

|  |
| --- |
| **FPT UNIVERSITY** |
| Capstone Project Document  Report 3  Project Management Plan |
| |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | |  |  | | --- | --- | | **Group 18** | | | **Group Members** | Đào Bảo Long – Team Leader – SE60690  Lê Phúc Lữ – Team Member – 60563  Nguyễn Thanh Tùng – Team Member – SE60609  Nguyễn Tấn Đức – Team Member – SE60660 | | **Supervisor** | Nguyễn Trọng Tài | | **Ext Supervisor** | N/A | | **Capstone Project Code** | i-Deliver |   Build a Web Application  for manages all activities of  delivery service system by coach. | |
|  |
|  |
| Hồ Chí Minh City, January 2014 |

|  |
| --- |
| *This page is intentionally left blank* |

# Table of figures

[Figure 1 Model-View-Controller Architecture 4](#_Toc384340552)

[Figure 2 Home page 28](#_Toc384340553)

[Figure 3 Search Routes Page 29](#_Toc384340554)

[Figure 4 Price Table Page 31](#_Toc384340555)

[Figure 5 Sent Request Page 32](#_Toc384340556)

[Figure 6 Payment Page 33](#_Toc384340557)

[Figure 7 Request management page (Customer role) 34](#_Toc384340558)

[Figure 8 Request Management Page (Staff role) 35](#_Toc384340559)

[Figure 9 Administrators Page 36](#_Toc384340560)

[Figure 10 Manage Fee Page 36](#_Toc384340561)

[Figure 11 Physical database design 45](#_Toc384340562)

[Diagram 1 Component Diagram 5](#_Toc384337898)

[Diagram 2 Entities Class Diagram 6](#_Toc384337899)

[Diagram 3 Algorithm Processing Class Diagram 17](#_Toc384337900)

[Diagram 4 Business Logic Class Diagram 19](#_Toc384337901)

[Diagram 5 State machine diagram: Requst status transition 23](#_Toc384337902)

[Diagram 6 State machine diagram: Package status transition 23](#_Toc384337903)

[Diagram 7 Sequence diagram: Post delivery requests 24](#_Toc384337904)

[Diagram 8 Sequence diagram: Search Routes 24](#_Toc384337905)

[Diagram 9 Sequence diagram: Create a trip 24](#_Toc384337906)

[Diagram 10 Sequence diagram: Edit a station 25](#_Toc384337907)

[Diagram 11 Sequence diagram: Add a price 26](#_Toc384337908)

[Diagram 12 Sequence diagram: Create a Staff Account 27](#_Toc384337909)

[Diagram 13 Sequence diagram: Approve requests 27](#_Toc384337910)

[Diagram 14 Sequence diagram: Assign package 28](#_Toc384337911)

# Software Design Description (SDD)



## Design Overview

The SDD describes the system architectural design and the detailed designs, including the user interfaces, of the system.

The components should communicate through interfaces. The detailed implementation of each component should be transparent to other components. The passive MVC III pattern is the preferable architecture for the website.

The detailed designs of the system should adopt the basic principles of software design, including “high cohesion, low coupling” and “open to extension, closed to modification” principles.

Each of the following sections is summarized below:

* Section 4.2: Gives a specification of the system architecture design, describing the overall architecture of the system and subsystems.
* Section 4.3: Specifies all the components that should be included in the system, and the communications between them.
* Section 4.4: Describes the class and package diagrams, as well as their descriptions in details, to provide a static view of the system.
* Section 4.4: Describes the sequence diagrams for the use cases specified in the SRS, to provide a dynamic view of the system.
* Section 4.5: Describes the user interfaces, to specify what will be seen by the users of the system.
* Section 4.5: Describes the Database Design, including the relations and the relationships between them.

## System Architectural Design

The MVC III (Model – View – Controller) pattern is used as the overall system architecture, because it specifies a clear distinction between the responsibilities of the components and is appropriate for developing web applications.

3. Exchange data

4. Results

**CONTROLLER**

* Intercepts user input;
* Coordinates the view and model;
* Handles communication between the model and data layer.

**VIEW**

* Binds to the model;
* Renders the UIs (HTML, CSS, JavaScript);
* Allows navigating between controllers.

**MODEL**

* Communicates with data source;
* Exposes functionalities, business logics, and data validation.

Database   
Server

Application Server

1. Request

8. Response

5. Select views

7. Changes made

2. Invoke methods

6. Query states

Method invocations

Change notifications

Figure 1 Model-View-Controller Architecture

## Component Diagram

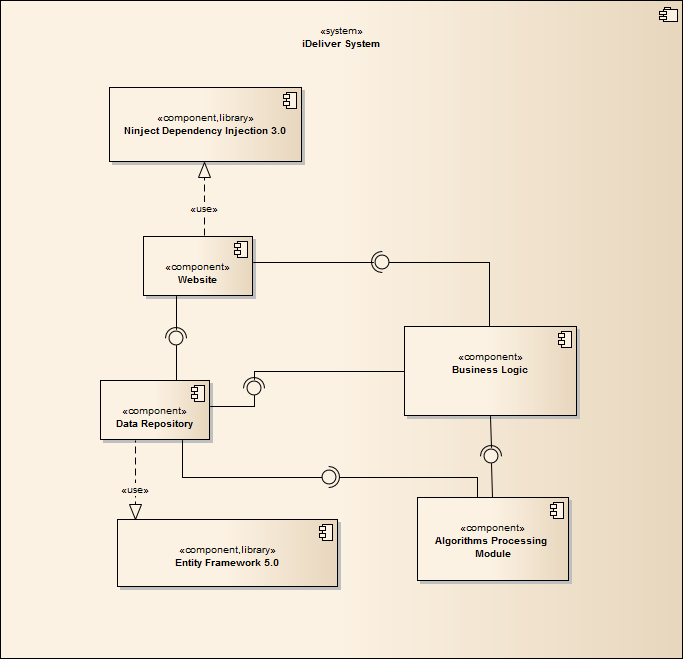


Diagram 1 Component Diagram

The i-Deliver system includes the following major components:

* **Entity Framework 5.0:** An object-relational mapper that enables working with relational data persistence using domain-specific objects;
* **Data Repository:** A data store that provides data access functionality, using the Entity Framework to communicates with the database;
* **Data Synchronization Module:** A module that communicates with the services provided by the BMT hotel data provider and uses interfaces exposed by the Data Repository to persist the data;
* **BMT Services:** A set of web services that provide information about hotels, room types, and price changes for its consumers;
* **Business Logic:** A module that uses interfaces exposed by the Data Repository to encapsulate important businesses of in the I-DELIVER system;
* **Ninject Dependency Injection 3.0:** A lightweight dependency injection framework for .NET applications which helps split the application into a collection of loosely-coupled, highly-cohesive pieces, and then glue them back together in a flexible manner.
* **Website:** A web-application that helps exposes all the functionalities to end-users.

From the point of view of the MVC-II pattern, the components Entity Framework 5.0, Data Repository, Data Synchronization Module, BMT Services, Business Logic, and Ninject Dependency Injection 3.0 all belong to the Model part. The Controller and the View parts are wrapped in the component Website.

## Detailed Designs



### Entity Classes



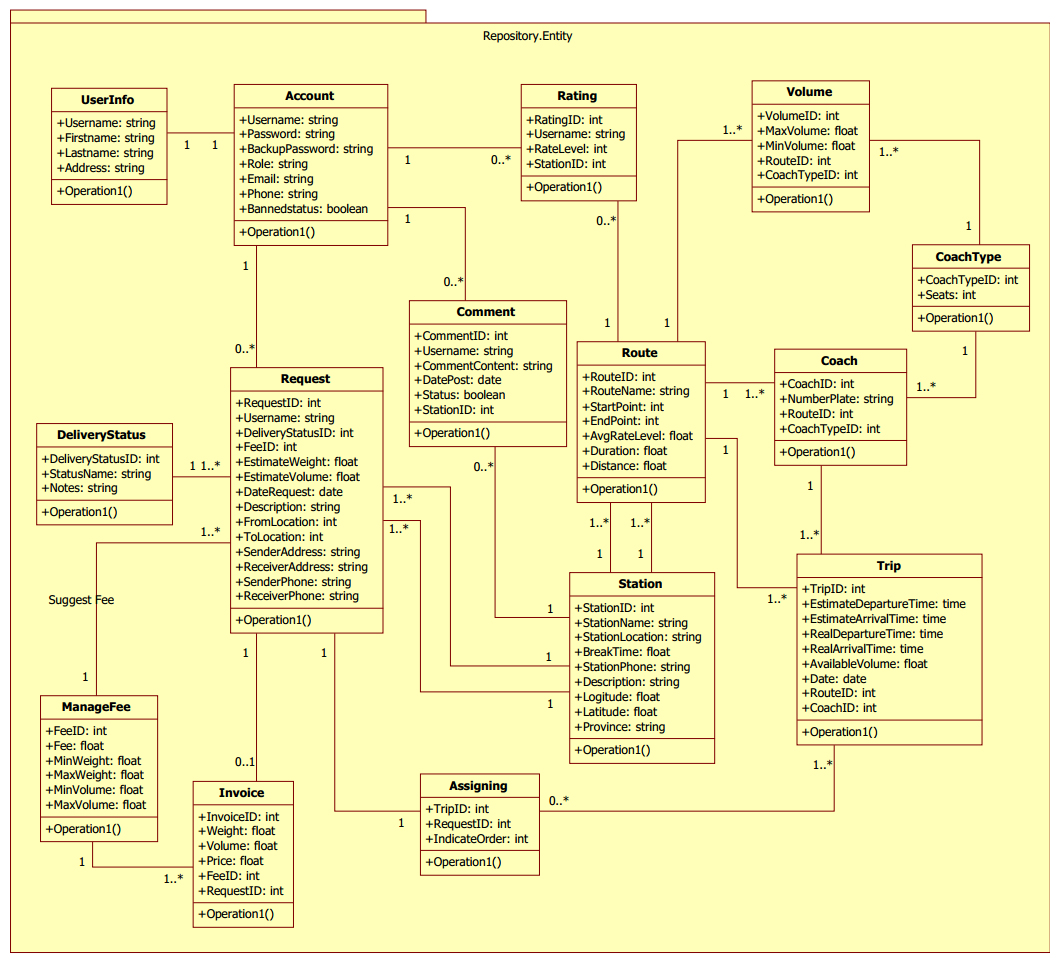


Diagram 2 Entities Class Diagram

#### Account Class

**Properties**

|  |  |  |  |
| --- | --- | --- | --- |
| Property name | Type | Visibility | Description |
| *Username* | string | public | Unique usename that use to logon the system. |
| *Password* | string | public | Password of account. |
| *BackupPassword* | string | public | Backup password use to confirm when user lost password. |
| *Role* | string | public | Role of account (Admin, Staff or Customer). |
| *Email* | string | public | Email address of customer. |
| *Phone* | string | public | Phone number of customer. |
| *BannedStatus* | Boolean | public | Status of account (Ban or Unban). |

**Operations:** N/A

#### UserInfo Class

**Properties**

|  |  |  |  |
| --- | --- | --- | --- |
| Property name | Type | Visibility | Description |
| *Username* | string | public | Unique usename that use to logon the system. |
| *Firstname* | string | public | Firstname of customer. |
| *Lastname* | string | public | Lastname of customer. |
| *Address* | string | public | Main address of customer. |

**Operations:** N/A

#### Route Class

**Properties**

|  |  |  |  |
| --- | --- | --- | --- |
| Property name | Type | Visibility | Description |
| *RouteID* | int | public | Unique id that differentiates an route from the others. |
| *RouteName* | string | public | Name of route. |
| *StartPoint* | int | public | Starting point of route. |
| *EndPoint* | int | public | Ending point of route. |
| *AvgRateLevel* | float | public | Average rating level of route. |
| *Duration* | float | public |  |
| *Distance* | float | public | Distance of route between of starting point and ending point. |

**Operations:** N/A

#### Trip Class

**Properties**

|  |  |  |  |
| --- | --- | --- | --- |
| Property name | Type | Visibility | Description |
| *TripID* | int | public | Unique id that differentiates a trip from the others. |
| *EstimateDepartureTime* | time | public | Estimate time when the trip begins. |
| *EstimateArrivalTime* | time | public | Estimate time when the trip arrives. |
| *RealDepartureTime* | time | public | Real time when the trip begins. |
| *RealArrivalTime* | time | public | Real time when the trip arrives. |
| *AvailableVolume* | string | public | Volume available of storage space in coach. |
| *Date* | date | public | Date of trip. |
| *RouteID* | int | public | Id of the route which the trip belongs to. |
| *CoachID* | int | public | Id of the coach which the trip belongs to. |

**Operations:** N/A

#### Station Class

**Properties**

|  |  |  |  |
| --- | --- | --- | --- |
| Property name | Type | Visibility | Description |
| *StationID* | int | public | Unique id that differentiates a province from the others |
| *StationName* | string | public | Name of the station. |
| *StationLocation* | string | public | Location of the station. |
| *BreakTime* | float | public | Break time when the coach arrive the station. |
| *StationPhone* | string | public | Phone number of the station. |
| *Description* | string | public | Description of the station. |
| *Longitude* | float | public | Longitude address of the station. |
| *Latitude* | float | public | Latitude address of the station. |
| *Province* | string | public | Province address of the station. |

**Operations:** N/A

#### Coach Class

**Properties**

|  |  |  |  |
| --- | --- | --- | --- |
| Property name | Type | Visibility | Description |
| *CoachID* | int | public | Unique id that differentiates a coach from the others |
| *NumberPlate* | string | public | Number plate of coach. |
| *RouteID* | int | public | Id of the route which the coach belongs to. |
| *CoachTypeID* | int | public | Id of the coach type which the coach belongs to. |

**Operations:** N/A

#### CoachType Class

**Properties**

|  |  |  |  |
| --- | --- | --- | --- |
| Property name | Type | Visibility | Description |
| *CoachTypeID* | int | public | Unique id that differentiates a coach type from the others. |
| *Seats* | int | public | Total number of seats in coach. |

