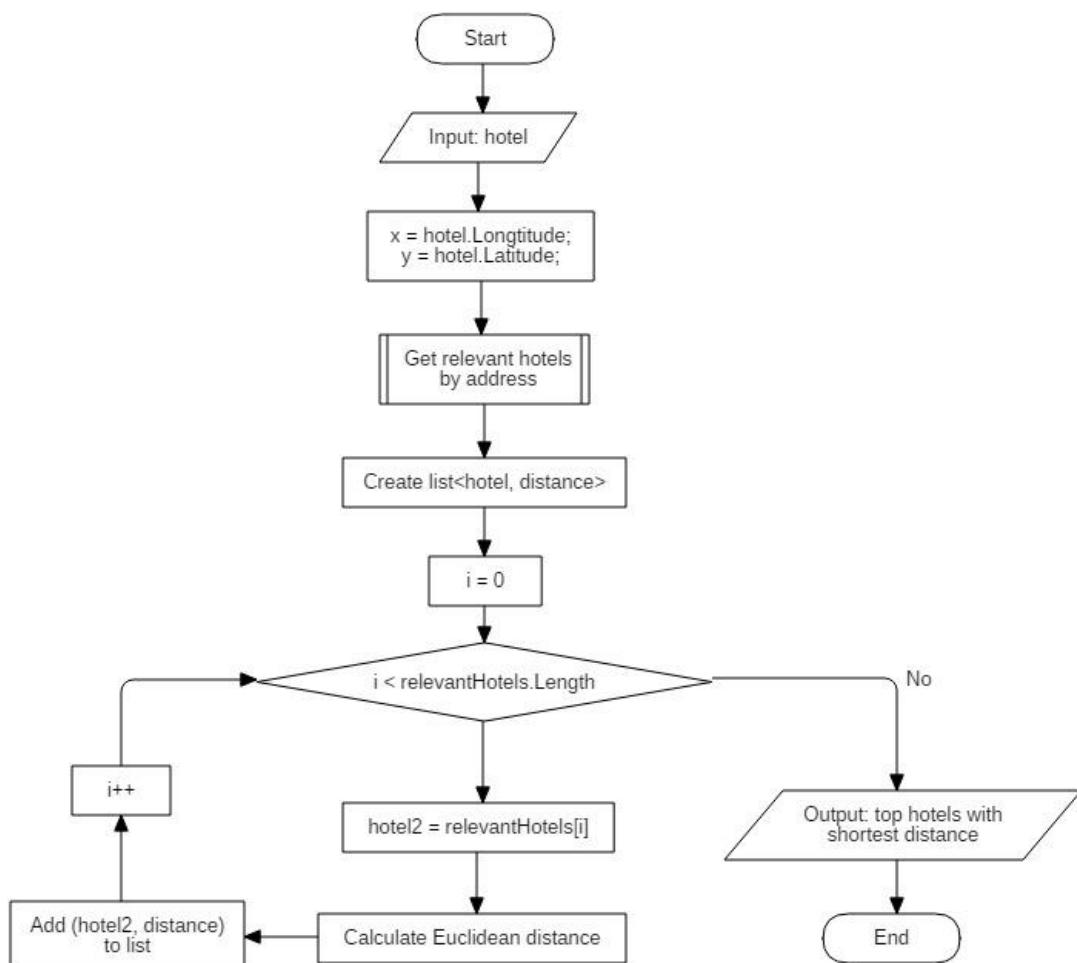
Solution: Euclidean distance

- Use Google map API to retrieve longitude and latitude based on the given address (perform when parsing data).
- Get list of relevant data items using RelScore with query terms only on field "Address".
- Retrieve longitude and latitude of the relevant data.
- Calculate Euclidean distance of two data item D1 and D2:

$$Distance(D1, D2) = \sqrt{(long1 - long2)^2 + (lat1 - lat2)^2}$$

- The final results are those data items in the top lowest distance to the given data item.

Flow chart



7.8. Categorize data by location

Definition

- Categorize hotels by location (city/province).

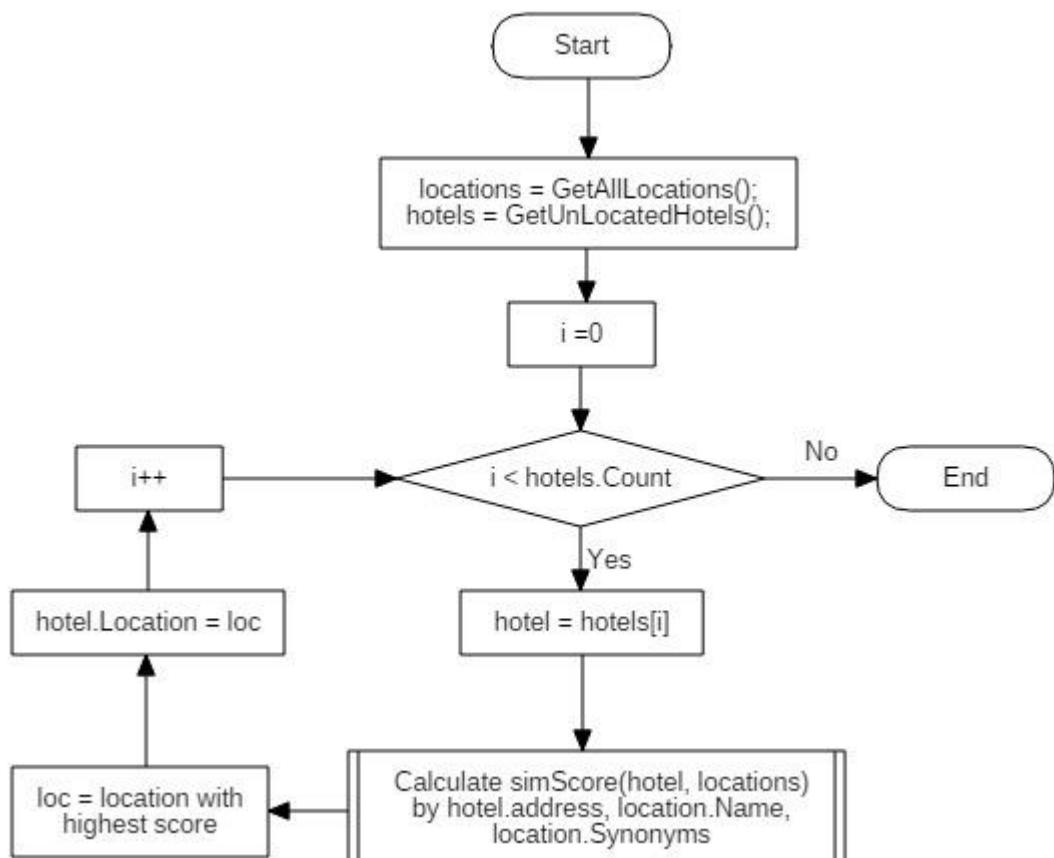
Problem definition

- Data parsed from different data sources has different format of address.
- Categorizing data by location is necessary to implement functions of filtering.

Solution

- Admin/Staff can create a standard list of locations. Each location has a official name and a list of synonyms.
- A scheduler is created to do the job:
 - Get a list of all locations.
 - Get a list of all “unlocated” data.
 - Calculate SimScore(location, data) by using official name and synonyms of location and “Address” field of data.
 - The data item will be categorized to the location with the highest SimScore.

Flow chart



7.9. Group hotel services

Definition

- Group similar hotel services.