**Operations:** N/A

#### Volume Class

**Properties**

|  |  |  |  |
| --- | --- | --- | --- |
| Property name | Type | Visibility | Description |
| *VolumeID* | int | public | Unique id that differentiates a volume from the others. |
| *MaxVolume* | float | public | The maximum volume that coach may contain. |
| *MinVolume* | float | public | The minimum volume that coach may contain. |
| *RouteID* | int | public | Id of the route which the coach belongs to. |
| *CoachTypeID* | int | public | Id of the coach type which the coach belongs to. |

**Operations:** N/A

#### Comment Class

**Properties**

|  |  |  |  |
| --- | --- | --- | --- |
| Property name | Type | Visibility | Description |
| *CommentID* | int | public | Unique id that differentiates a comment from the others |
| *Username* | string | public | Username pf customer comment on the station. |
| *CommentContent* | string | public | Content of comment. |
| *DatePost* | date | public | Date sent comment. |
| *Status* | boolean | public | Status of commen, |
| *StationID* | int | public | Id of the station which the comment belongs to. |

**Operations:** N/A

#### Rating Class

**Properties**

|  |  |  |  |
| --- | --- | --- | --- |
| Property name | Type | Visibility | Description |
| *RatingID* | int | public | Unique id that differentiates a rating from the others |
| *Username* | string | public | Username of customer who rating on the station. |
| *RateLevel* | int | public | Level of rating. |
| *StationID* | int | public | Id of the station which the rating belongs to. |

**Operations:** N/A

#### Assigning Class

**Properties**

|  |  |  |  |
| --- | --- | --- | --- |
| Property name | Type | Visibility | Description |
| *TripID* | int | public | Id of the trip which the assigning relates |
| *RequestID* | int | public | Id of the request which the assigning relates |
| *IndicateOrder* | int | public | Sequence of assigning. |

**Operations:** N/A

#### DeliveryStatus Class

**Properties**

|  |  |  |  |
| --- | --- | --- | --- |
| Property name | Type | Visibility | Description |
| *DeliveryStatusID* | int | public | Unique id that differentiates a delivery status from the others. |
| *StatusName* | string | public | Name of a delivery status. |
| *Notes* | string | public | Notes of a delivery status. |

**Operations:** N/A

#### Request Class

**Properties**

|  |  |  |  |
| --- | --- | --- | --- |
| Property name | Type | Visibility | Description |
| *RequestID* | int | public | Unique id that differentiates a request status from the others. |
| *Username* | string | public | Username of customer who sent request delivery package. |
| *DeliveryStatusID* | int | public | Id of the delivery status which the request relates. |
| *FeeID* | int | public | Id of the fee which the request relates. |
| *EstimateWeight* | float | public | Estimate weight of package. |
| *EstimateVolume* | float | public | Estimate volume of package (length, width, high). |
| *DateRequest* | date | public | Date sent request. |
| *Description* | string | public | Description of request. |
| *FromLocation* | int | public | Send from location. |
| *ToLocation* | int | public | Send to location. |
| *SenderAddress* | string | public | Address of sender. |
| *ReceiverAddress* | string | public | Address of receiver. |
| *SenderPhone* | string | public | Phone number of sender. |
| *ReceiverPhone* | string | public | Phone number of receiver. |

**Operations:** N/A

#### ManageFee Class

**Properties**

|  |  |  |  |
| --- | --- | --- | --- |
| Property name | Type | Visibility | Description |
| *FeeID* | int | public | Unique id that differentiates a fee from the others |
| *Fee* | float | public | Customer pay fee to use service. |
| *MinWeight* | float | public | Minimum weight of package. |
| *MaxWeight* | float | public | Maximum weight of package. |
| *MinVolume* | float | public | Minimum volume of package. |
| *MaxVolume* | float | public | Maximum volume of package. |

**Operations:** N/A

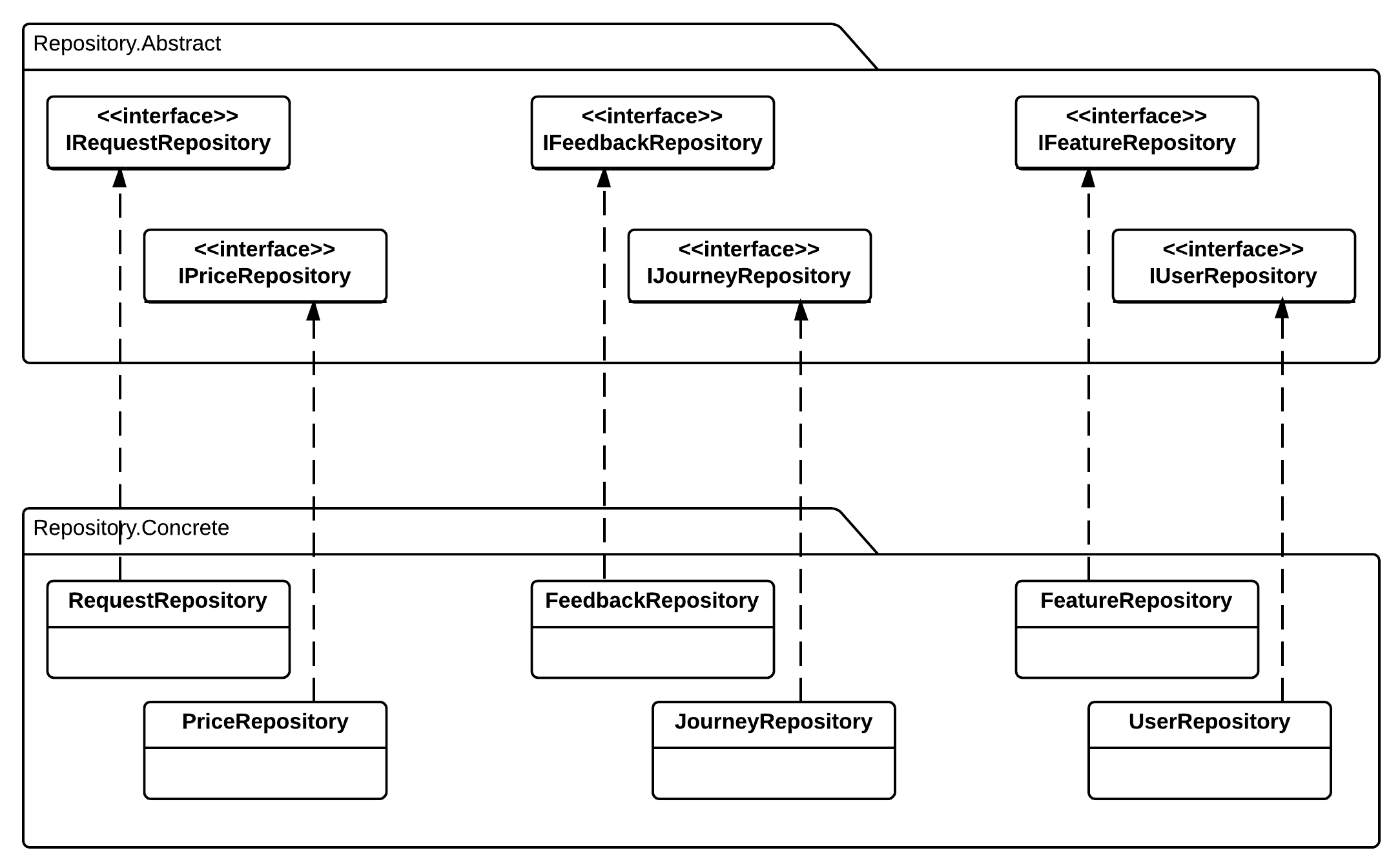
#### Invoice Class

**Properties**

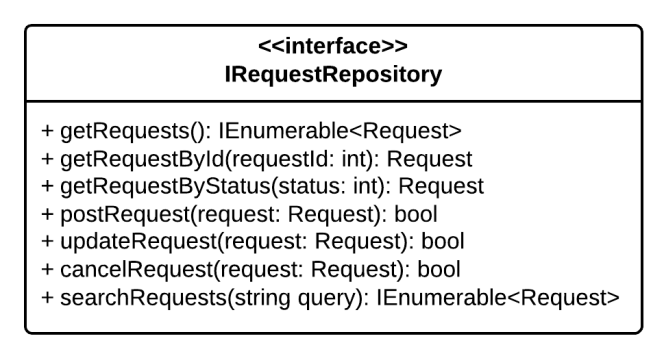
|  |  |  |  |
| --- | --- | --- | --- |
| Property name | Type | Visibility | Description |
| *InvoiceID* | int | public | Unique id that differentiates a invoice from the others |
| *Weight* | float | public | Weight of the package. |
| *Volume* | float | public | Volume of the package. |
| *Price* | float | public | Total price that customer must pay. |
| *FeeID* | int | public | Id of the fee which the invoice relates. |
| *RequestID* | int | public | Id of the request which the invoice belongs to. |

**Operations:** N/A

#### Class diagram



#### IRequestRepository Interface



**Properties:** N/A

**Operations**

|  |  |  |
| --- | --- | --- |
| Signature: IEnumerable<Request> Get Requests()  Description: Retrieves all the requests from the database | | |
| N/A | N/A | N/A |

|  |  |  |
| --- | --- | --- |
| Signature: Request GetRequestById(int requestId)  Description: Retrieves a request from the database given its id | | |
| Parameter | **Type** | **Description** |
| *requestId* | int | Id of the request to be retrieved |

|  |  |  |
| --- | --- | --- |
| Signature: IEnumerable<Request> GetRequestByStatus(int status)  Description: Retrieves all the request from the database given its status | | |
| Parameter | **Type** | **Description** |
| *Status* | Int | Status of the request to be retrived |

|  |  |  |
| --- | --- | --- |
| Signature: bool postRequest(Request request)  Description: post new delivery request to system and save to database | | |
| Parameter | **Type** | **Description** |
| *request* | Request | New request to be saved |

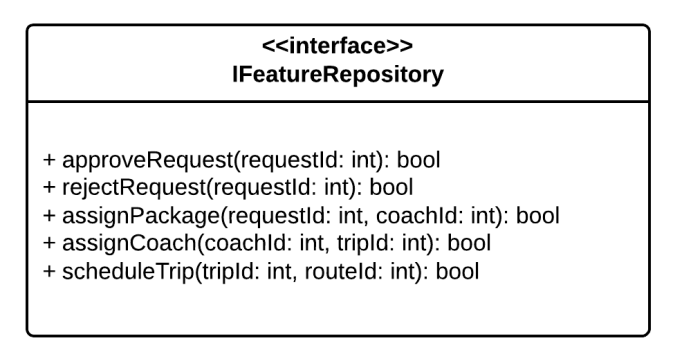
|  |  |  |
| --- | --- | --- |
| Signature: bool updateRequest(int requestId)  Description: Update existing request to the database given its id | | |
| Parameter | **Type** | **Description** |
| *requestId* | int | Id of the request to be updated |

|  |  |  |
| --- | --- | --- |
| Signature: bool cancelRequest(int requestId)  Description: Cancel/Remove existing request from the database given its id | | |
| Parameter | **Type** | **Description** |
| *requestId* | int | Id of the request to be updated |

|  |  |  |
| --- | --- | --- |
| Signature: IEnumerable<Request> serchRequest(string query)  Description: Retrieves all the request from the database given by query string | | |
| Parameter | **Type** | **Description** |
| *query* | String | Query string to find matching request |

|  |  |  |
| --- | --- | --- |
| Signature:  Description: | | |
| Parameter | **Type** | **Description** |
| *provinceId* | int | Id of the hotel to be checked |

#### IFeatureRepository Interface



**Properties:** N/A

**Operations**

|  |  |  |
| --- | --- | --- |
| Signature: bool approveRequest(int requestId)  Description: Approve pending requests | | |
| Parameter | **Type** | **Description** |
| *requestId* | int | Id of the request to be approved |

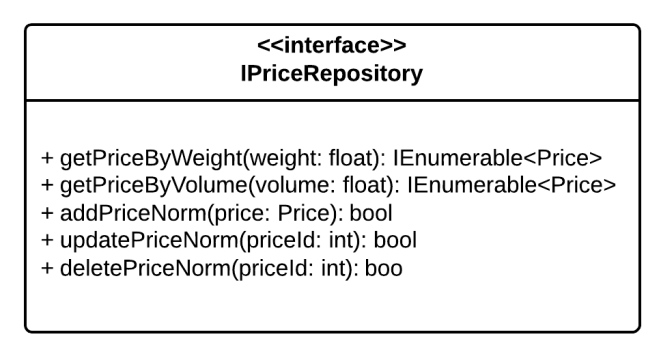
|  |  |  |
| --- | --- | --- |
| Signature: bool rejectRequest(int requestId)  Description: Reject pending requests | | |
| Parameter | **Type** | **Description** |
| *requestId* | int | Id of the request to be rejected |

|  |  |  |
| --- | --- | --- |
| Signature: bool assignPackage(int requestId, int coachId)  Description: Assign approved package for specified coach | | |
| Parameter | **Type** | **Description** |
| *requestId* | int | Id of the package to be assigned |
| *coachId* | int | Id of the coach to assign package |

|  |  |  |
| --- | --- | --- |
| Signature: bool assignCoach(int coachId, int tripId)  Description: Assign specific coach to for specified trip | | |
| Parameter | **Type** | **Description** |
| *coachId* | int | Id of the coach to be assigned |
| *tripId* | int | Id of the trip to assign coach |

|  |  |  |
| --- | --- | --- |
| Signature: bool scheduleTrip(int tripId, int routeId)  Description: Schedule trips for specified route | | |
| Parameter | **Type** | **Description** |
| *tripId* | int | Id of the trip to be schedule |
| *routeId* | int | Id of the route for trip scheduling |

#### IPriceRepository Interface



**Properties:** N/A

**Operations**

|  |  |  |
| --- | --- | --- |
| Signature: IEnumerable<Price> GetPriceByWeight(float weight)  Description: Get delivery price/fee based on package weight | | |
| Parameter | **Type** | **Description** |
| *weight* | float | Weight of delivered package |

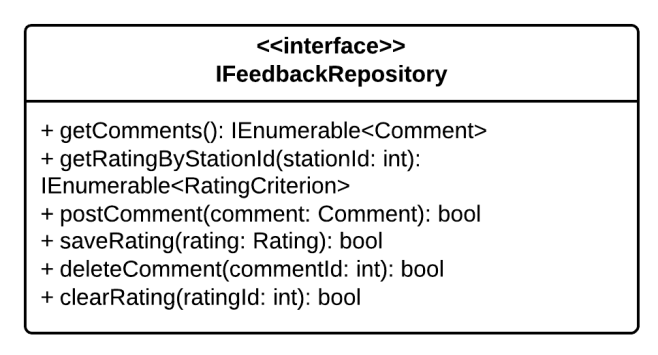
|  |  |  |
| --- | --- | --- |
| Signature: IEnumerable<Price> GetPriceByVolume(float volume)  Description: Get delivery price/fee based on package volume | | |
| Parameter | **Type** | **Description** |
| *volume* | float | Volume of delivered package |

|  |  |  |
| --- | --- | --- |
| Signature: bool addPriceNorm(Price price)  Description: Add new price/fee norm to database | | |
| Parameter | **Type** | **Description** |
| *price* | Price | New price norm to be added to database |

|  |  |  |
| --- | --- | --- |
| Signature: bool updatePriceNorm(priceId: int)  Description: Update a price norm to the database | | |
| Parameter | **Type** | **Description** |
| *priceId* | int | The price norm to be saved |

|  |  |  |
| --- | --- | --- |
| Signature: bool updatePriceNorm(priceId: int)  Description: Delete/Remove a price norm from the database | | |
| Parameter | **Type** | **Description** |
| *priceId* | int | The price norm to be deleted |

#### IFeedbackRepository Interface



**Properties:** N/A

**Operations**

|  |  |  |
| --- | --- | --- |
| Signature: IEnumerable<Comment> GetComments()  Description: Retrieves all the comments from the database | | |
| Parameter | **Type** | **Description** |
| N/A | N/A | N/A |

|  |  |  |
| --- | --- | --- |
| Signature: IEnumerable<RatingCriterion> GetRatingByStationId(int stationId)  Description: Retrieves all the rating that belong to a given station from the database | | |
| Parameter | **Type** | **Description** |
| *stationId* | int | Id of the station that has the rating to be retrieved |

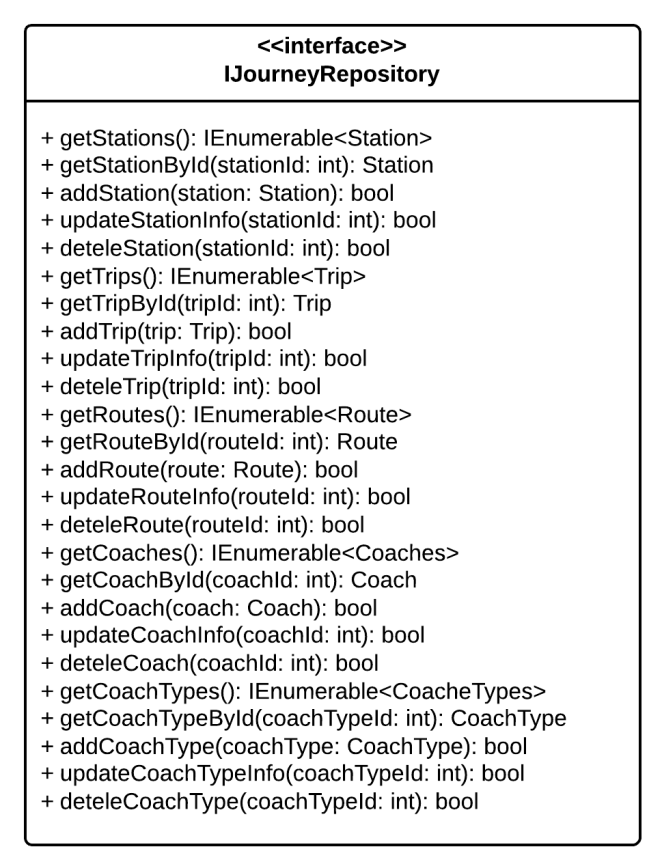
|  |  |  |
| --- | --- | --- |
| Signature: bool postComment(Comment comment)  Description: Post new comment and save to database | | |
| Parameter | **Type** | **Description** |
| *comment* | Comment | The new comment to be saved to database |

|  |  |  |
| --- | --- | --- |
| Signature: bool saveRating(Rating rating)  Description: Save station rating to database | | |
| Parameter | **Type** | **Description** |
| *rating* | Rating | The new station rating to be saved to database |

|  |  |  |
| --- | --- | --- |
| Signature: bool deleteComment(int commentId)  Description: Marks a coment as deleted in the database | | |
| Parameter | **Type** | **Description** |
| *commentId* | int | Id of the comment to be deleted |

|  |  |  |
| --- | --- | --- |
| Signature: bool clearRating(int ratingId)  Description: Marks a station rating as deleted in the database | | |
| Parameter | **Type** | **Description** |
| *ratingId* | Int | Id of the rating to be deleted |

#### IJourneyRepository Interface



**Properties:** N/A

**Operations**

|  |  |  |
| --- | --- | --- |
| Signature: IEnumerable<Station> GetStations()  Description: Retrieves all the stations from the database | | |
| Parameter | **Type** | **Description** |
| N/A | N/A | N/A |

|  |  |  |
| --- | --- | --- |
| Signature: Station GetStationByStationId(int stationId)  Description: Retrieves the station given its station id from the database | | |
| Parameter | **Type** | **Description** |
| *stationId* | int | Id of the station to be retrieved |

|  |  |  |
| --- | --- | --- |
| Signature: Station AddStation(Station station)  Description: Add new station to the database | | |
| Parameter | **Type** | **Description** |
| *station* | Station | The new station to be added to database |

|  |  |  |
| --- | --- | --- |
| Signature: bool UpdateStation(int stationId)  Description: Update station information from given its id from the database | | |
| Parameter | **Type** | **Description** |
| *stationId* | int | Id of the station to be updated |

|  |  |  |
| --- | --- | --- |
| Signature: bool DeleteStation(int stationId)  Description: Delete the station given its it from the database | | |
| Parameter | **Type** | **Description** |
| *stationId* | int | Id of the station to be deleted |

|  |  |  |
| --- | --- | --- |
| Signature: IEnumerable<Trip> GetTrips()  Description: Retrieves all the trips from the database | | |
| Parameter | **Type** | **Description** |
| N/A | N/A | N/A |

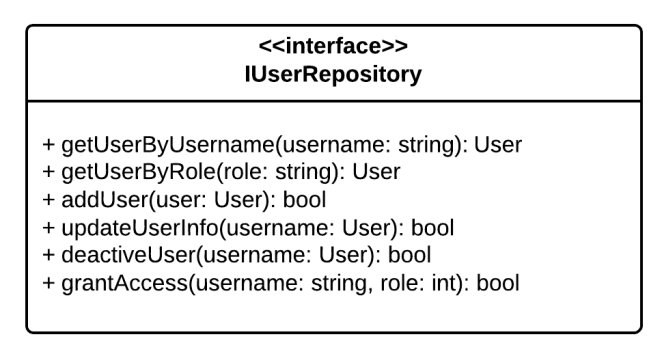
|  |  |  |
| --- | --- | --- |
| Signature: Station GetTripByTripId(int tripId)  Description: Retrieves the station given its station id from the database | | |
| Parameter | **Type** | **Description** |
| *tripId* | int | Id of the trip to be retrieved |

|  |  |  |
| --- | --- | --- |
| Signature: Station AddTrip(Trip trip)  Description: Add new trip to the database | | |
| Parameter | **Type** | **Description** |
| *trip* | Trip | The new trip to be added to database |

|  |  |  |
| --- | --- | --- |
| Signature: bool UpdateTrip(int tripId)  Description: Update trip information from given its id from the database | | |
| Parameter | **Type** | **Description** |
| *tripId* | int | Id of the trip to be updated |

|  |  |  |
| --- | --- | --- |
| Signature: bool DeleteTrip(int tripId)  Description: Delete the trip given its it from the database | | |
| Parameter | **Type** | **Description** |
| *tripId* | int | Id of the trip to be deleted |

#### IUserRepository Interface



**Properties:** N/A

**Operations**

|  |  |  |
| --- | --- | --- |
| Signature: User GetUserByUsername(String username)  Description: Retrieves a user given its username from the database | | |
| Parameter | **Type** | **Description** |
| *username* | String | Username of user to be retrieved |

|  |  |  |
| --- | --- | --- |
| Signature: User GetUserByRole(String role)  Description: Retrieves a user given its role from the database | | |
| Parameter | **Type** | **Description** |
| *role* | String | Role of user to be retrieved |

|  |  |  |
| --- | --- | --- |
| Signature: bool addUser(User user)  Description: Add new user to the database | | |
| Parameter | **Type** | **Description** |
| *user* | UserProfile | The user to be saved |

|  |  |  |
| --- | --- | --- |
| Signature: bool updateUser(int username)  Description: Update/Save user information to the database | | |
| Parameter | **Type** | **Description** |
| *username* | String | Username of user to be updated |

|  |  |  |
| --- | --- | --- |
| Signature: bool GrantAccess(String username, int roleId)  Description: Grants the selected role the the user with the user id | | |
| Parameter | **Type** | **Description** |
| *username* | String | Username of the user to be granted access |
| *roleId* | Role | Role that will be granted to the user |

|  |  |  |
| --- | --- | --- |
| Signature: bool DeactivateUser(String username)  Description: Marks the the user as deactivated in the database | | |
| Parameter | **Type** | **Description** |
| *username* | String | Username of the user to be marked as deactivated |

### Algoritms Processing Classes



#### Class diagram

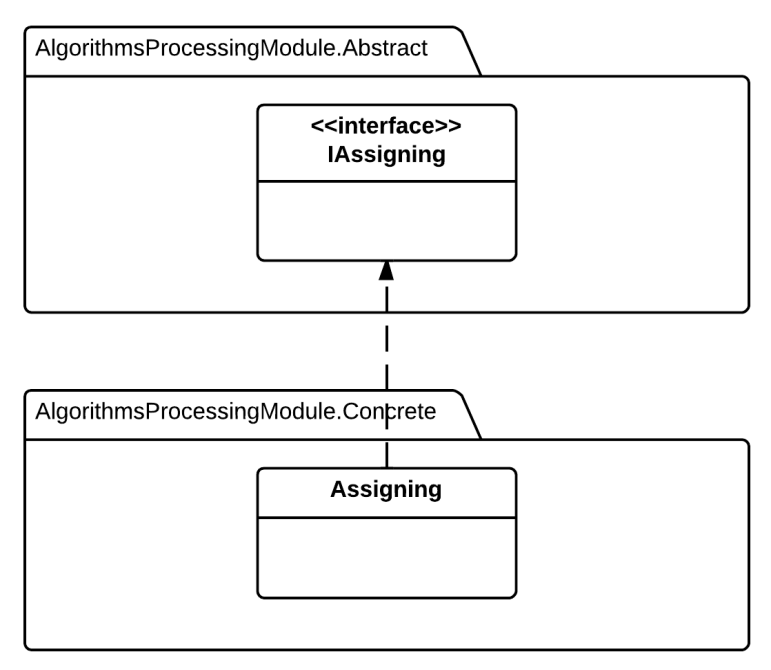
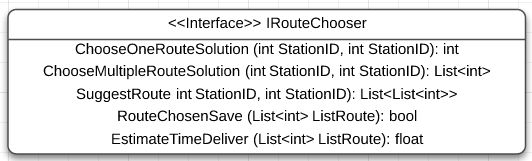


Diagram 3 Algorithm Processing Class Diagram

#### IRouteChooser



**Properties:** N/A

**Operations**

|  |  |  |
| --- | --- | --- |
| **Signature:** int ChooseOneRouteSolution(int StartStation, int EndStation)  **Description:** return one suitable route to deliver from the start station to the end station, if the route not exsit, return -1. | | |
| **Parameter** | **Type** | **Description** |
| StartStation | int | The ID of the start station of the request |
| EndStation | int | The ID of the end station of the request |

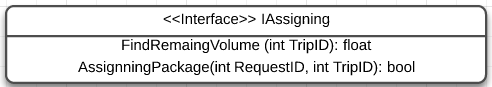
|  |  |  |
| --- | --- | --- |
| **Signature:** List <int> ChooseMultipleRouteSolution(int StartStation, int EndStation)  **Description:** return the list of suitable route to deliver from the start station to the end station. | | |
| **Parameter** | **Type** | **Description** |
| StartStation | int | The ID of the start station of the request |
| EndStation | int | The ID of the end station of the request |

|  |  |  |
| --- | --- | --- |
| **Signature:** List<int> SuggestRoute(int StartStation, int EndStation)  **Description:** Suggest for staff the best suitable route or list of routes base on the input start and end station of request. | | |
| **Parameter** | **Type** | **Description** |
| StartStation | int | The ID of the start station of the request |
| EndStation | int | The ID of the end station of the request |

|  |  |  |
| --- | --- | --- |
| **Signature:** bool RouteChosenSave (List<int> RouteList)  **Description:** Retrieves the list of route and save them to the table with exactly indicate order. | | |
| **Parameter** | **Type** | **Description** |
| RouteList | List<int> | List of route that staff choose or system suggest |

|  |  |  |
| --- | --- | --- |
| **Signature:** float EstimateTimeDeliver (List<int> RouteList)  **Description:** Find the estimate time to deliver the package base on the input route list and their relative information. | | |
| **Parameter** | **Type** | **Description** |
| RouteList | List<int> | List of route that staff choose or system suggest |