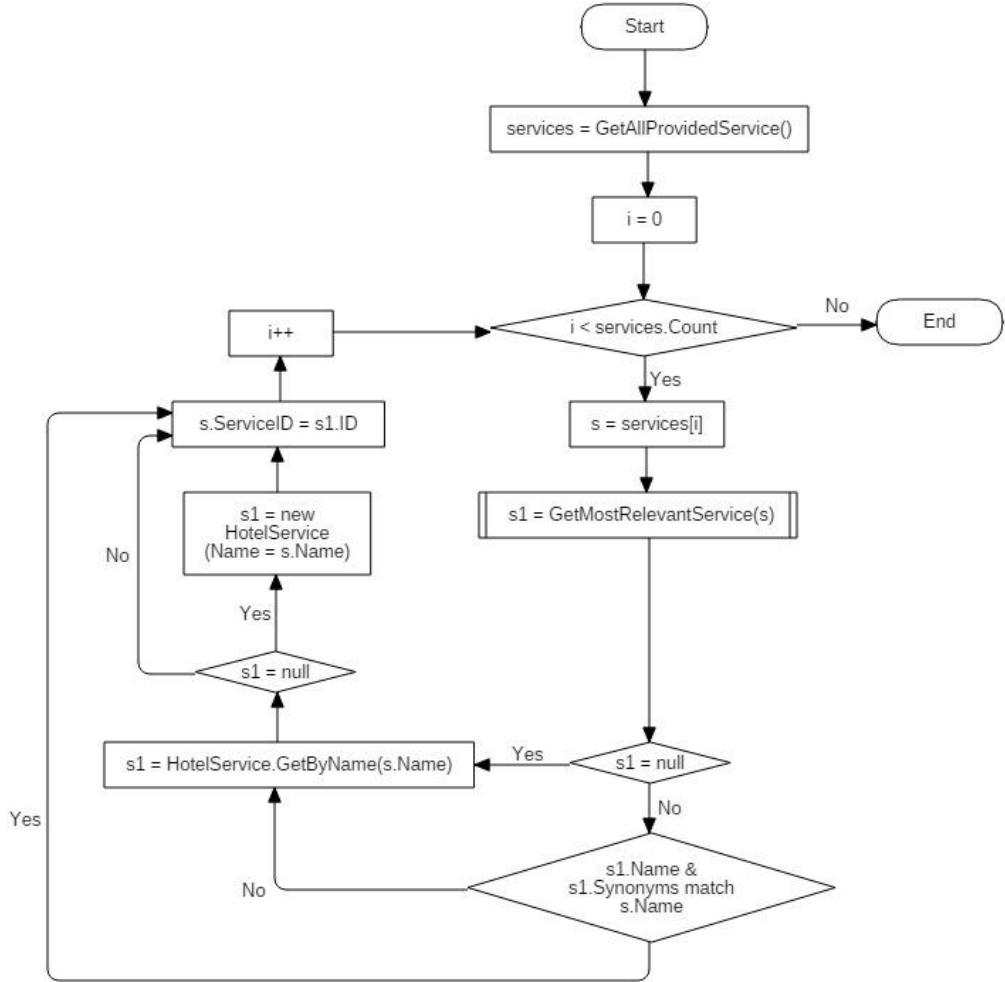
Problem definition

- Data parsed from different data sources has different format of hotel services.
- Grouping hotel services is necessary to implement functions of filtering.

Solution

- When parsing data, add provided services:
 - HotelID = hotel ID
 - ServiceName = parsed data
 - ServiceID = null
- A scheduler is created to group those raw services:
 - Get all uncategorized provided services, grouped by ServiceName.
 - For each service S, search for the approved service S1 with the highest simScore.
 - If cannot find any similar approved service, add new service as unapproved if there's no service with the same name as S.
 - Otherwise, if the service S1 has name & synonyms match name of S, group S to S1; else, add new add new service as unapproved if there's no service with the same name as S.
- Admin can edit, approve or remove those newly created services.

Flow chart



7.10. Updating data process

Definition

- Periodically check state of data in main storage and update data to Lucene server.

Problem definition

- Data being stored in main storage (Microsoft SQL Server) cannot perform efficient search with large data. Thus, it's necessary to update all the needed data in the main storage to a no-SQL storage (Lucene) to optimize the performance.

Solution

- 6 concurrent schedulers are created to run periodically to do the following jobs:
 - Scheduler ParseData: Create a thread for each enabled data source to perform parsing by the specified intervals and configurations.

- Scheduler UpdateData: Check the data states and update newly parsed data to Lucene.
- Scheduler RemoveRedundantData: Check and remove out-of-date flight data.
- Scheduler RemoveRedundantKeyword: Check the state and last-search-date of all the keywords and update keyword collections in Lucene.
- Scheduler CategorizeHotelByLocation: Categorize newly updated hotels by location using hotel addresses.
- Scheduler GroupHotelServices: Group hotel services and update service synonyms.

E. System Implementation & Test

1. Introduction

1.1. Overview

This section describes the approach and methodologies used by group to plan, organize and manage the testing of EZGO. It provides in the detail all necessary information about the implementation and testing procedure of the system included test plans, test cases, test result, test environments, pass/fail criteria and risks estimations as well as a checklist to cover all possible cases.

1.2. Test Approach

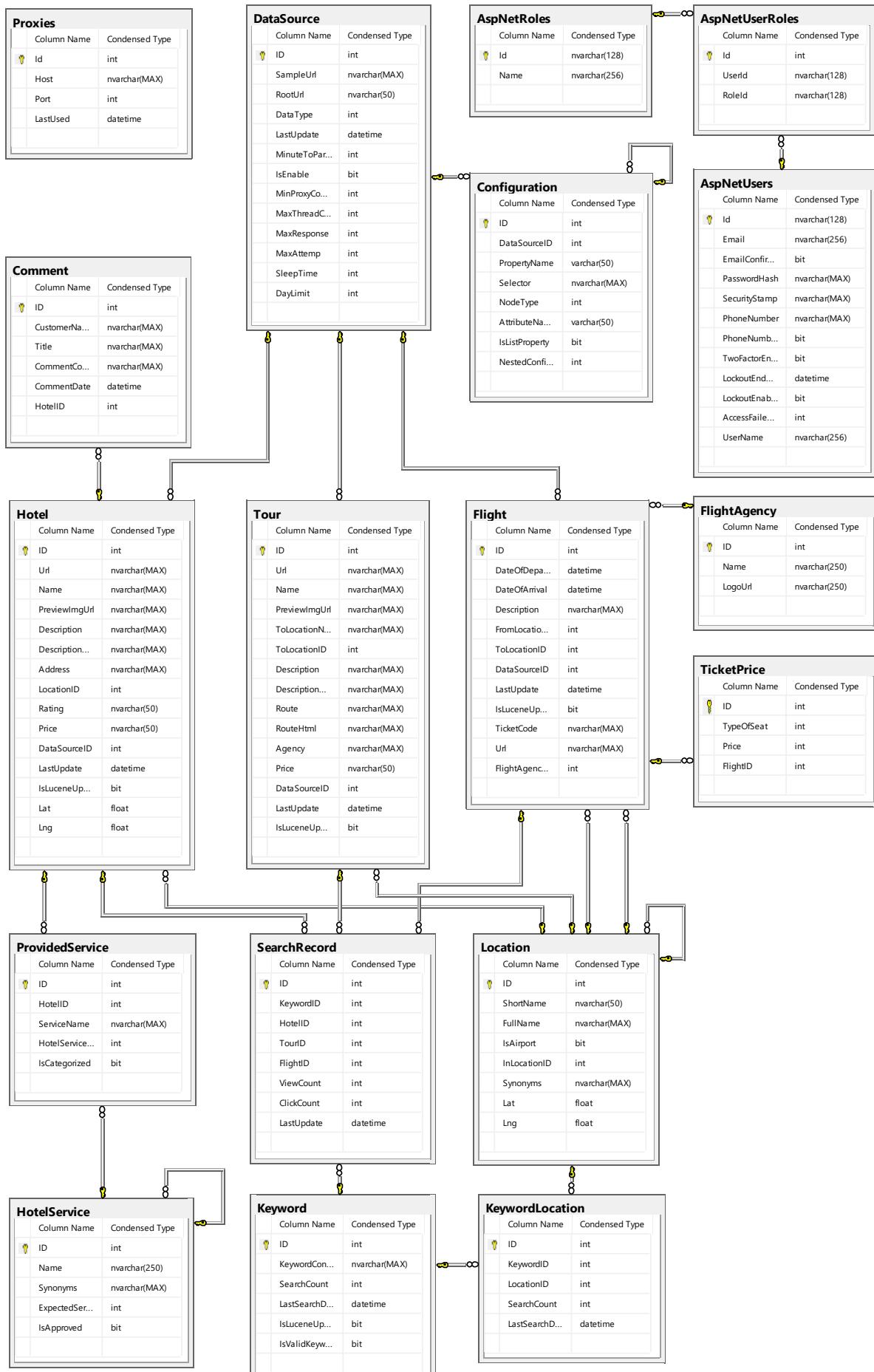
- **Goal:** Test all features in the whole system based on the core flow.
- **Method:** black-box testing
- **Technique:** check list

The testing for this project will consists of Integration System test level. Testing the program which was integrated and as a complete system to ensure that the software requirements have been met.

- Integration testing would be performed by all member of team and approved by team leader.
- System testing is focused on assessing the system's reliability. This process is concerned with finding errors that result from unanticipated interactions between components and component interface problems.