#### IAssigning



**Properties:** N/A

**Operations**

|  |  |  |
| --- | --- | --- |
| **Signature:** float FindRemaingVolume (int TripID)  **Description:** Return the remaining volume that the coach on input trip can be serve. | | |
| **Parameter** | **Type** | **Description** |
| TripID | int | The ID of trip that staff want to estimate. |

|  |  |  |
| --- | --- | --- |
| **Signature:** List <int> AssigningPackage (int RequestID)  **Description:** Automatically find the list of trip suitable for deliver the package. | | |
| **Parameter** | **Type** | **Description** |
| RequestID | int | The ID of request that staff processing. |

### Business Logic Classes



#### Class diagram

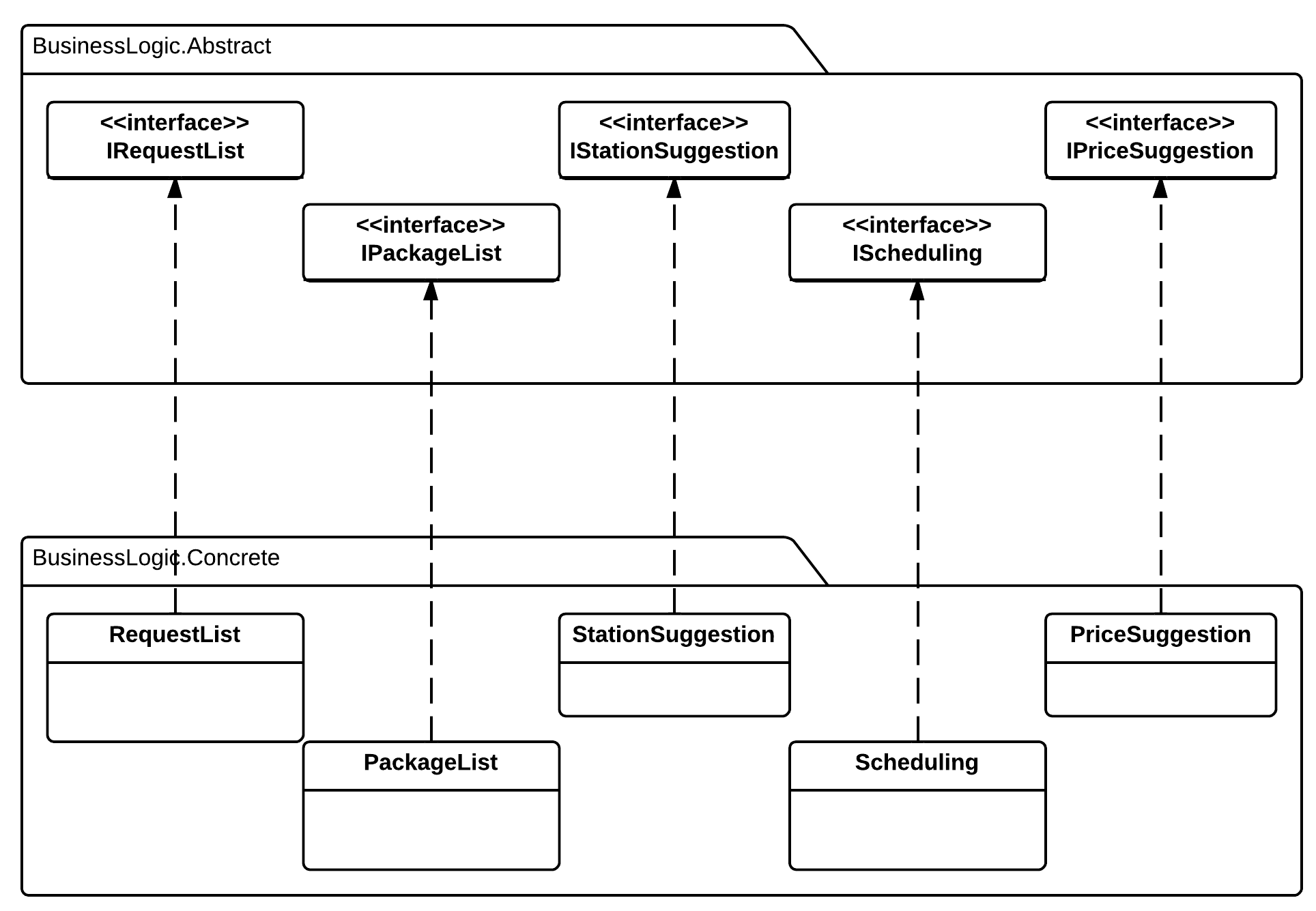
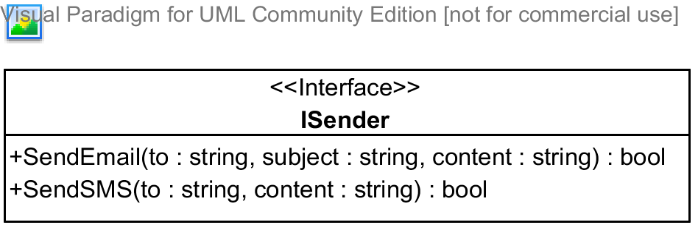


Diagram 4 Business Logic Class Diagram

#### IRequestList Interface



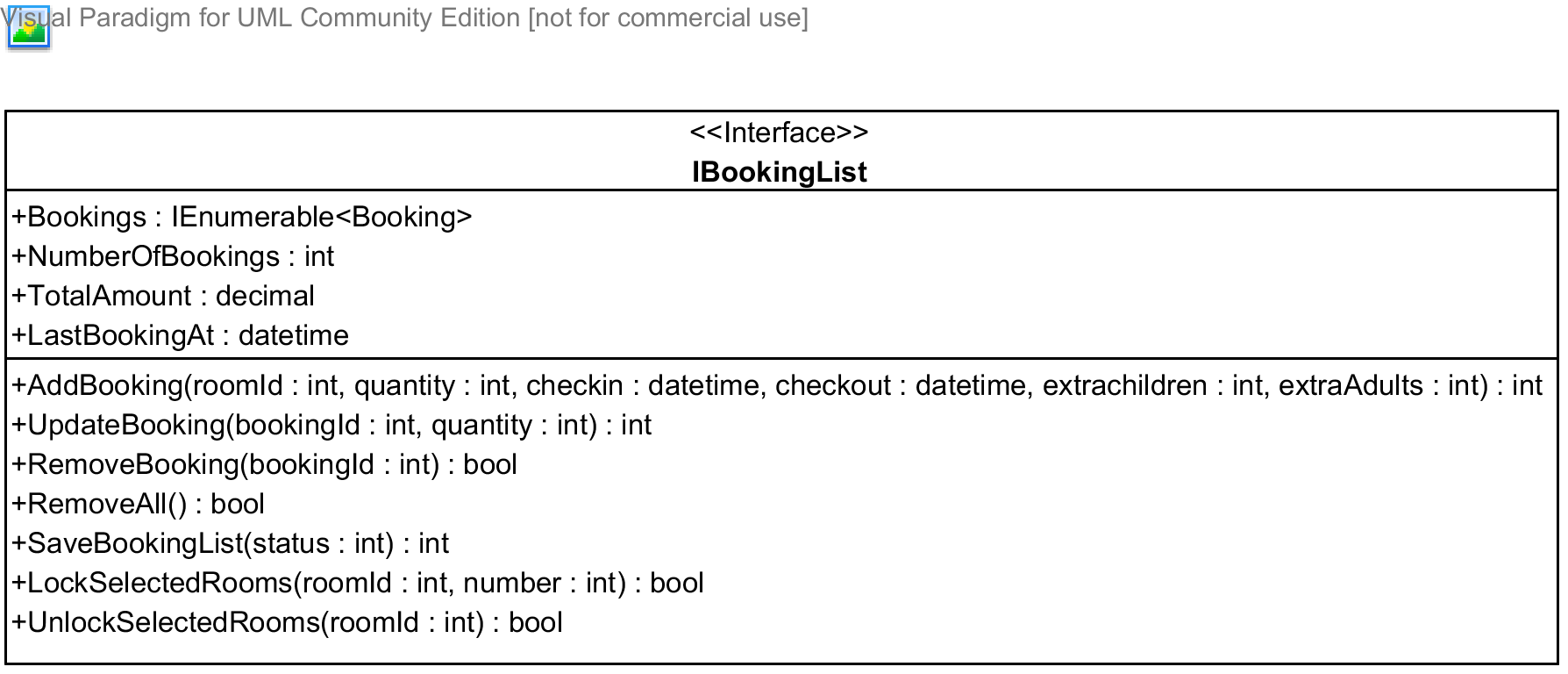
**Properties:** N/A

**Operations**

|  |  |  |
| --- | --- | --- |
| Signature: bool SendEmail(string to, string subject, string content)  Description: Sends an email to the address specified | | |
| Parameter | **Type** | **Description** |
| *to* | string | The email address to send to |
| *subject* | string | The subject of the email |
| *content* | string | The content of the email |

|  |  |  |
| --- | --- | --- |
| Signature: bool SendSMS(string to, string content)  Description: Sends an SMS message to the address specified | | |
| Parameter | **Type** | **Description** |
| *to* | string | The telephone number to send to |
| *content* | string | The content of the message |

#### IStationSuggestion Interface



**Properties**

|  |  |  |  |
| --- | --- | --- | --- |
| Property name | Type | Visibility | Description |
| *Bookings* | IEnumerable<Booking> | public | A list of all bookings in the booking list for the current user |
| *NumberOfBooking* | int | public | Number of bookings in the booking list |
| *TotalAmount* | decimal | public | Total amount of all the booking |
| *LastBookingAt* | datetime | public | Time when the last booking was added to the booking list |

**Operations**

|  |  |  |
| --- | --- | --- |
| Signature: int AddBooking(int roomId, int quantity, datetime checkin, datetime checkout, int extraChildren, int extraAdults)  Description: Adds a bookings made by the current user to the booking list | | |
| Parameter | **Type** | **Description** |
| *roomId* | int | Id of room type being booked |
| *quantity* | int | Numbers of rooms needed |
| *checkin* | datetime | Check-in date |
| *checkout* | datetime | Check-out date |
| *extraChildren* | int | Number of extra children, if allowed by the hotel owner |
| *extraAdults* | int | Number of extra adults, if allowed by the hotel owner |

|  |  |  |
| --- | --- | --- |
| Signature: int UpdateBooking(int bookingId, int quantity)  Description: Update a booking in the booking list given its id | | |
| Parameter | **Type** | **Description** |
| *bookingId* | int | Id of the booking to be updated |
| *quantity* | int | Number of rooms to be updated |

|  |  |  |
| --- | --- | --- |
| Signature: bool RemoveBooking(int bookingId)  Description: Removed a booking in the booking list given its id | | |
| Parameter | **Type** | **Description** |
| *bookingId* | int | Id of the booking to be removed |

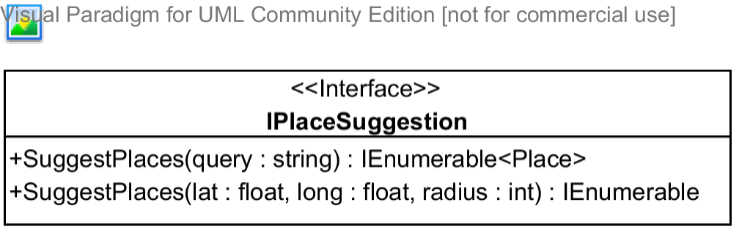
|  |  |  |
| --- | --- | --- |
| Signature: bool RemoveAll(int bookingId)  Description: Removed all the bookings int the booking list | | |
| Parameter | **Type** | **Description** |

|  |  |  |
| --- | --- | --- |
| Signature: bool SaveBookingList(int status)  Description: Saves the booking list to the database | | |
| Parameter | **Type** | **Description** |
| *status* | int | Status of the bookings to be saved |

|  |  |  |
| --- | --- | --- |
| Signature: bool LockSelectedRooms(int roomId, int number)  Description: Locks the rooms selected so that other user cannot book these rooms | | |
| Parameter | **Type** | **Description** |
| *roomId* | int | Id of the room type to be locked |
| *number* | int | Number of rooms to be locked for this type |

|  |  |  |
| --- | --- | --- |
| Signature: bool UnlockSelectedRooms(int roomId)  Description: Unlocks the rooms selected so that other user can book these rooms | | |
| Parameter | **Type** | **Description** |
| *roomId* | int | Id of the room type to be unlocked |

#### IPriceSuggestion Interface



**Properties:** N/A

**Operations**

|  |  |  |
| --- | --- | --- |
| Signature: IEnumerable<Place> SuggestPlaces(string query)  Description: Suggests places(hotel address, provinces, and other type of places,...) given a query string | | |
| Parameter | **Type** | **Description** |
| *query* | string | A string that will be used to search and suggest places |

|  |  |  |
| --- | --- | --- |
| Signature: IEnumerable<Place> SuggestPlaces(float lat, float long, int radius)  Description: Suggests places(hotel address, provinces, and other type of places,...) given the center of the destination and the radius to be searched | | |
| Parameter | **Type** | **Description** |
| *lat* | float | Latitude of the destination center |
| *long* | float | Longitude of the destination center |
| *radius* | int | Radius to limit the searching |

#### IPackageList Interface

****

**Properties:** N/A

**Operations**

|  |  |  |
| --- | --- | --- |
| Signature: bool AddPageTabToHotel(string pageId, int hotelId)  Description: Adds a page tab to the hotel | | |
| Parameter | **Type** | **Description** |
| *pageId* | string | Id of the Facebook page to be added to the hotel |
| *hotelId* | int | Id of the hotel to be added |

|  |  |  |
| --- | --- | --- |
| Signature: Hotel RetrieveHotelForPage(string pageId)  Description: Retrieves a hotel given its pageId | | |
| Parameter | **Type** | **Description** |
| *pageId* | string | Facebook page id of the hotel to be retrieved |

|  |  |  |
| --- | --- | --- |
| Signature: bool RemovePage(int hotelId)  Description: Removes the Facebook page from the hotel given its id | | |
| Parameter | **Type** | **Description** |
| *hotelId* | int | Id of the hotel to remove the Facebook page |