2. Database relationship diagram

2.1. Physical Diagram



2.2. Data Dictionary

Entity Name	Description
DataSource	Contains all the information about websites to be parsed.
Configuration	Specifies needed properties for parsing data.
Hotel	Contains all the properties about hotels.
Flight	Contains all the properties about tours.
Tour	Contains all the properties about flights.
Keyword	Recorded from search keywords of users.
SearchRecord	Used to update click count & view count from user behaviors.
Location	Used to categorize data by cities/provinces.
Comment	Parsed comments associated with hotels.
HotelService	Official hotel services used for filtering search results.
ProvidedService	Parsed services associated with hotels.
FlightAgency	Parsed agencies associated with hotels.
TicketPrice	Categorize flight data into 3 type of tickets: Adult, Children, Orphan
Proxy	Contains the list of proxies used to parse data without being banned by the data sources.
User	Store information of admin and staff of the system.
Role	Has 2 types: Admin, Staff

3. Test Plan

3.1. Feature to be tested

- Search by keywords
- Search hotels/tours/flights by criterias
- Sort & filter hotels/tours/flights
- Suggest: relevance keywords, relevance hotels/tours/flights, nearby hotels
- Configure data source parsing
- Schedulers: data parsing, data updating, data categorizing

3.2. Feature not to be tested

- View data detail
- Admin: CRUD operation

4. System Testing Test Case

4.1. Test case

1.2.1. <User> Search

ID	Test Case Description	Test case procedure	Expected output	Inter-test case dependence	Result	Test Date
TC01	Search all without key keyword	<ul style="list-style-type: none"> • Go to home page • Press “Search” button 	Show message “Vui long nhập từ khóa”		Pass	From 8/4/16 To 8/10/16
TC02	Search hotel without key keyword	<ul style="list-style-type: none"> • Go to home page • Press “Search” button 	Show message “Vui long nhập từ khóa”		Pass	From 8/4/16 To 8/10/16
TC03	Search tour without key keyword	<ul style="list-style-type: none"> • Go to home page • Press “Search” button 	Show message “Vui long nhập từ khóa”		Pass	From 8/4/16 To 8/10/16
TC04	Search round trip flight without date departure	<ul style="list-style-type: none"> • Go to home page • Press “Search” button 	Show message “Vui long chọn ngày đi”		Pass	From 8/4/16 To 8/10/16
TC05	Search round trip flight without date return	<ul style="list-style-type: none"> • Go to home page • Input date departure • Press “Search” button 	Show message “Vui long chọn ngày về”		Pass	From 8/4/16 To 8/10/16
TC06	Search round trip flight with date return less than date departure	<ul style="list-style-type: none"> • Go to home page • Input date departure value is “07/04/2016” • Input date return value is “06/04/2016” • Press “Search” button 	Show message “Ngày về không được nhỏ hơn ngày đi”		Pass	From 8/4/16 To 8/10/16

TC07	Search round trip flight with departure and arrival in duplicate.	<ul style="list-style-type: none"> • Go to home page • Input date departure value is “Hồ Chí Minh” • Input date return value is “Hồ Chí Minh” • Press “Search” button 	Show message “Điểm đi và điểm đến không được trùng nhau”		Pass	From 8/4/16 To 8/10/16
-------------	---	---	---	--	------	---------------------------

1.2.2. <User> Log in

ID	Test Case Description	Test case procedure	Expected output	Inter-test case dependence	Result	Test Date
TC08	Login	<ul style="list-style-type: none"> • Go to login page. • Input username “admin” and password “admin” • Press “Login” button. 	Show “Dashboard” page		Pass	From 8/4/16 to 8/4/16
TC09	Login with wrong username	<ul style="list-style-type: none"> • Go to login page. • Input username “aaa” and password “admin” • Press “Login” button. 	Show message “Tên đăng nhập hoặc mật khẩu không đúng”		Pass	From 8/4/16 to 8/4/16
TC10	Login with wrong password	<ul style="list-style-type: none"> • Go to login page. • Input username “admin” and password “123456” • Press “Login” button. 	Show message “Tên đăng nhập hoặc mật khẩu không đúng”		Pass	From 8/4/16 to 8/4/16

1.2.3.<User> Log out

ID	Test Case Description	Test case procedure	Expected output	Inter-test case dependence	Result	Test Date
TC11	Logout	<ul style="list-style-type: none"> • Guest opens menu on the right top page. • Patient clicks on “Logout” item. 	Show “Login” page		Pass	From 8/4/16 to 8/4/16

1.2.4.<Staff> Manage Hotels

ID	Test Case Description	Test case procedure	Expected output	Inter-test case dependence	Result	Test Date
TC13	Edit location hotel	<ul style="list-style-type: none"> • Go to “Manager Hotel” page. • Click on “Edit” item, show “Edit” popup • Select location • Press “Save changes” 	Show “Success” popup		Pass	From 8/4/16 to 8/4/16
TC14	Remove hotel	<ul style="list-style-type: none"> • Go to “Manager Hotel” page. • Select hotel must remove. • Click on “Remove” item. 	Show “Success” popup		Pass	From 8/4/16 to 8/4/16

1.2.5.<Staff> Manage Tours

ID	Test Case Description	Test case procedure	Expected output	Inter-test case dependence	Result	Test Date
TC15	Edit location tour	<ul style="list-style-type: none"> • Go to “Manager Tour” page. • Click on “Edit” item, show “Edit” popup • Select location • Press “Save changes” 	Show “Success” popup		Pass	From 8/4/16 to 8/4/16
TC16	Remove tour	<ul style="list-style-type: none"> • Go to “Manager Tour” page. • Select hotel must remove. • Click on “Remove” item. 	Show “Success” popup		Pass	From 8/4/16 to 8/4/16

1.2.6.<Staff> Manage Hotel Services

ID	Test Case Description	Test case procedure	Expected output	Inter-test case dependence	Result	Test Date
TC17	Edit hotel service synonyms	<ul style="list-style-type: none"> • Go to “Manager Hotel Service” page. • Select tab “Services” • Click on “Edit” item, show “Edit” popup • Input synonyms keyword • Press “Save changes” 	Show “Success” popup		Pass	From 8/4/16 to 8/4/16
TC18	Remove hotel service	<ul style="list-style-type: none"> • Go to “Manager Hotel Service” page. 	Show “Success” popup		Pass	From 8/4/16 to 8/4/16

		<ul style="list-style-type: none"> • Select tab “Services” • Select hotel service must remove. • Click on “Remove” item. 				
TC19	Approve new hotel service	<ul style="list-style-type: none"> • Go to “Manager Hotel Service” page. • Select tab “New Services” • Select hotel service must approve. • Press on “Approve” button. 	Show “Success” popup		Pass	From 8/4/16 to 8/4/16
TC20	Edit new hotel service	<ul style="list-style-type: none"> • Go to “Manager Hotel Service” page. • Select tab “New Services” • Click on “Edit” item, show “Edit” popup • Input synonyms keyword • Press “Save changes” 	Show “Success” popup		Pass	From 8/4/16 to 8/4/16
TC21	Remove new hotel service	<ul style="list-style-type: none"> • Go to “Manager Hotel Service” page. • Select tab “New Services” • Select hotel service must remove. • Click on “Remove” item. 	Show “Success” popup		Pass	From 8/4/16 to 8/4/16

F. Software User's Manual

1. Installation Guide

1.1. Setting up environment at server side

The specifications are based on the dependencies requirements and performance test result from previous section of this document.

1.1.1. Hardware requirement

Hardware	Specification
Internet Connection	8 Mbps
Computer Processor	Intel® Core(TM) i5 CPU , M 460 @ 2.53GHz
Computer Memory	4GB of RAM or more
Hard Disk Drive	40GB or more

Table 12: Installation Guide - Hardware Requirement

1.1.2. Software requirement

Software	Application name / version
Operating System	Window Server 2012
.Net	4.5
DBMS	SQL Server 2014

Table 13: Installation Guide - Software Requirement

1.2. Web Application Deployment Process

1.2.1. Check environment

Check .Net Framework version.

Open command prompt and enter: dir %WINDIR%\Microsoft.Net\Framework\v*
Output:

Directory of C:\Windows\Microsoft.Net\Framework

```
11/21/2010 10:31 AM <DIR> v1.0.3705
07/14/2009 10:20 AM <DIR> v1.1.4322
04/10/2016 06:27 PM <DIR> v2.0.50727
11/21/2010 02:06 PM <DIR> v3.0
03/28/2016 02:48 AM <DIR> v3.5
04/17/2016 12:18 PM <DIR> v4.0.30319
```

Note: .NET Framework 4.5 is stored in a sub-directory of 4.0.