#### IScheduling Interface

****

**Properties:** N/A

**Operations**

|  |  |  |
| --- | --- | --- |
| Signature: bool AddPageTabToHotel(string pageId, int hotelId)  Description: Adds a page tab to the hotel | | |
| Parameter | **Type** | **Description** |
| *pageId* | string | Id of the Facebook page to be added to the hotel |
| *hotelId* | int | Id of the hotel to be added |

|  |  |  |
| --- | --- | --- |
| Signature: Hotel RetrieveHotelForPage(string pageId)  Description: Retrieves a hotel given its pageId | | |
| Parameter | **Type** | **Description** |
| *pageId* | string | Facebook page id of the hotel to be retrieved |

|  |  |  |
| --- | --- | --- |
| Signature: bool RemovePage(int hotelId)  Description: Removes the Facebook page from the hotel given its id | | |
| Parameter | **Type** | **Description** |
| *hotelId* | int | Id of the hotel to remove the Facebook page |

## Sequence Diagrams



### State Machine Diagram: Request Status Transition

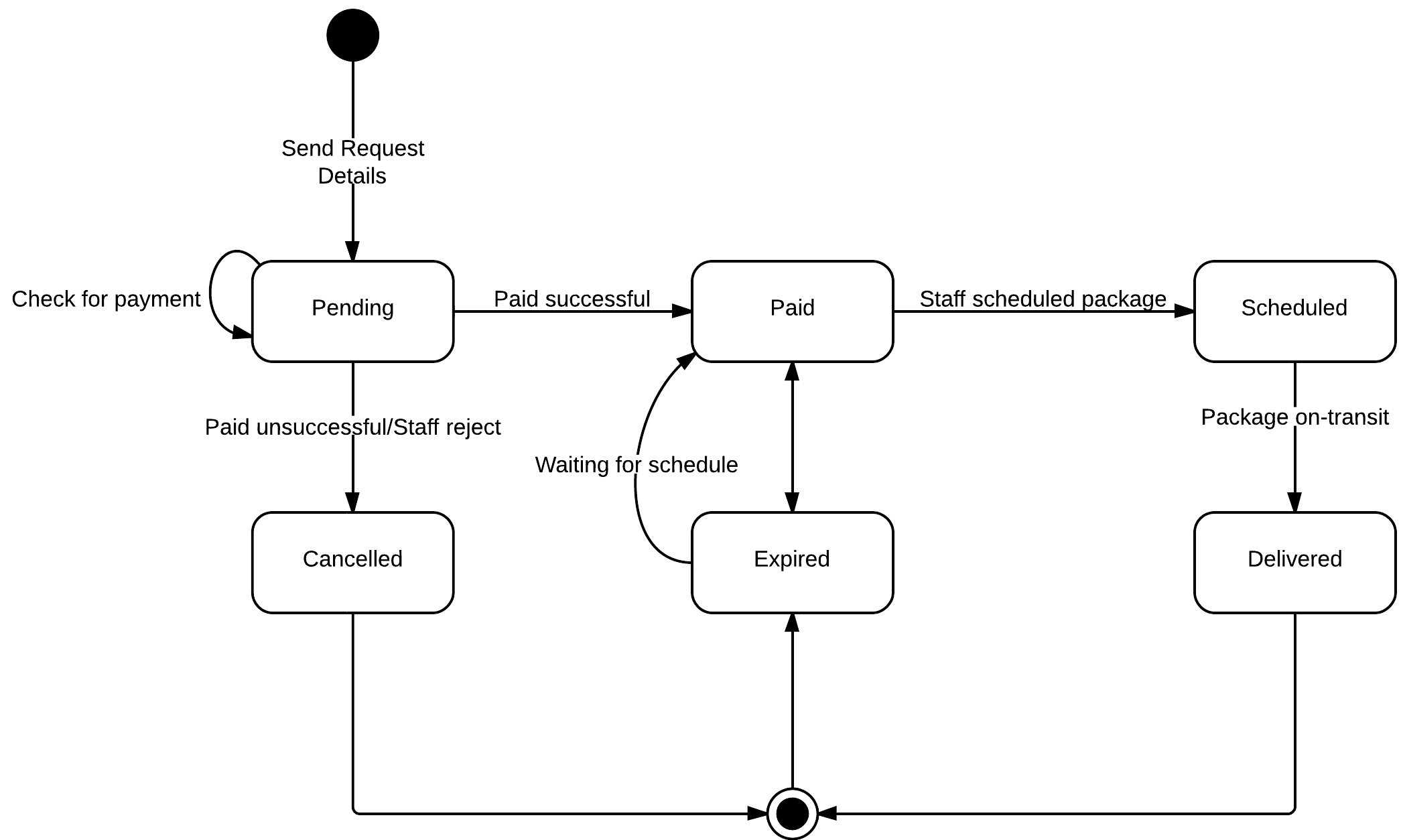


Diagram 5 State machine diagram: Requst status transition

### State Machine Diagram: Package Status Transition

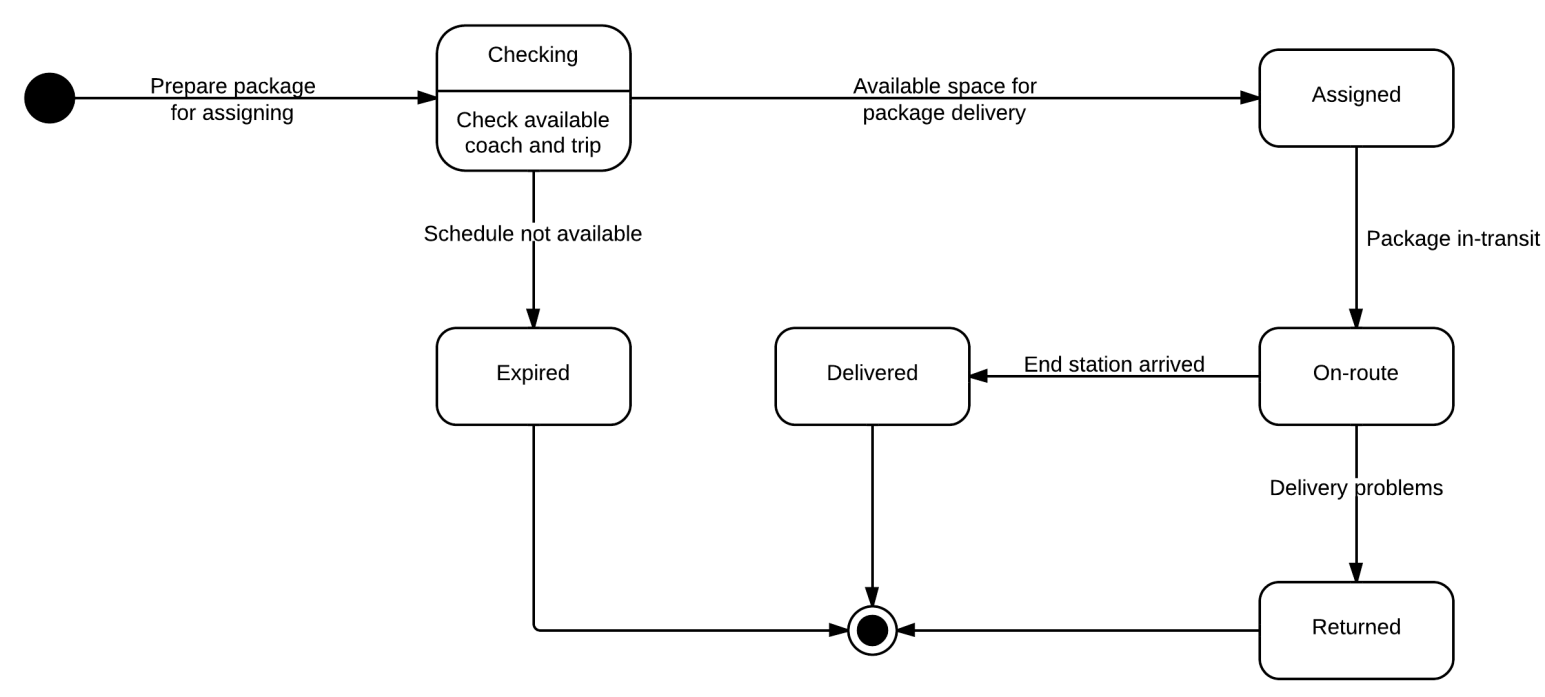


Diagram 6 State machine diagram: Package status transition

### Sequence Diagram: Post Delivery Requests

Diagram 7 Sequence diagram: Post delivery requests

### Sequence Diagram: Search Routes

Diagram 8 Sequence diagram: Search Routes

### Sequence Diagram: Create a Trip

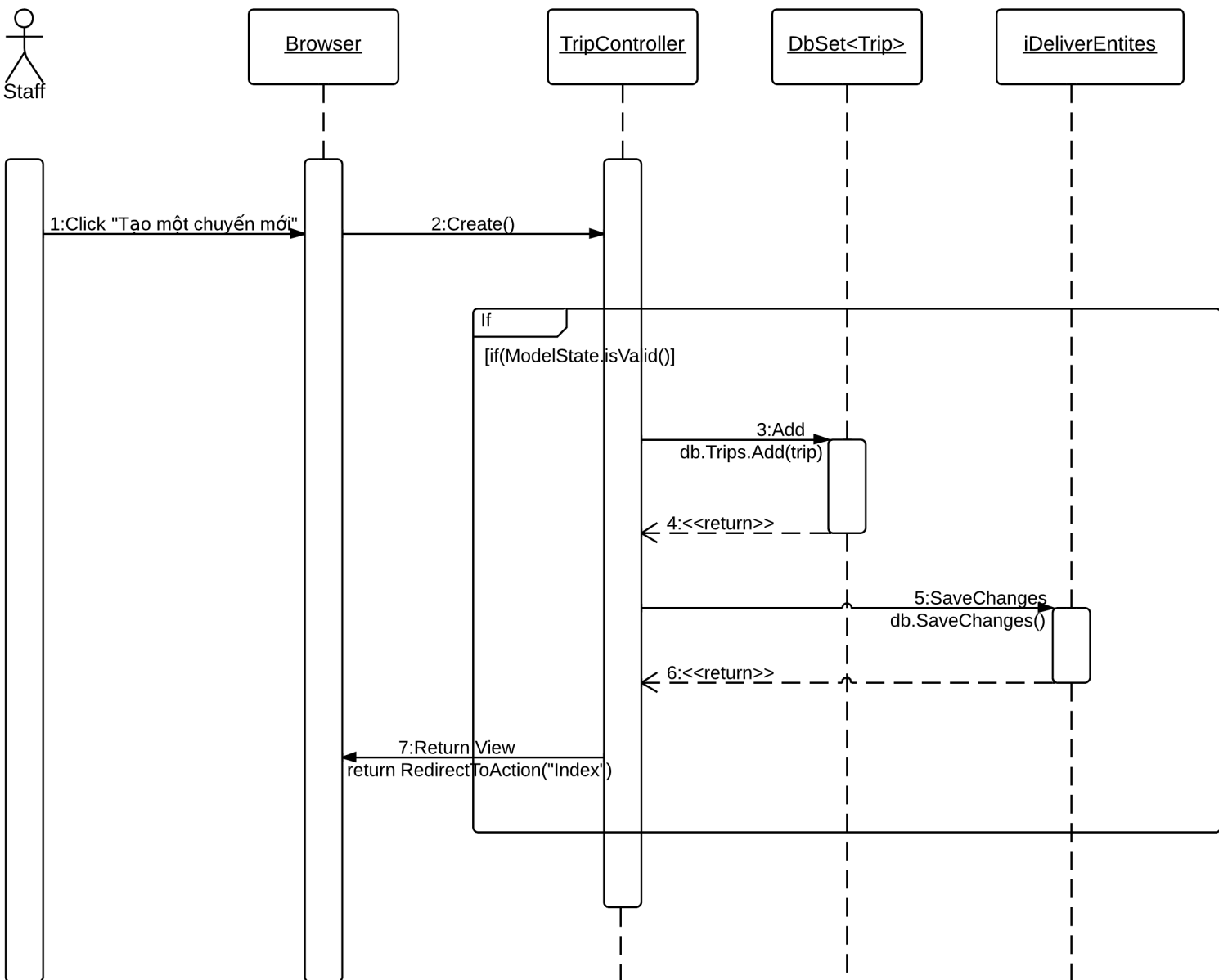


Diagram 9 Sequence diagram: Create a trip

### Sequence Diagram: Edit a Station

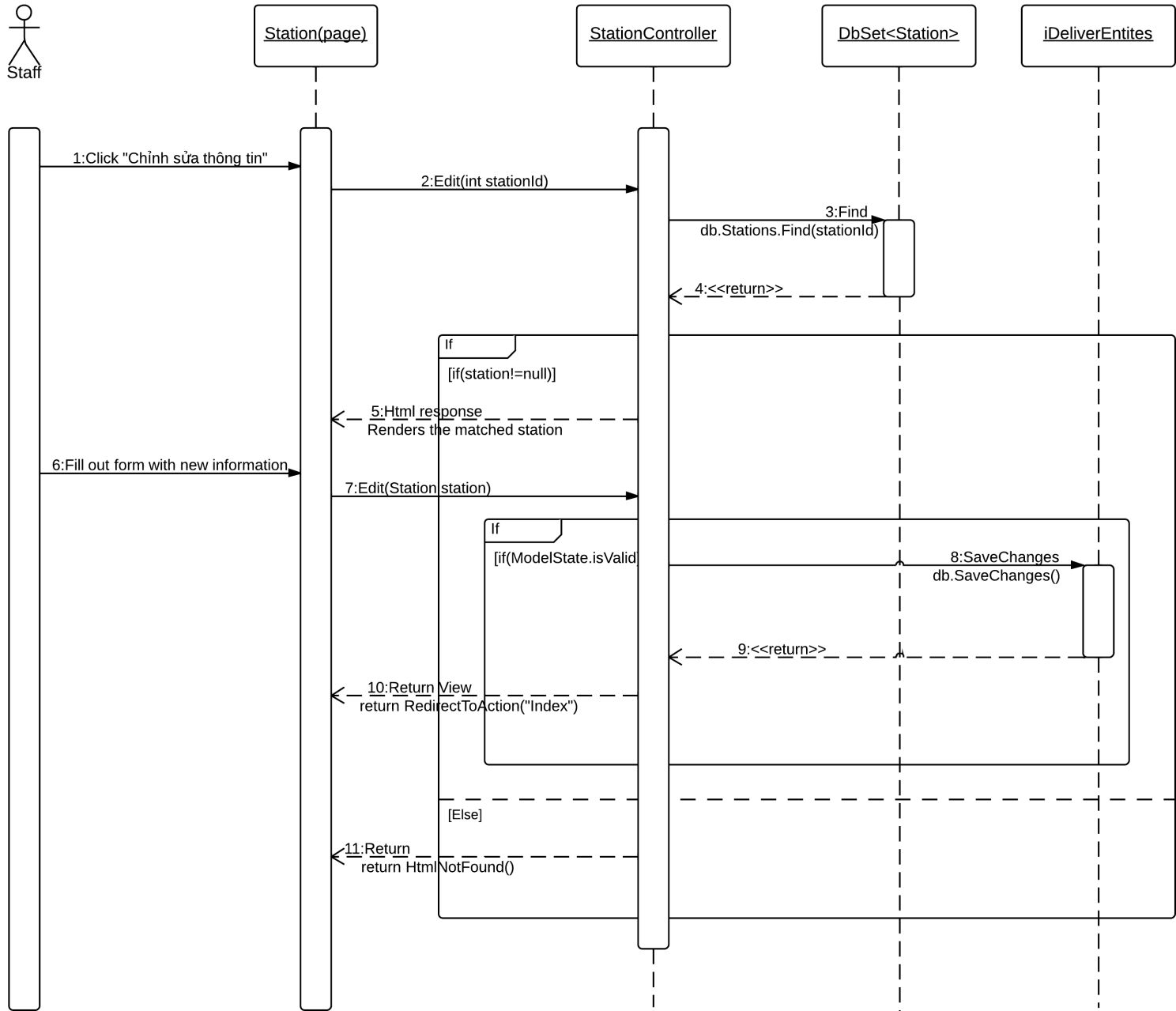