1.2.2. Import Database

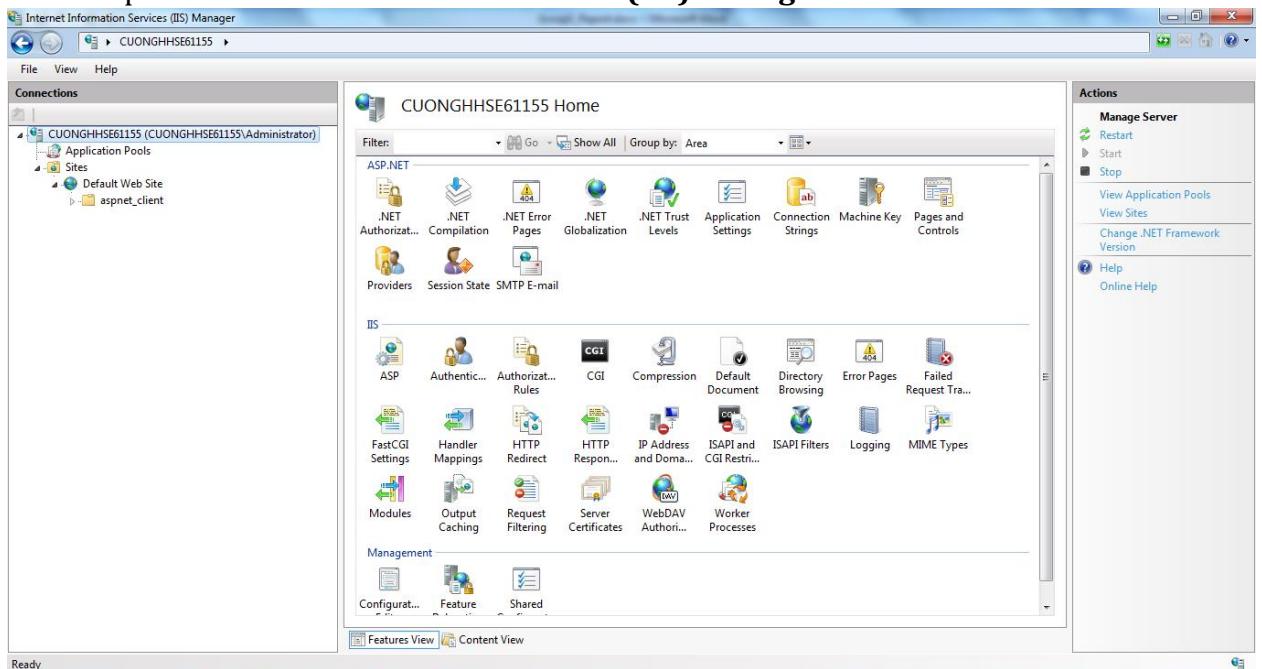
Using file database.sql located under Deployment directory from this document.

1.2.3. Build project

1. Open **SearchEngine Solution (SearchEngine.sln)** in Visual Studio 2013
2. Right-click on **SearchEngine Solution**, select “Build”

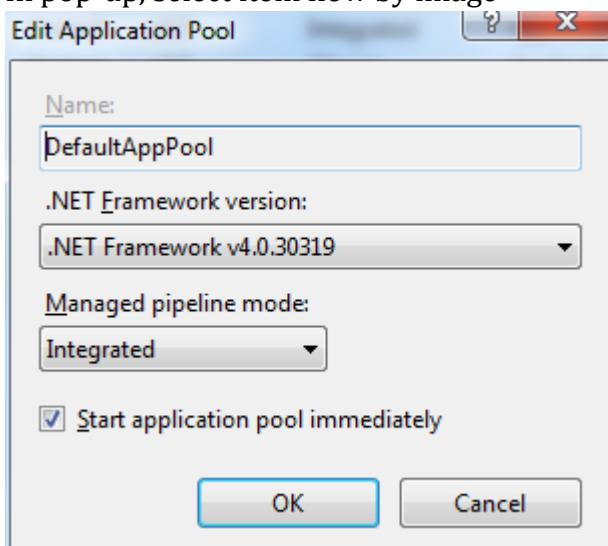
1.2.4. Deploy project to IIS

1. Open source code folder, copy **SearchEngine** folder to another disk.
2. Open **Internet Information Services (IIS) Manager**



3. Config application Pool to .Net Framework v4.0.30319

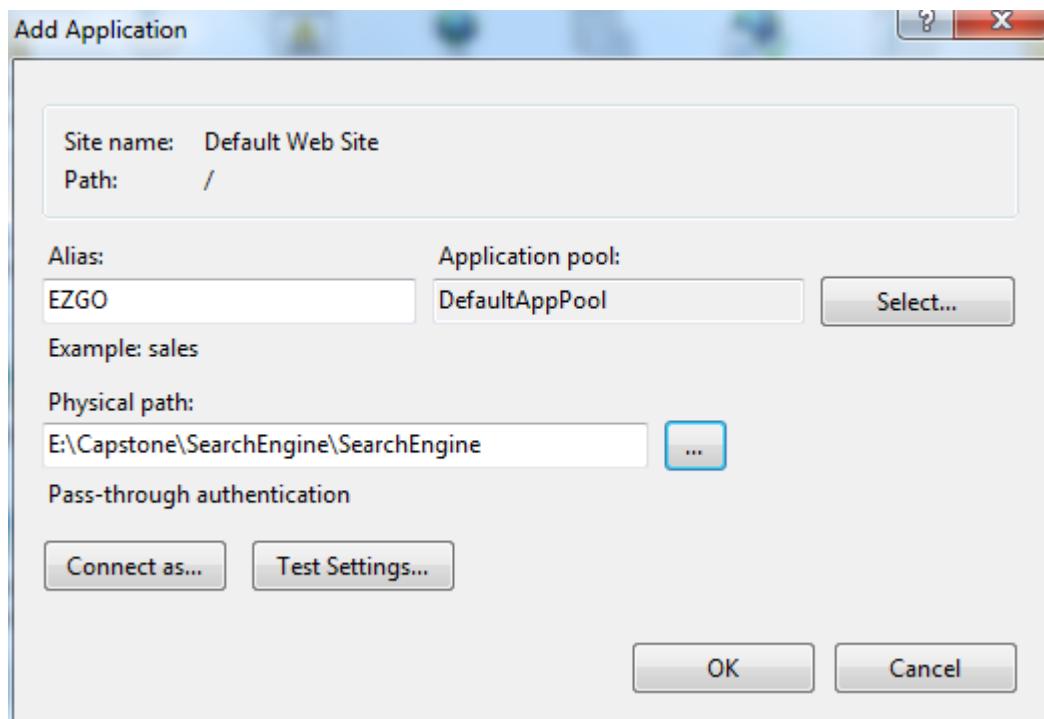
- Select “**Application Pool**”
- Double click on “**DefaultAppPool**”
- In pop-up, select item flow by image



- Click “OK”

4. Deploy project to IIS

- Right click on “**Default Web Site**”, select “**Add Application**”
- In pop-up input flow by image



Note: - Physical path: located direction of SearchEngine.Web project that copy on step 1.

- Alias: Name of web after deploy to IIS, can be free input.

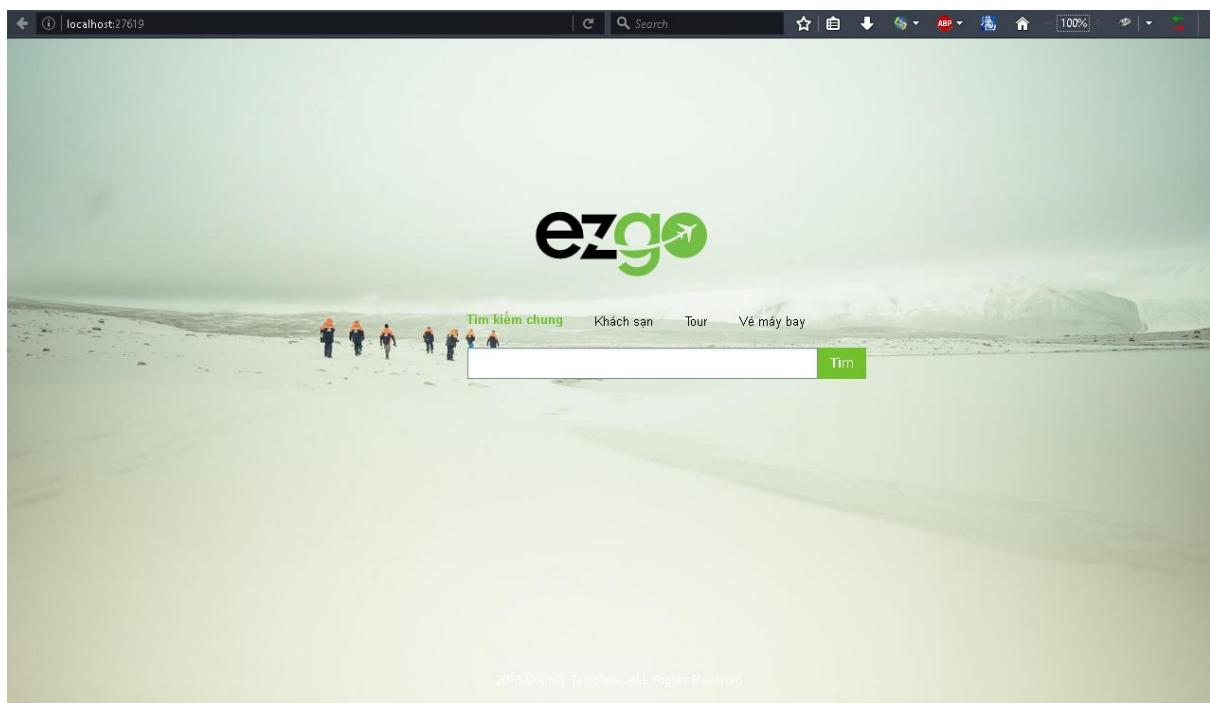
5. Click “OK”

2. User Guide

2.1. Web application

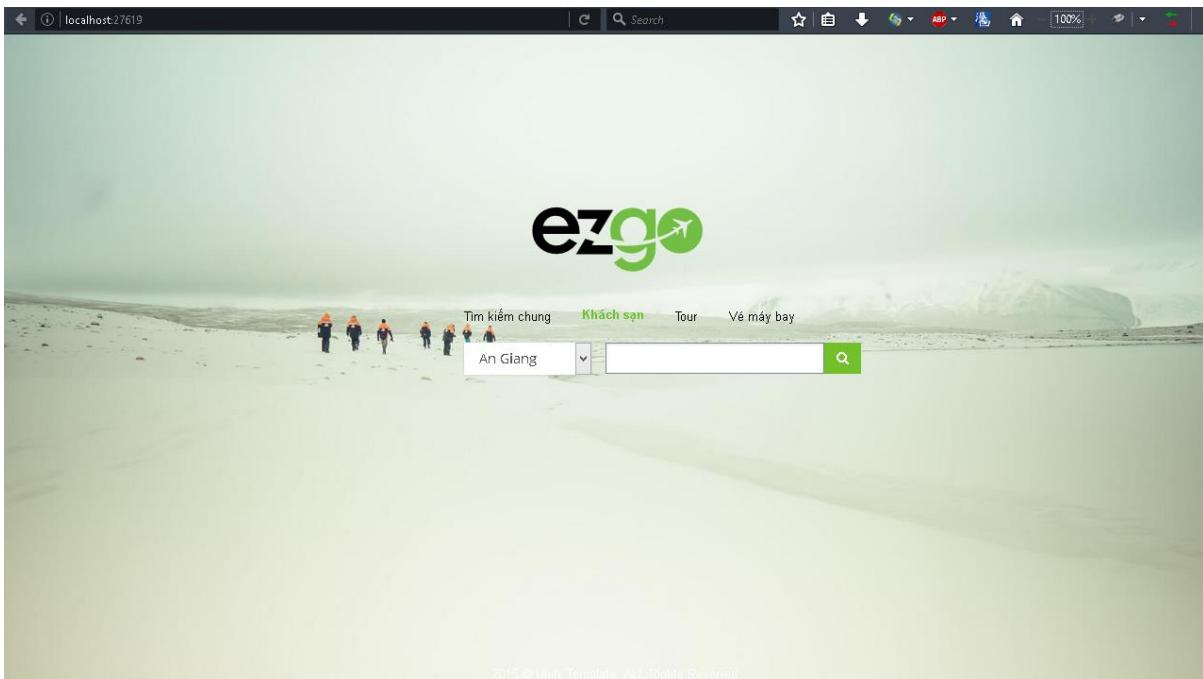
2.1.1. User

2.1.1.1. General search



Step	Description
1	Click on “Tim kiêm chung”.
2	Fill in field: Searching keywords.
3	Click “Tìm” button.

2.1.1.2. Search for hotels



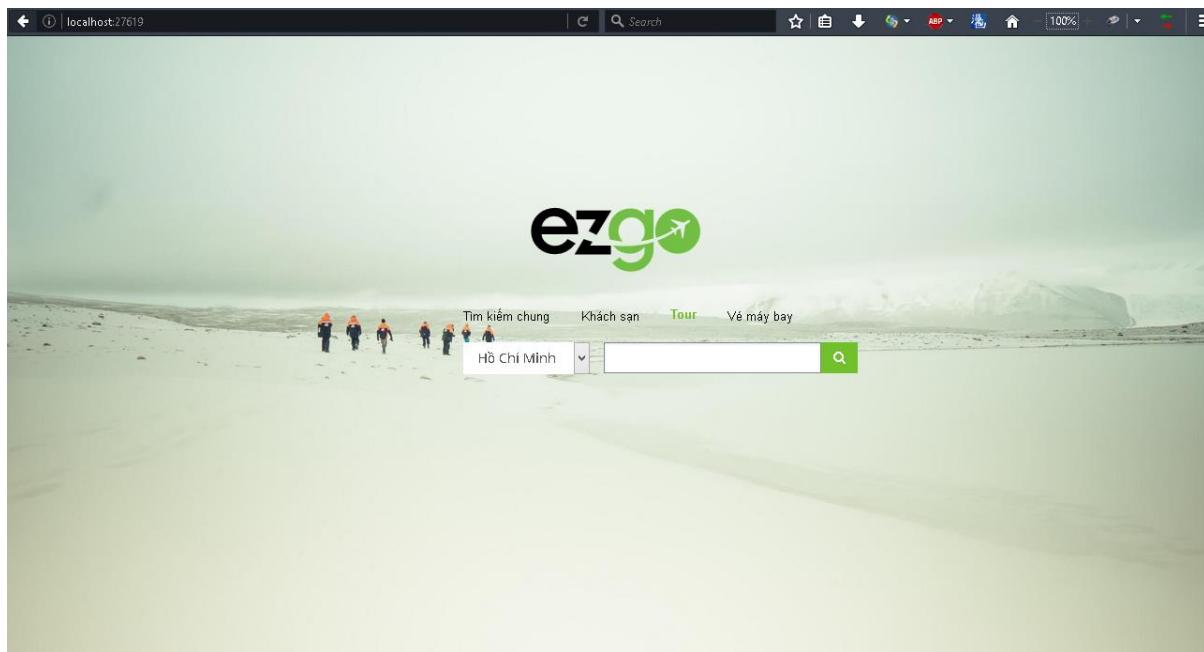
Step	Description
1	Click on “Khách sạn”.
2	Drop down list: List of provinces. Fill in field: Searching keywords.
3	Click on button with magnitude glass icon.

The screenshot shows the search results for "ho chi minh" with type=0 and location=Ho Chi Minh City. The results are filtered by province ("An Giang") and sorted by price (0.0).

- Result 1:** Khách sạn A25 Sunrise Sài Gòn (highlighted with a red box)
- Result 2:** Khách sạn Đại Nam Sài Gòn (highlighted with a red box)
- Result 3:** Khách sạn Đệ Nhất (highlighted with a red box)
- Filter Options (Left Side):**
 - 2)** Theo mức giá: Dưới 500.000, 500.000 - 1.000.000, 1.000.000 - 1.500.000, 1.500.000 - 2.000.000, 2.000.000 - 2.500.000, Trên 2.500.000.
 - 3)** Theo dịch vụ khách sạn: Bữa sáng miễn phí, Bãi đỗ xe, Bể bơi ngoài trời, Giặt là, Hỗ trợ đặt tour, Internet, Massage, Nhà hàng, Quán cafe, Tâm hợi.
 - 4)** Sắp xếp: Giá tăng dần, Giá giảm dần, Đánh giá tăng dần.
- Right Side (Additional Information):**
 - Tour du lịch Tây Ninh - địa đạo Củ Chi trong 1 ngày
 - Khách Sạn Sài Gòn - Ninh Chữ
 - Khách sạn Sài Gòn Ninh Chữ
 - Khách sạn Bình Minh
 - Văn Minh Resort **5)**

Step	Description
4	Click on “Khách sạn lân cận” to show nearby hotels. (2) Filter results by price. (3) Filter results by hotel services. (4) Sort results. (5) Show relevant results.
5	Click on (1) to go to the detail page.

2.1.1.3. Search for tours



Step	Description
1	Click on “Tour”.
2	Drop down list: List of provinces. Fill in field: Searching keywords.
3	Click on button with magnitude glass icon.

Step	Description
4	(2) Filter results by price. (3) Sort results. (5) Show relevant results.
5	Click on (1) to go to the detail page.

2.1.1.4. Search for flight tickets

Step	Description
1	Click on “Vé máy bay”.
2	<p>Radio buttons:</p> <ul style="list-style-type: none"> - “Khứ hồi”: Round-trip ticket. - “Một chiều”: One-way ticket. <p>Fill in fields:</p> <ol style="list-style-type: none"> 1) Date of departure. 2) Date of arrival, 3) Departure location. 4) Arrival location.
3	Click on button with magnitude glass icon.

Chuyến bay	Giờ bay - Giờ đến	Giá vé	Xem chi tiết
Hà Nội - Hồ Chí Minh (VN7241)	19:40 - 21:50	959,000	Xem chi tiết
Hà Nội - Hồ Chí Minh (VN0271)	20:05 - 22:15	959,000	Xem chi tiết
Hà Nội - Hồ Chí Minh (VN7255)	20:25 - 22:35	959,000	Xem chi tiết

Chuyến bay	Giờ bay - Giờ đến	Giá vé	Xem chi tiết
Hà Nội - Hồ Chí Minh (BL0781)	06:30 - 09:35	839,000	Xem chi tiết
Hà Nội - Hồ Chí Minh (BL0791)	08:40 - 10:45	839,000	Xem chi tiết
Hà Nội - Hồ Chí Minh (BL0801)	16:00 - 18:05	839,000	Xem chi tiết
Hà Nội - Hồ Chí Minh (BL0797)	17:25 - 19:30	839,000	Xem chi tiết

Step	Description
4	<p>(1) Filter results by company.</p> <p>(2) Filter results by date of departure/date of arrival.</p> <p>(3) Sort results.</p>
5	Click on “Xem chi tiết” button.