Diagram 10 Sequence diagram: Edit a station

### Sequence Diagram: Add a Price

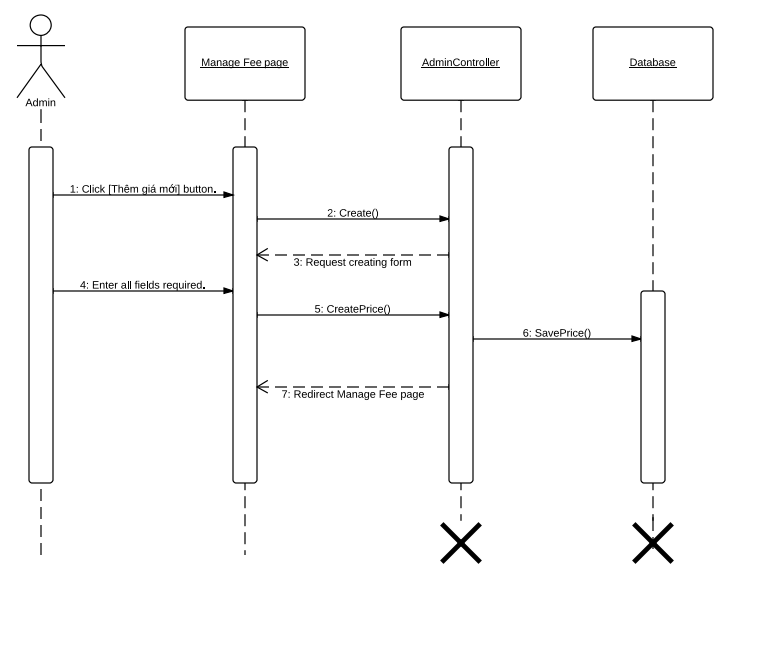


Diagram 11 Sequence diagram: Add a price

### Sequence Diagram: Create a Staff Account

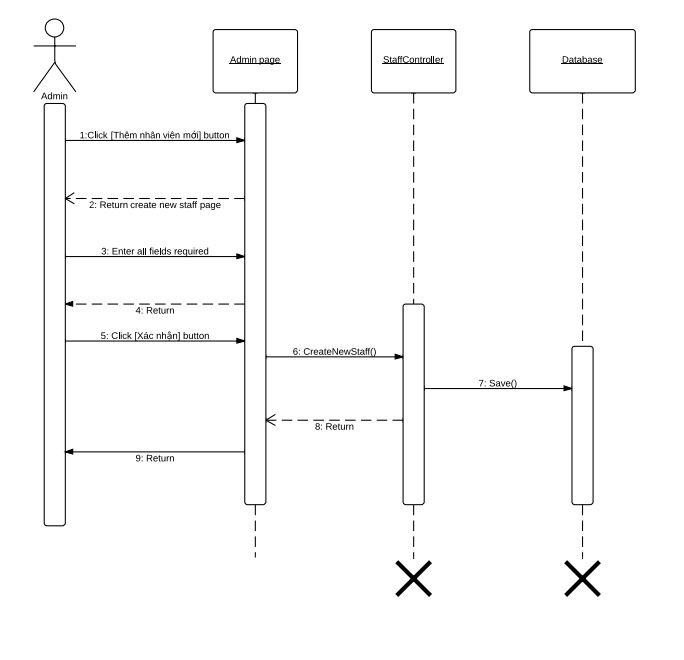


Diagram 12 Sequence diagram: Create a Staff Account

### Sequence Diagram: Approve Requests

Diagram 13 Sequence diagram: Approve requests

### Sequence Diagram: Assign Package

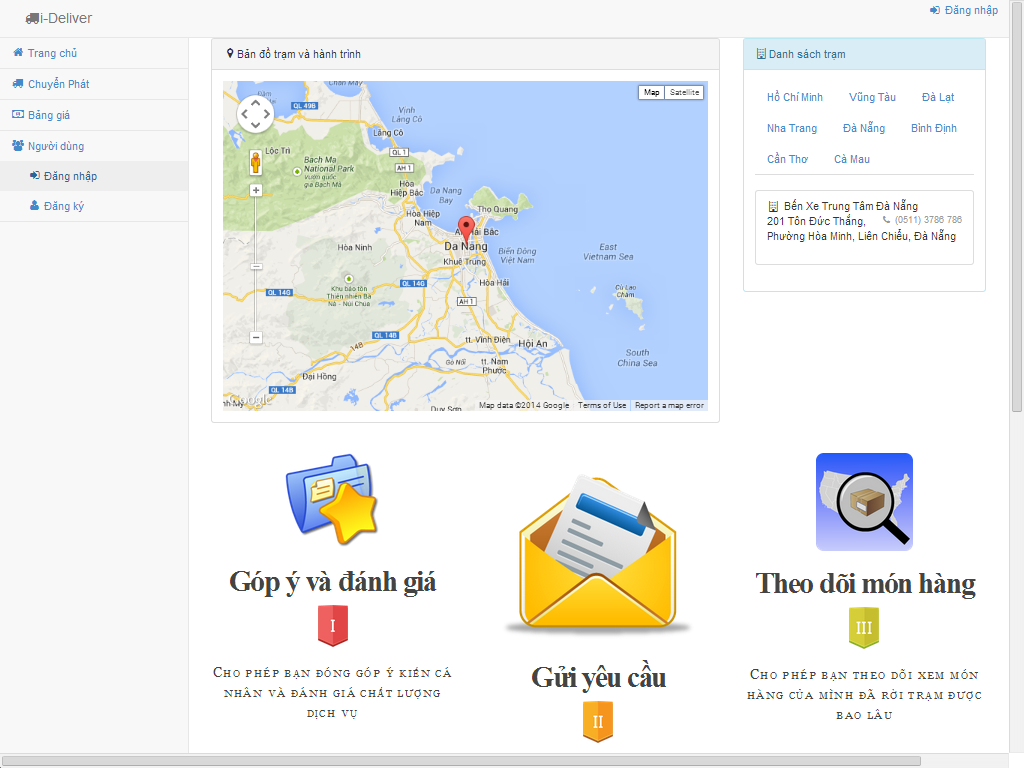
Diagram 14 Sequence diagram: Assign package

## User Interface Design

Below are some important screen designs for the i-Deliver systems.



### Home Page



7

5

4

6

3

2

1

Figure 2 Home page

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| No. | Element | Description | Read Only | Mandatory | Control Type | Data type |
| 1. | “Trang chủ” link | A link to the home page of the hotel information portal | N/A | N/A | Link | N/A |
| 2. | “Chuyển phát” link | A link to the hotel list page | N/A | N/A | Link | N/A |
| 3. | “Bảng giá” link | A link to the price/fee page | N/A | N/A | Button | N/A |
| 4. | “Đăng nhập” button | A button that displays the log in popup | N/A | N/A | Button | N/A |
| 5. | “Đăng ký” button | A button that displays the register popup | N/A | N/A | Button | N/A |
| 6. | “Danh sách trạm” section | Provinces where the has stations | N/A | N/A | N/A | Text |
| 7. | “Bản đồ trạm và hành trình” maps | The marker indicate station location on Google maps | N/A | N/A | N/A | Image |

### Search Routes Page

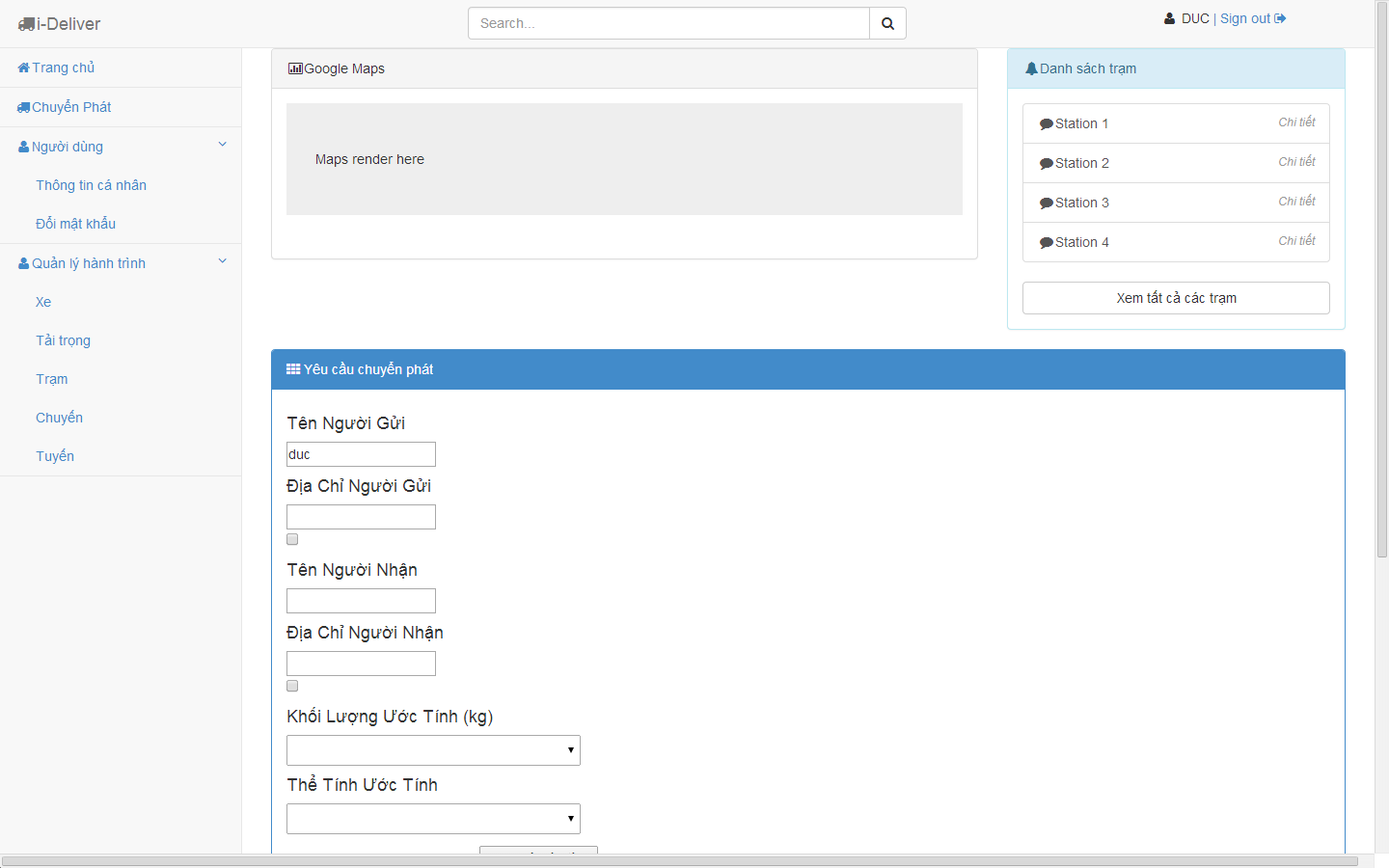
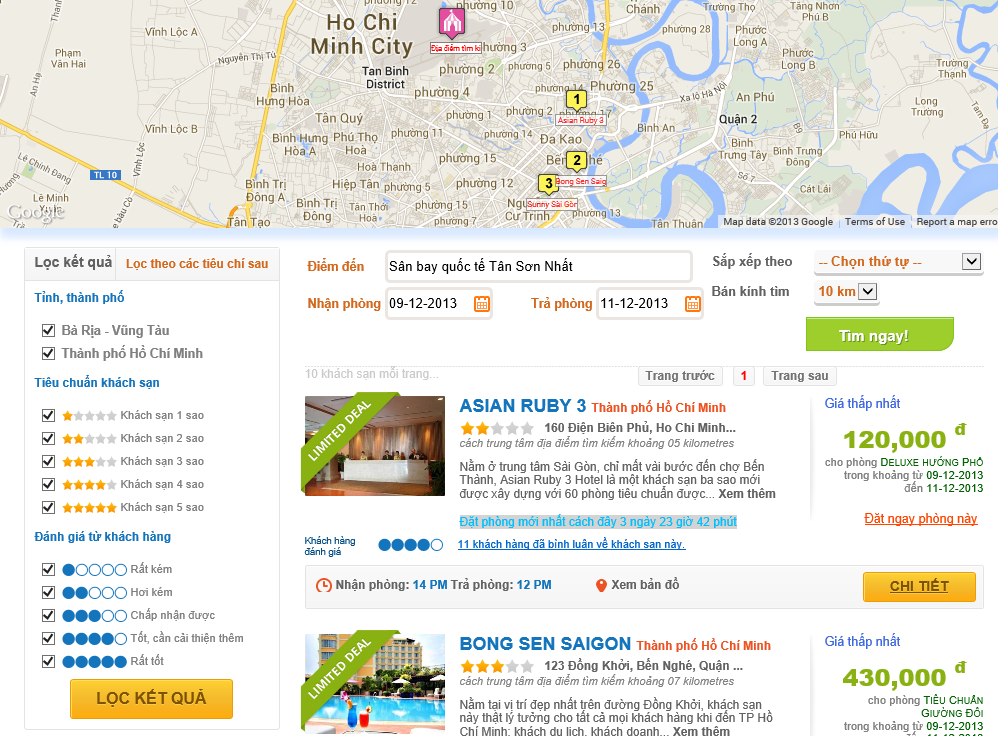