2.1.2. Staff

2.1.2.1. Manage Data sources

2.1.2.1.1. Add new data source

#	Name	Type	Interval	Last update	Auto-run	Actions
1	http://wwwabay.vn	Flight	5 minutes	26/01/2016	<input checked="" type="checkbox"/>	<button>CONFIG</button> <button>FORCE START</button> <a>Edit / Remove
2	http://mytourvn	Hotel	120 minutes	08/04/2016	<input checked="" type="checkbox"/>	<button>CONFIG</button> <button>FORCE START</button> <a>Edit / Remove
3	http://mytourvn/tour	Tour	80 minutes	14/04/2016	<input checked="" type="checkbox"/>	<button>CONFIG</button> <button>FORCE START</button> <a>Edit / Remove

Step	Description
1	Click on “Data sources”.
2	Fill in fields: “Data type”: Type of data this site offered. “Root URL”. URL root of the site. “Sample URL”. Sample url contains the sample structure.
3	Click on “+” button.

2.1.2.1.2. Modify existing data sources

The screenshot shows the EZGO administration interface for managing data sources. On the left is a sidebar with links to Dashboard, Data sources, Hotel service, Flight agency, Location, Manager Hotel, Manager Tour, and Scheduler. The main area is titled "Data Source" and displays a table of configured data sources. The table columns include #, Name, Type, Interval, Last update, Auto-run, CONFIG, and Edit / Remove. Three entries are listed:

#	Name	Type	Interval	Last update	Auto-run	CONFIG	Edit / Remove
1	http://www.abay.vn	Flight	5 minutes	26/01/2016	<input checked="" type="checkbox"/>	<button>CONFIG</button>	<button>Edit / Remove</button>
2	http://mytourvn	Hotel	120 minutes	08/04/2016	<input checked="" type="checkbox"/>	<button>CONFIG</button>	<button>Edit / Remove</button>
3	http://mytourvn/tour	Tour	30 minutes	14/04/2016	<input checked="" type="checkbox"/>	<button>CONFIG</button>	<button>Edit / Remove</button>

Step	Description
1	Click on “Data sources”.
2	Click on “Edit”.

The screenshot shows the "Edit data source" dialog box overlaid on the main data source list. The dialog has fields for "Data type" (set to "Flight") and "Data source URL" (set to "http://www.abay.vn/"). At the bottom right of the dialog are "CLOSE" and "SAVE CHANGES" buttons. The background shows the same data source list as the previous screenshot.

Step	Description
3	Fill in fields: “Data type”: Type of data this site offered. “Root URL”. URL root of the site.
4	Click on “Save changes” button.

2.1.2.1.3. Configure existing data source of type Hotel / Tour

The screenshot shows the EZGO software's Data Source configuration screen. The left sidebar includes links for Dashboard, Data sources, Hotel service, Flight agency, Location, Manager Hotel, Manager Tour, and Scheduler. The main area is titled "Data Source" and shows a table of data sources. The table columns are #, Name, Type, Interval, Last update, and Auto-run (with a toggle switch). The first entry is for a Flight service with a 5-minute interval. The second entry is for a Hotel service with a 120-minute interval. The third entry is for a Tour service with a 30-minute interval. Each row has "CONFIG" and "FORCE START" buttons.

#	Name	Type	Interval	Last update	Auto-run	Actions
1	http://wwwabay.vn	Flight	5 minutes	26/01/2016	<input checked="" type="checkbox"/>	CONFIG FORCE START
2	http://mytourvn	Hotel	120 minutes	08/04/2016	<input checked="" type="checkbox"/>	CONFIG FORCE START
3	http://mytourvn/tour	Tour	30 minutes	14/04/2016	<input checked="" type="checkbox"/>	CONFIG FORCE START

Step	Description
1	Click on “Data sources”.
2	Click on “Config” button.

Step	Description
3	Click on “Load” button to reload sample page to (2).
4	Select an attribute from (1). Double-click to an item in (2) to copy it’s query string to (1).
5	Click on “Save changes” button.

2.1.2.1.4. Configure existing data source of type Flight

Step	Description
1	Click on “Data sources”.
2	Click on “Config” button.

Step	Description
3	<p>Fill in fields:</p> <ul style="list-style-type: none"> - “Data source”: URL root of the site. - “MinProxyCount”: Minimum number of proxies require to parse this site. - “MaxThreadCount”: Number of threads. - “MaxResponse”: Maximum response time for a connection. - “MaxAttemp”: Maximum number of attempt to connect to the site per proxy.
4	Click on “Save changes” button.

2.1.2.1.5. Execute data sources

The screenshot shows the 'Data Source' section of the EzGo Admin interface. On the left is a sidebar with links: Dashboard, Data sources (selected), Hotel service, Flight agency, Location, Manager Hotel, Manager Tour, and Scheduler. The main area has a 'Data type' dropdown set to 'Hotel', a 'Root URL' input field, and a 'Sample URL' button. Below is a table with three entries:

#	Name	Type	Interval	Last update	Auto-run	Actions
1	http://wwwabay.vn	Flight	5 minutes	26/01/2016	<input checked="" type="checkbox"/>	<button>CONFIG</button> <button>Edit / Remove</button>
2	http://mytourvn	Hotel	120 minutes	08/04/2016	<input checked="" type="checkbox"/>	<button>CONFIG</button> <button>Edit / Remove</button>
3	http://mytourvn/tour	Tour	30 minutes	14/04/2016	<input checked="" type="checkbox"/>	<button>CONFIG</button> <button>Edit / Remove</button>

Step	Description
1	Click on “Data sources”.
2	Click on “Force start” button to make the scheduler runs 1 time. Click on “Auto-run” switch to make scheduler runs periodically.

2.1.2.2. Manage Hotel Services

The screenshot shows the 'Hotel Service' section of the EzGo Admin interface. On the left is a sidebar with links: Dashboard, Data sources, Hotel service (selected), Flight agency, Location, Manager Hotel, Manager Tour, and Scheduler. The main area has tabs for 'SERVICES' (selected) and 'NEW SERVICES'. It includes fields for 'Name' and 'Synonyms'. Below is a table with three entries:

#	Name	Synonyms	Actions
1	Bữa sáng miễn phí	Bữa sáng miễn phí	<button>Edit / Remove</button>
2	Bãi đỗ xe	Có đỗ xe Có bãi Giữ xe miễn phí Bãi đậu xe ô tô tại khách sạn	<button>Edit / Remove</button>
3	Bể bơi ngoài trời	Hồ bơi	<button>Edit / Remove</button>

Step	Description
1	Click on “Hotel Services”.
2	<p>Click on “+” button to add new service with “Name” and “Synonyms”.</p> <p>Click on “Edit” to edit services.</p> <p>Click on “Remove” to remove services.</p> <p>Click on “New services” to show list of unapproved services.</p>

2.1.2.3. Manage Flight agency

#	Name	Logo	Action
1	Vietnam Airlines	[Logo Placeholder]	Edit / Remove
2	VietJet	[Logo Placeholder]	Edit / Remove
3	Jetstar Pacific	[Logo Placeholder]	Edit / Remove

Step	Description
1	Click on “Flight agency”.
2	<p>Click on “Add new” button to add new agency.</p> <p>Click on “Edit” to edit agency.</p> <p>Click on “Remove” to remove agency.</p>

2.1.2.4. Manage Location

#	ShortName	FullName	Synonyms	Action
1	AG	An Giang	AG	Edit / Remove
2	VT	Bà Rịa Vũng Tàu	VT;Vũng Tàu;Đảo	Edit / Remove
3	BG	Bắc Giang	Dont have data	Edit / Remove
4	Dont have data	Bắc Kạn	Dont have data	Edit / Remove
5	Dont have data	Bắc Liêu	Dont have data	Edit / Remove
6	Dont have data	Bắc Ninh	Dont have data	Edit / Remove

Step	Description
1	Click on “Location”.
2	Click on “Add new” button to add new location. Click on “Edit” to edit location. Click on “Remove” to remove location.

2.1.2.5. Manage schedulers

Update data

Status: Completed START NOW STOP

▼ View log

```
_LuceneJob - Flight - thread 3: added: 145997 - Sân bay quốc tế Nội Bài, Hà Nội - Sân bay quốc tế Tân Sơn Nhất, Hồ Chí Minh : 09/22/2016
_LuceneJob - Flight - thread 3: added: 145998 - Sân bay quốc tế Nội Bài, Hà Nội - Sân bay quốc tế Tân Sơn Nhất, Hồ Chí Minh : 09/22/2016
_LuceneJob - Flight: finished.
```

Categorize data by location

Status: Completed START NOW STOP

▼ View log

```
_LocationCategorizingJob - Tour: Done
_LocationCategorizingJob - Hotel: started
_LocationCategorizingJob - Hotel: started
_LocationCategorizingJob - Hotel: Done
_LocationCategorizingJob - Tour: started.
_LocationCategorizingJob - Tour: Done
```

Remove low-value keywords

Status: Running START NOW STOP

▼ View log

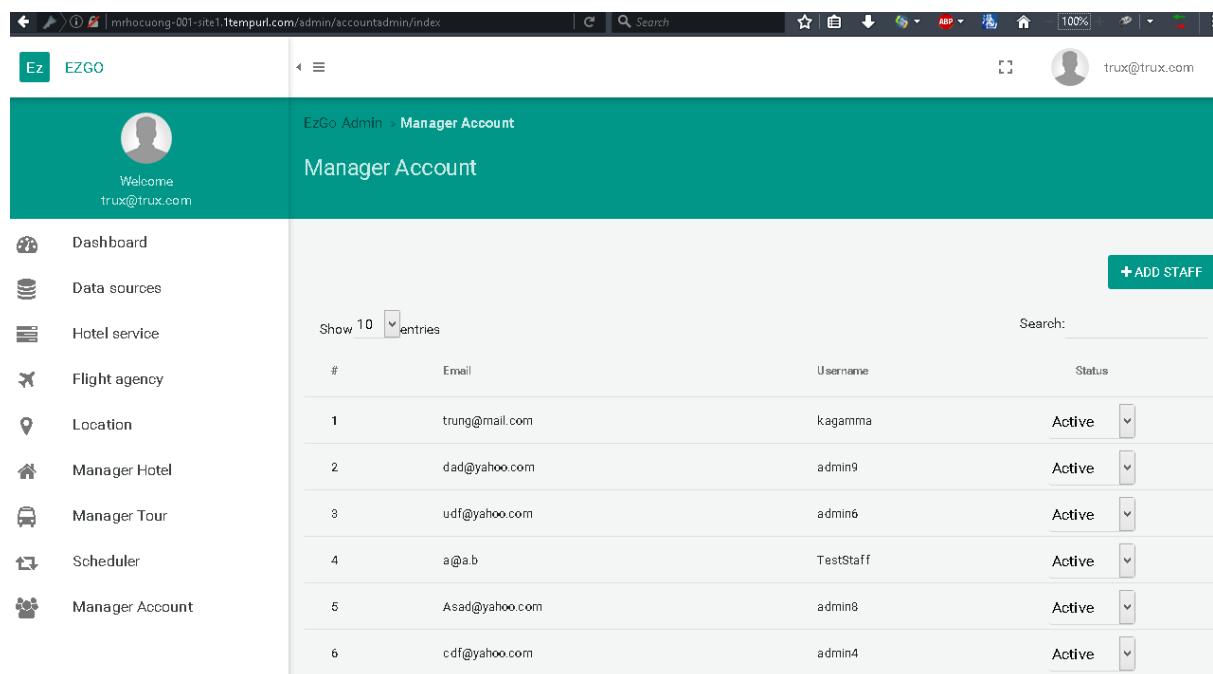
Categorize hotel services

Status: Completed START NOW STOP

Step	Description
1	Click on “Scheduler”.
2	Click on “Start now” to start a scheduler. Click on “Stop” to stop a scheduler.

2.1.3. Admin

2.1.3.1. Add new staff

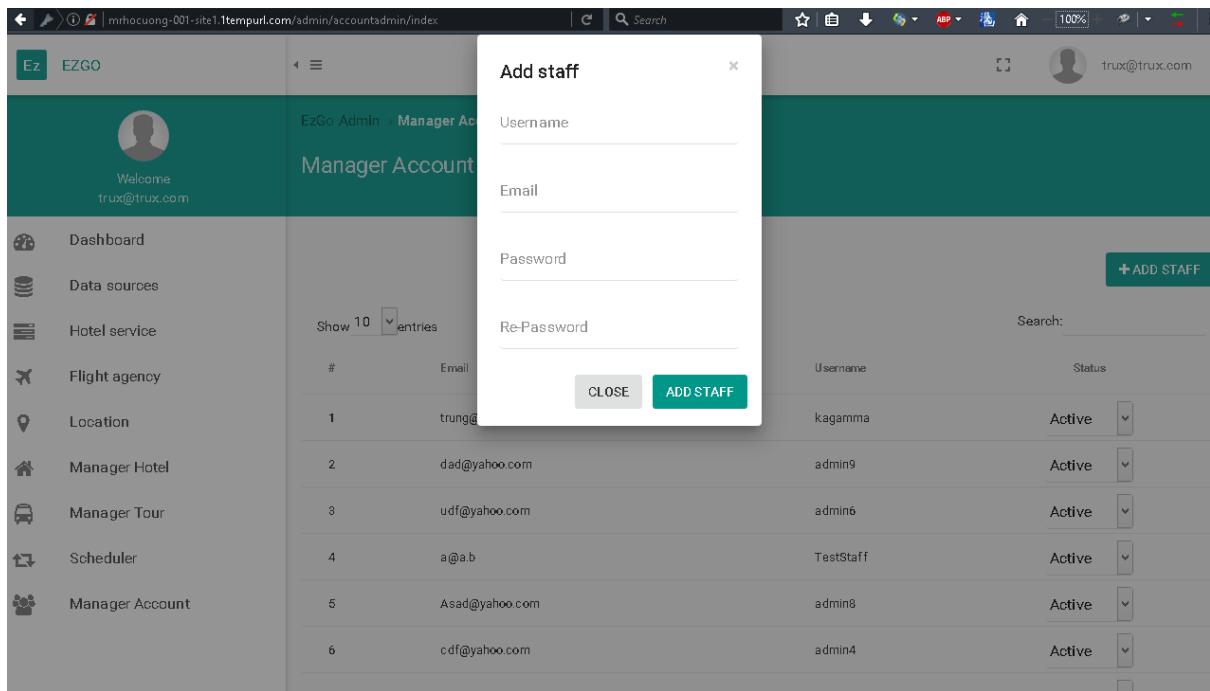


The screenshot shows a web-based administration interface for 'EzGo Admin Manager Account'. On the left, there is a sidebar with various menu items: Dashboard, Data sources, Hotel service, Flight agency, Location, Manager Hotel, Manager Tour, Scheduler, and Manager Account. The 'Manager Account' item is currently selected. The main content area displays a table titled 'Manager Account' with the following data:

#	Email	Username	Status
1	trung@mail.com	kagamma	Active
2	dad@yahoo.com	admin9	Active
3	udf@yahoo.com	admin6	Active
4	a@a.b	TestStaff	Active
5	Asad@yahoo.com	admin8	Active
6	cdf@yahoo.com	admin4	Active

At the top right of the main content area, there is a green button labeled '+ ADD STAFF'. The browser's address bar shows the URL: 'mrhocuong-001-site1临时.com/admin/account/admin/index'.

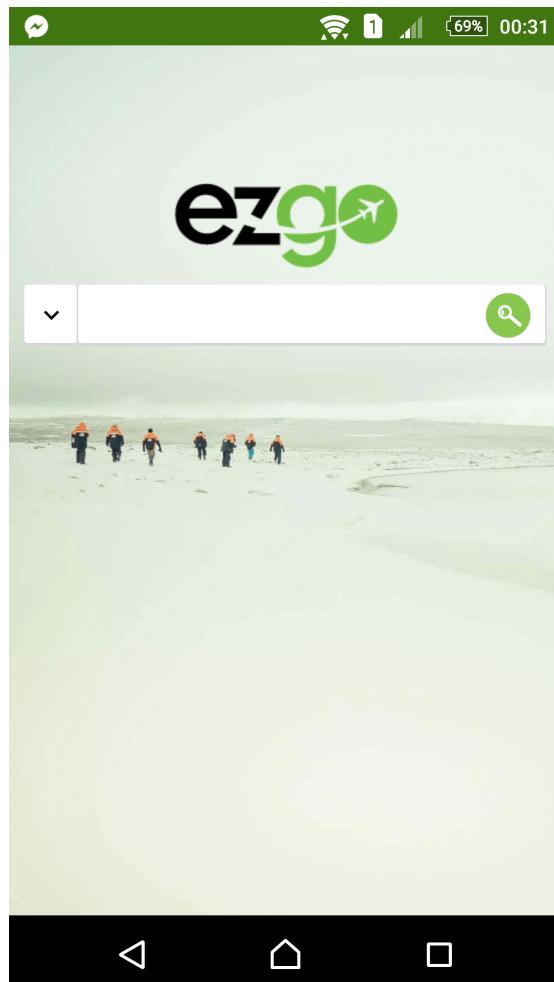
Step	Description
1	Click on “Manager Account”.
2	Click on “Add staff” button.



Step	Description
3	<p>Fill in fields:</p> <p>“Username”: Staff’s username. Must be at least 6 characters.</p> <p>“Email”: Staff’s email.</p> <p>“Password”: Staff’s password.</p> <p>“Re-Password”: Confirm password.</p>
4	Click on “Add staff” button.