Figure 3 Search Routes Page

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| No. | Element | Description | Read Only | Mandatory | Control Type | Data type |
| 1. | “Điểm đến” textbox | The place where the traveler wishes to go to | No | Yes | Textbox | Text |
| 2. | “Từ ngày” textbox | The date when the traveler wants to check in | No | Yes | Textbox | Text |
| 3. | “Đến ngày” textbox | The date when the traveler wants to check out | No | Yes | Textbox | Text |
| 4. | “Chọn thứ tự” dropdown list | The order of the results | No | No | Dropdown list | N/A |
| 5. | “Bán kính” dropdown list | The radius from the center of the destination | No | No | Button | N/A |
| 6. | “Tìm kiếm” button | A button that searches and displays hotels | N/A | N/A | Button | N/A |
| 7. | “Tỉnh, thành phố” checkboxes | A groups of provinces to filter the results | No | No | Check-boxes | Text |
| 8. | “Tiêu chuẩn khách sạn” checkboxes | A groups of star qualities to filter the results | No | No | Check-boxes | Text |
| 9. | “Đánh giá” checkboxes | A group of user rating averages to filter the results | No | No | Check-boxes | Text |
| 10. | “Lọc kết quả” button | A button that filters the results found | N/A | N/A | Link | N/A |
| 11. | Details of a hotel section | Detailed information of a hotel | N/A | N/A | Div | N/A |
| 12. | Price of a hotel section | Lowest price for a hotel. | N/A | N/A | Div | N/A |
| 13. | Google map section | A Google map that displays all the hotels found | N/A | N/A | Map | N/A |
| 14. | Marker for a destination | A Google marker for the destination the user searches | N/A | N/A | Marker | N/A |
| 15. | Marker for a hotel | A Google marker for a hotel in the result | N/A | N/A | Marker | N/A |



15

14

13

12

11

10

1

4

5

6

9

8

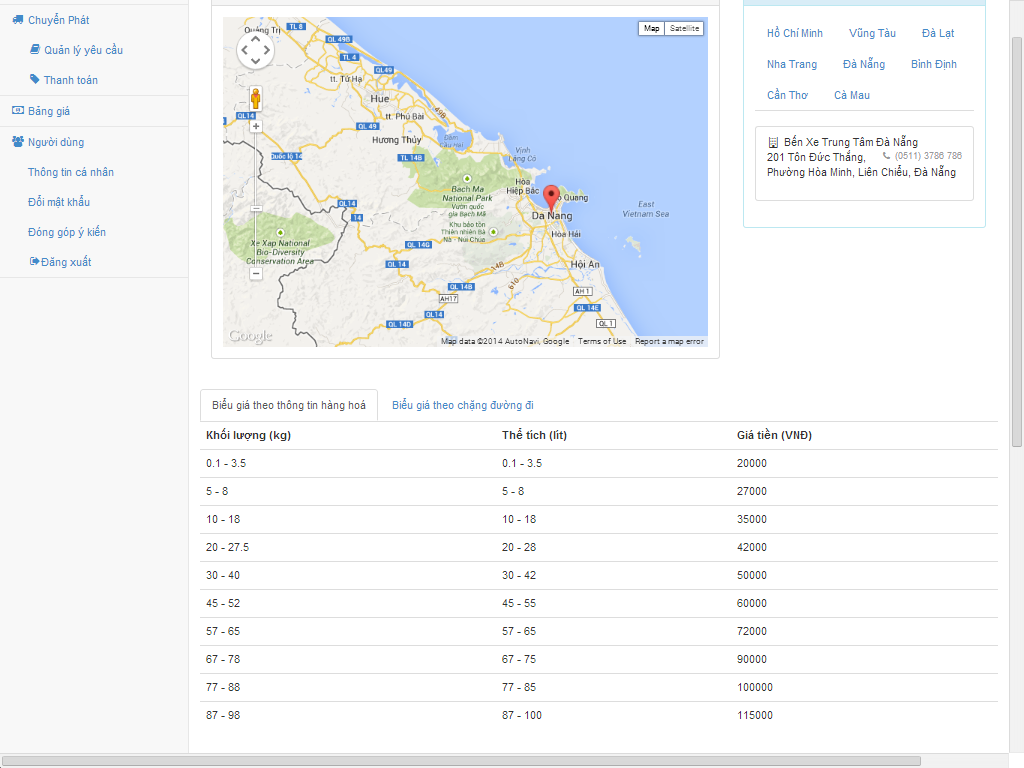
7

3

2

Figure 3 Search Routes Page

### Price Table Page



5

4

3

1

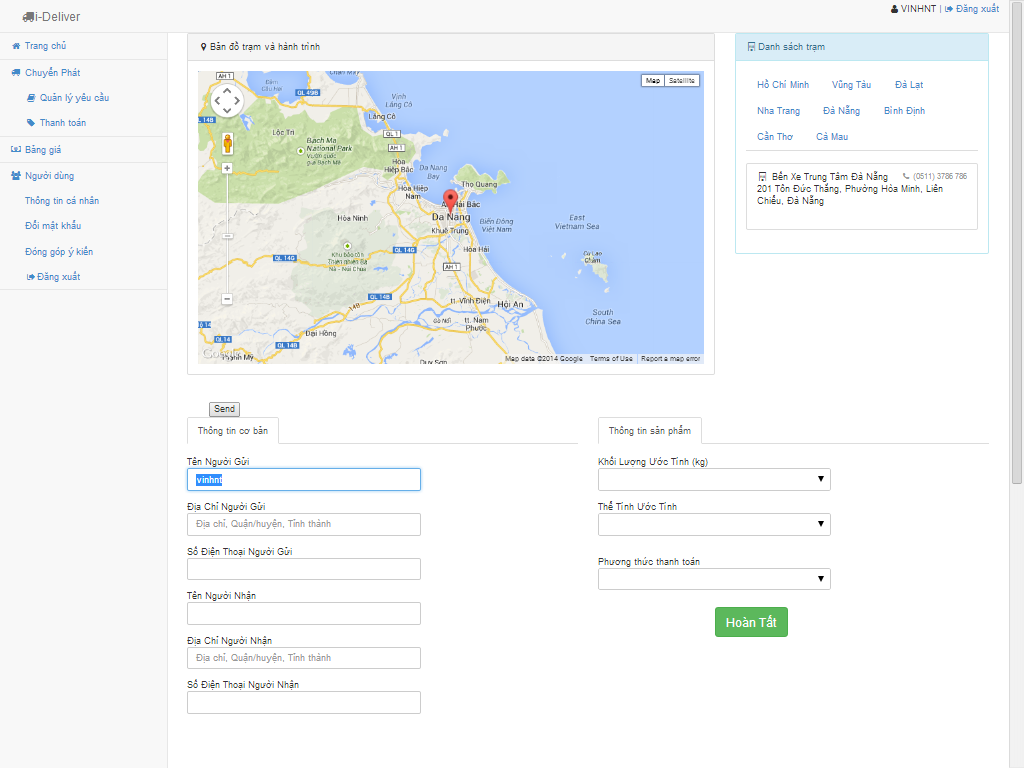
2

Figure 4 Price Table Page

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| No. | Element | Description | Read Only | Mandatory | Control Type | Data type |
| 1. | Navigation/Menu bar | Left-side bar display menu to navigate to other pages | N/A | N/A | Link | N/A |
| 2. | Google Maps | Google maps with markers indicate stations location | N/A | N/A | N/A | N/A |
| 3. | List provinces and list stations | List provinces and stations in that province | Yes | N/A | Link | Text |
| 4. | “Biểu giá theo kích thước hàng hóa” tab | A tab which shows price/fee defined by packages dimension | Yes | N/A | Tab | Text |
| 5. | “Biểu giá theo chặng đường đi” tab | A tab that shows defined price for each routes/stages | Yes | N/A | Tab | Text |

### Sent Request Page

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| No. | Element | Description | Read Only | Mandatory | Control Type | Data type |
| 1. | “Giới tính” dropdown list | Gender of the traveler | No | Yes | Textbox | Text |
| 2. | “Họ và tên” textbox | Full name of the traveler | No | Yes | Textbox | Text |
| 3. | “Địa chỉ” textbox | Address line of the traveler | No | Yes | Textbox | Text |
| 4. | “Ngày tháng năm sinh” textbox | Date of birth of the traveler |  | Yes | Textbox | Text |
| 5. | “Email” textbox | Email address of the traveler | No | Yes | Textbox | Text |
| 6. | “Số điện thoại” textbox | Telephone number of the traveler | No | Yes | Textbox | Text |
| 7. | “Yêu cầu” text area | Any custom request the traveler want to send to the hotel | No | No | Text area | Text |
| 8. | “Đặt thêm phòng” link | A link to go back to the hotel to book more rooms | N/A | N/A | Link | N/A |
| 9. | “Chi tiết đặt phòng” section | The booking detail of the rooms booked | N/A | N/A | Div | N/A |
| 10. | Details button | A button that displays the price details of a booking when it is hovered | N/A | N/A | Button | N/A |
| 11. | Edit booking button | A button to edit the current booking | N/A | N/A | Button | N/A |
| 12. | Delete booking button | A button to remove the current booking | N/A | N/A | Button | N/A |
| 13. | “Thanh toán” section | The detail of total amount | N/A | N/A | Div | N/A |
| 14. | “Thanh toán” button | A button that leads to the payment page | N/A | N/A | Button | N/A |