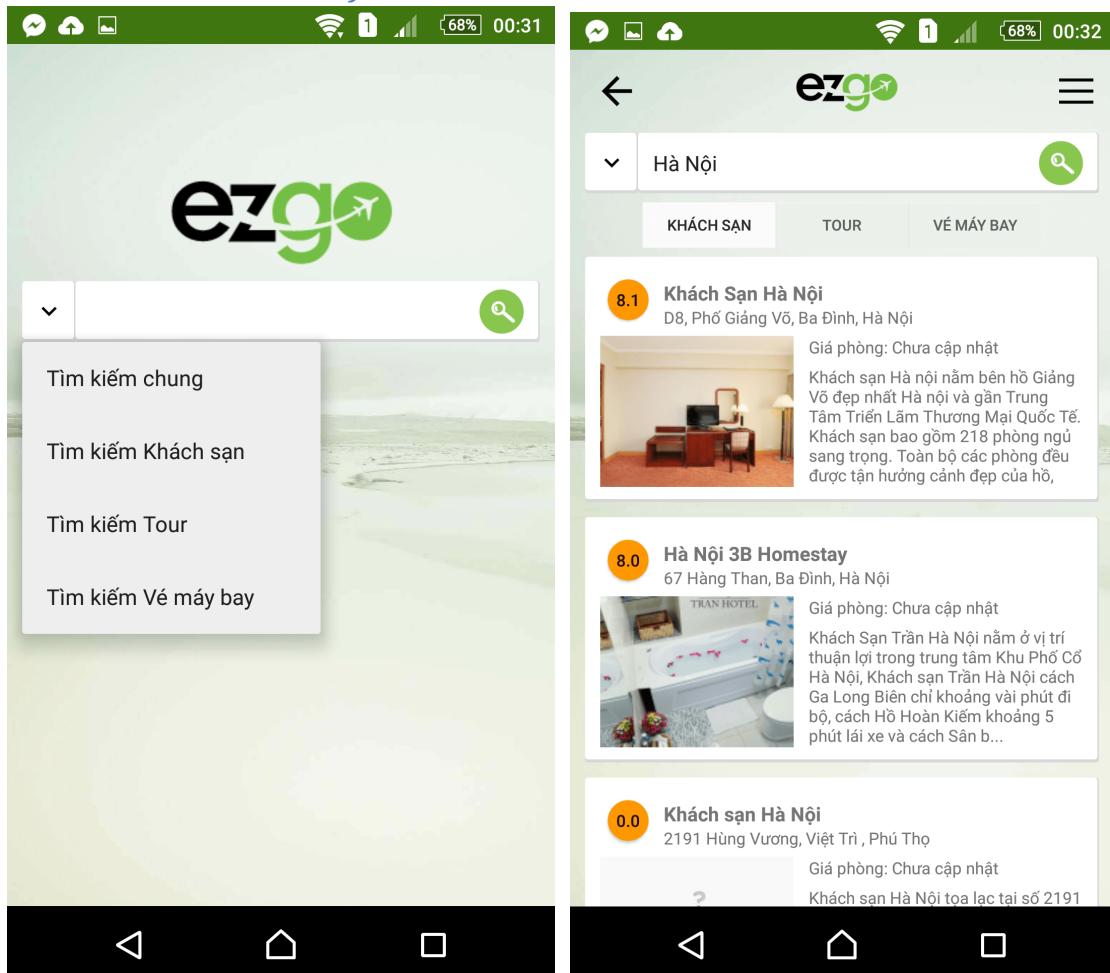
2.2. Mobile application

2.2.1. General search



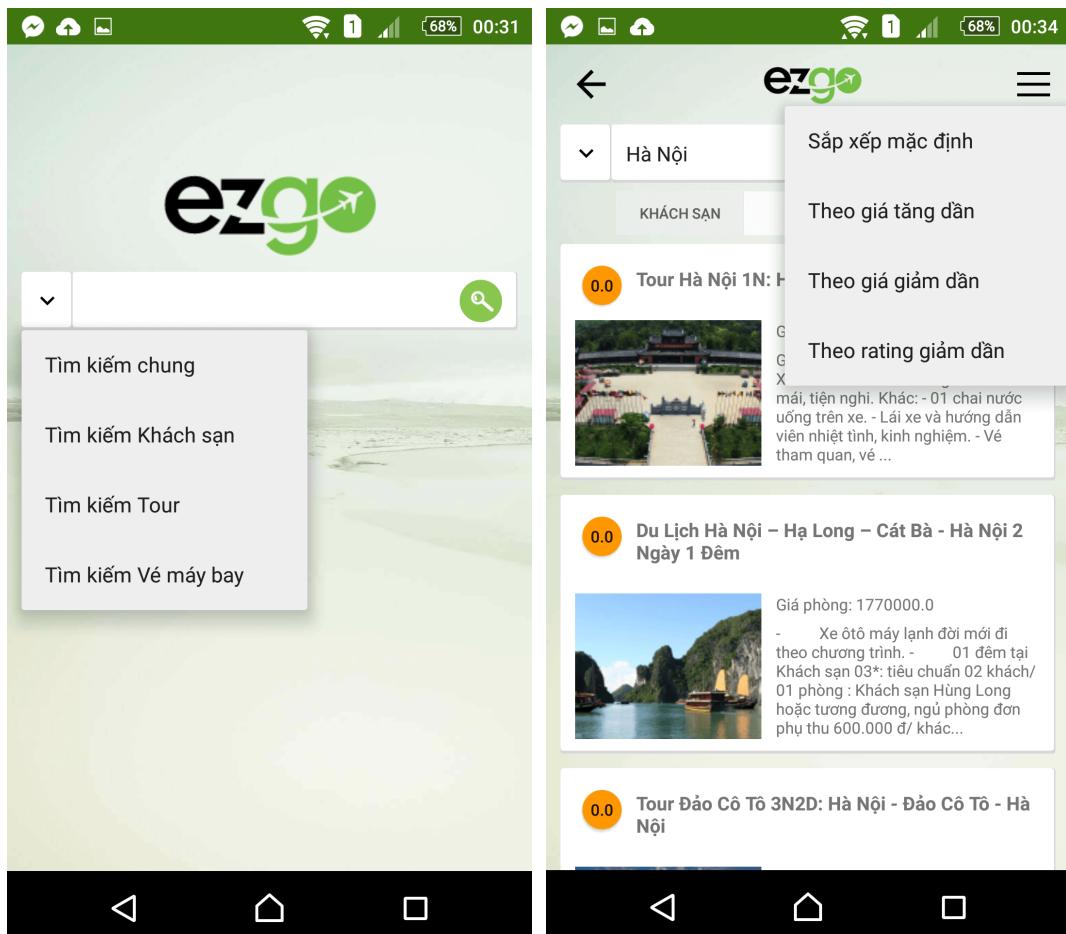
Step	Description
1	Input keyword to the textbox.
2	Touch to the button with magnitude glass icon.

2.2.2. Search for hotels



Step	Description
1	Select “Tìm kiếm khách sạn” from the drop down list.
2	Select province from the drop down list. Input keywords to the textbox.
3	Touch to the button with magnitude glass icon.

2.2.3. Search for tours



Step	Description
1	Select “Tim kiem Tour” from the drop down list.
2	Select province from the drop down list. Input keywords to the textbox.
3	Touch to the button with magnitude glass icon.

G. Appendix

1. **The Unified Modeling Language** [Online]. Available: <http://www.uml-diagrams.org/>. Accessed on: Mar. 2016.
2. **C# Coding Convention** [Online]. Available: <http://msdn.microsoft.com/en-us/library/vstudio/ff926074.aspx>. Accessed on: Jan. 2016
3. Toby Segaran, “**Programming Collective Intelligence**”, Sebastopol, CA, USA: O'Reilly Media Inc., 2007.
4. Andrew Smith, **Introducing to Lucene .NET** [Online]. Available: <http://www.codeproject.com/Articles/29755/Introducing-Lucene-Net>. Accessed on: Jan 2016.
5. **Lucene .NET 3.0.3 Documentation** [Online]. Available: https://lucenenet.apache.org/docs/3.0.3/db/d65/class_lucene_1_1_net_1_1_documents_1_1_document.html. Accessed on: Jan. 2016.
6. **Lucene Similarity** [Online]. Available: https://lucene.apache.org/core/3_6_0/api/all/org/apache/lucene/search/Similarity.html. Accessed on: Feb. 2016.
7. **Lucene Scoring** [Online]. Available: <http://www.lucenetutorial.com/advanced-topics/scoring.html>. Accessed on: Feb. 2016.
8. Doug Turnbull, **Build your own Custom Lucene query and scorer** [Online]. Available: <http://opensourceconnections.com/blog/2014/01/20/build-your-own-custom-lucene-query-and-scorer/>. Accessed on: Feb. 2016.
9. Rion Williams, **Preven Repeated Request using ActionFilters in ASP .NET MVC** [Online]. Available: <http://www.codeproject.com/Articles/582845/PreventplusRepeatedplusRequestsplususingplusAction>. Accessed on: Mar. 2016.
10. Brian Dean, **Google's 200 Ranking Factors: The Complete List** [Online]. Available: <http://backlinko.com/google-ranking-factors>. Accessed on: Mar. 2016.