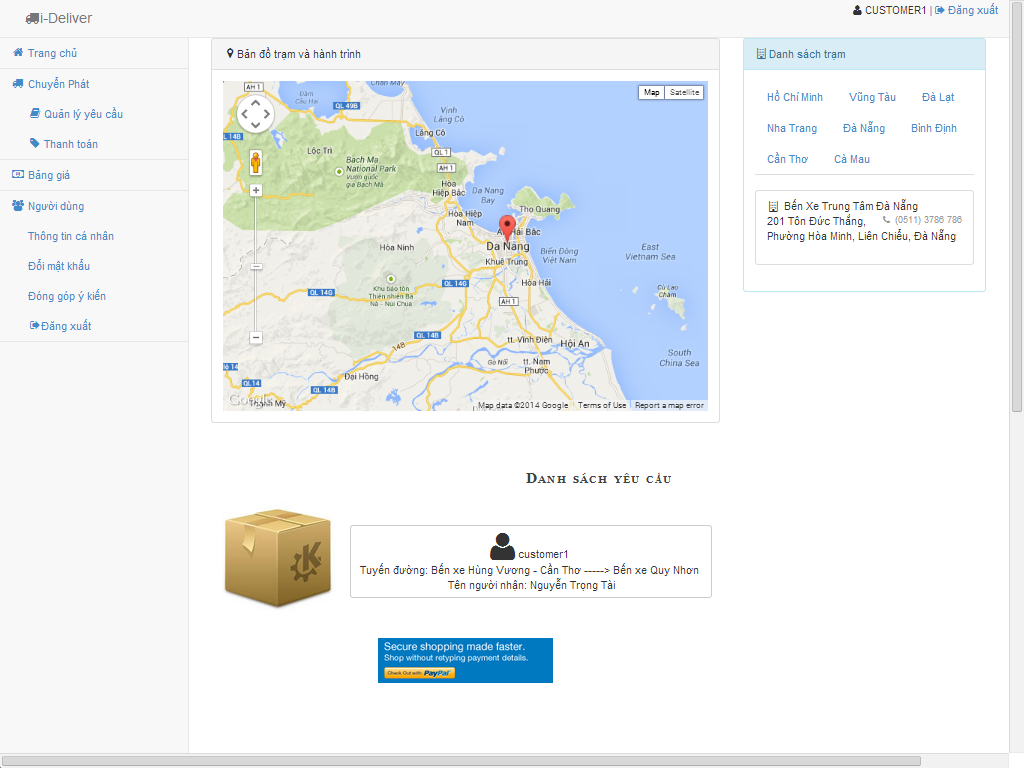
2

1

3

Figure 5 Sent Request Page

### Payment Page



2

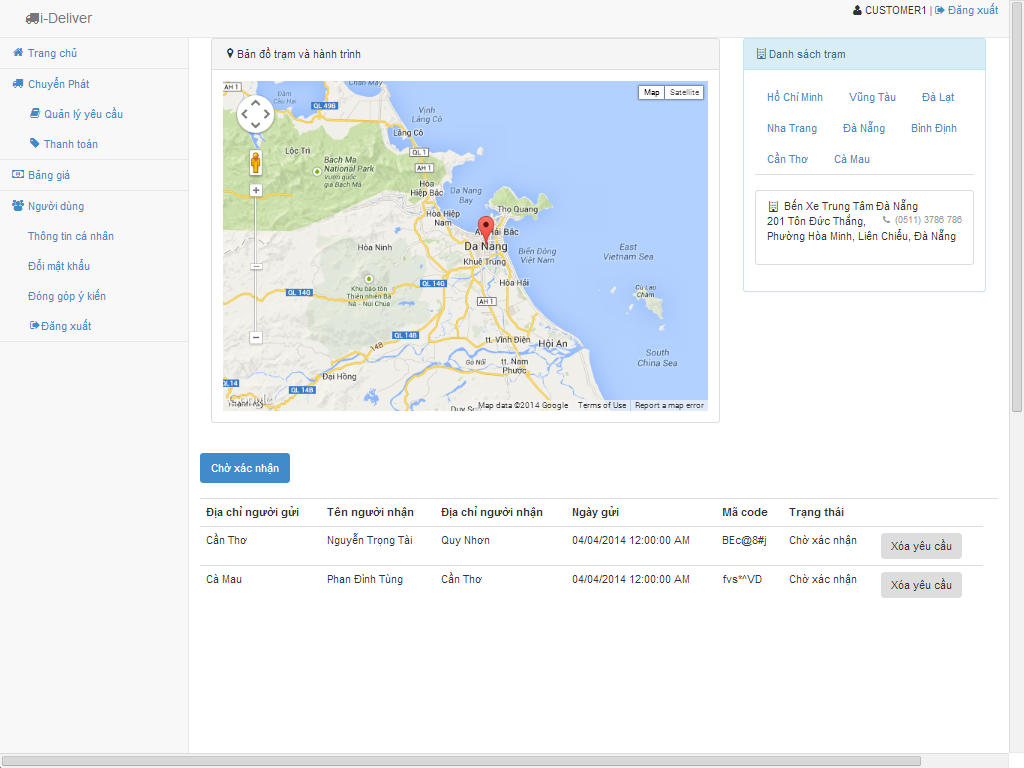
1

3

Figure 6 Payment Page

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| No. | Element | Description | Read Only | Mandatory | Control Type | Data type |
| 1. | Payment information section | The details of the payment made by the customer | N/A | N/A | Div | N/A |
| 2. | Booking code | The booking code generated for the booking made | N/A | N/A | Div | N/A |
| 3. | Customer information | The detailed information of the customer who made the booking | N/A | N/A | Div | N/A |
| 4. | Details of bookings | The details of the booking made by the customer | N/A | N/A | Div | N/A |

### Request Management Page (Customer role)



1

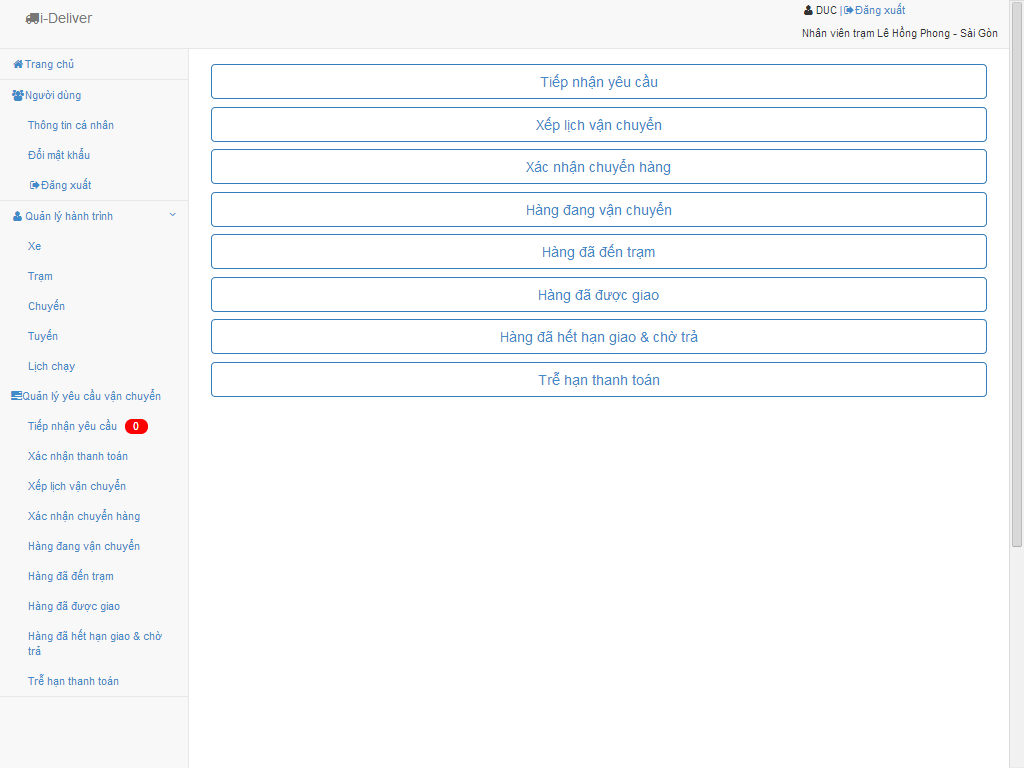
2

3

Figure 7 Request management page (Customer role)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| No. | Element | Description | Read Only | Mandatory | Control Type | Data type |
| 1. | User information section | Information of the logged user | N/A | N/A | Div | N/A |
| 2. | “Khách sạn” link | A link to the dashboard page | N/A | N/A | Link | N/A |
| 3. | “Đặt phòng” link | A link to the booking manangement page | N/A | N/A | Link | N/A |
| 4. | “Thống kê” link | A link to the statistics page | N/A | N/A | Link | N/A |
| 5. | “Kết nối tới Bookee” link | A link to the data synchronization page | N/A | N/A | Link | N/A |
| 6. | New hotel opening requests notification | A notification for new hotel opening request issued by hotel owner | N/A | N/A | Div | N/A |
| 7. | New bookings notification | A notification for new bookings in all the hotels | N/A | N/A | Div | N/A |
| 8. | New bookings notification for a hotel | A notification for new bookings in a specific hotel | N/A | N/A | Div | N/A |
| 9. | “Tất cả” button | A button to list all the hotels | N/A | N/A | Button | N/A |
| 10. | “Tự quản lý” button | A button to filter on local hotels | N/A | N/A | Button | N/A |
| 11. | “i-Deliver quản lý” button | A button to filter on external hotels | N/A | N/A | Button | N/A |
| 12. | “Tên” button | A button to sort the hotels by their names | N/A | N/A | Button | N/A |
| 13. | “Mới cập nhật” button | A button to sort the hotels by their last modified time | N/A | N/A | Button | N/A |
| 14. | “Mới đặt phòng” button | A button to sort the hotels by their last booked time | N/A | N/A | Button | N/A |
| 15. | Hotel list section | A list of all the hotels | N/A | N/A | Div | N/A |
| 16. | “Tạo khách sạn mới” button | A button that redirects to the hotel creating page | N/A | N/A | Button | N/A |
| 17. | Newly booked hotels section | A list of 10 hotel that have the most new bookings. | N/A | N/A | Div | N/A |

### Request Management Page (Staff role)



3

2

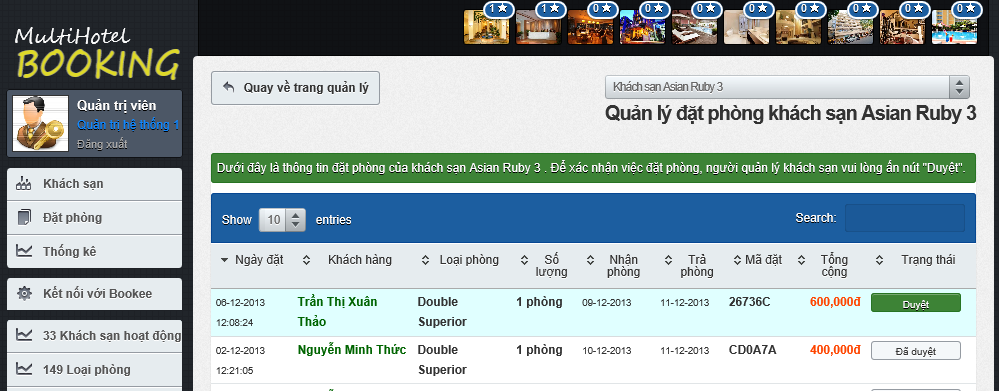
1

4

Figure 8 Request Management Page (Staff role)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| No. | Element | Description | Read Only | Mandatory | Control Type | Data type |
| 1. | Main page navigation | Links to homepage and user profile page | N/A | N/A | Link | Text |
| 2. | “Quản lý hành trình” section | Section with links to journey related page (coach, station, trip, route and schedule) | N/A | N/A | Link | Text |
| 3. | “Quản lý hành trình” section | Section with link to pages specified by request/package delivery status | N/A | N/A | Link | Text |
| 4. | Request section | Sections with links to reques lists group by delivery status | N/A | N/A | Link | Text |

### Administrstor’s Page



3

2

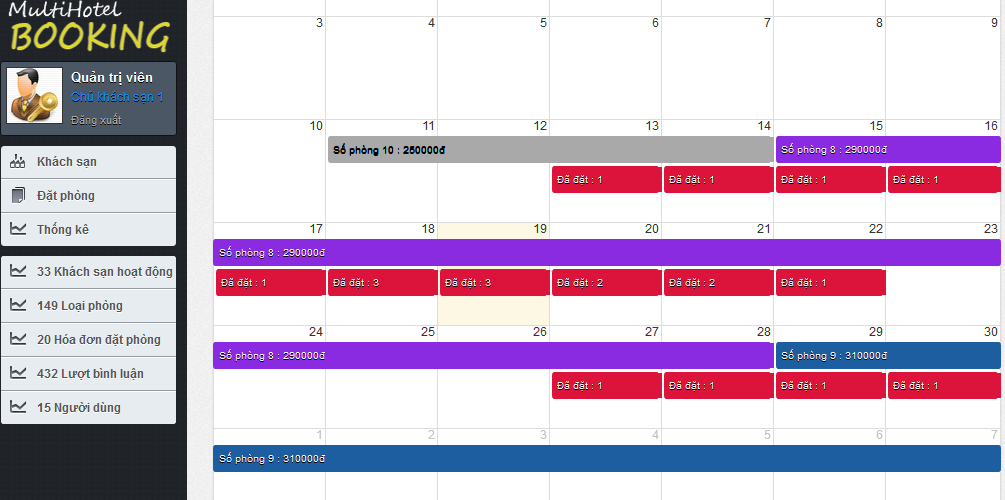
1

Figure 9 Administrators Page

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| No. | Element | Description | Read Only | Mandatory | Control Type | Data type |
| 1. | Booking detail row | Details of a booking | N/A | N/A | Row | N/A |
| 2. | “Duyệt” link | A button to approve of a booking made by a traveler | N/A | N/A | Button | N/A |
| 3. | “Đã duyệt” link | A disabled button that indicates that the booking was approved sucessfully | Yes | N/A | Button | N/A |

### Prices Monitor Page

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| No. | Element | Description | Read Only | Mandatory | Control Type | Data type |
| 1. | Gray price bar | A price bar (in gray) that represents a price in the past. | N/A | N/A | Div | N/A |
| 2. | Violet price bar | A price bar (in violet) that represents a price crossing from the past to the future (including the current system date). | N/A | N/A | Div | N/A |
| 3. | Blue price bar | A price bar (in blue) that represents a price in the future. | N/A | N/A | Div | N/A |
| 4. | Red bar | The number of bookings made on a specific day for the selected room type. | N/A | N/A | Div | N/A |



1

3

2

4

Figure 10 Manage Fee Page

## Database Design



### Logical database design



#### User

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Column Name** | **Type** | **Allow Null** | **Default Value** | **P/F Key** | **Description** |
| Username | int |  | - | PK | Username of user |
| Password | nvarchar(50) |  | - |  | Password of user |
| BackupPassword | nvarchar(50) |  | - |  | Backup password for user |
| Role | nvarchar(50) |  | - |  | Role of user in the system:  customer, admin or staff |
| Email | nvarchar(50) |  | - |  | Email of user |
| Phone | nvarchar(50) |  | - |  | Phone of user |
| BannedStatus | bit |  | - |  | User is banned or not |
| StationID | int |  | 11 | FK | The station of the staff belong to (just for staff account) |

Unique: Username.

Foreign key: StationID.

#### UserInfo

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Column Name** | **Type** | **Allow Null** | **Default Value** | **P/F Key** | **Description** |
| Username | nvarchar(50) |  | - | FK | Username of user registerd to the system or the staff |
| Firstname | nvarchar(50) |  | - |  | Firstname of user registerd to the system or the staff |
| Lastname | nvarchar(50) |  | - |  | Lastname of user registerd to the system or the staff |
| Address | nvarchar(500) |  | - |  | Address of user registerd to the system or the staff |

Unique: Username.

Foreign key: N/A.

#### Coach

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Column Name** | **Type** | **Allow Null** | **Default Value** | **P/F Key** | **Description** |
| CoachID | int |  | Auto generated | PK | Uniquely identifies of the coach |
| NumberPlate | nvarchar(50) |  | - |  | Number plate of coach |
| CoachTypeID | int |  | - | FK | The type of coach |

Unique: CoachID, NumberPlate.

Foreign key: CoachTypeID (table CoachType).

#### CoachType

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Column Name** | **Type** | **Allow Null** | **Default Value** | **P/F Key** | **Description** |
| CoachTypeID | int |  | Auto generated | PK | Uniquely identifies of the coach |
| Seats | int |  | - |  | The number of seats of the coach, that define the type of coach |
| Capacity | float |  |  |  | The percentage of the cabin that coach can be serve. |

Unique: Seats.

Foreign key: N/A.

#### Route

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Column Name** | **Type** | **Allow Null** | **Default Value** | **P/F Key** | **Description** |
| RouteID | int |  | Auto generated | PK | Uniquely identifies a route |
| RouteName | nvarchar(50) |  | - |  | The name of route |

Unique: RouteID.

Foreign key: StartPoint (table Station), EndPoint (table Station).

#### Trip

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Column Name** | **Type** | **Allow Null** | **Default Value** | **P/F Key** | **Description** |
| TripID | int |  | Auto generated | PK | Uniquely identifies a route |
| EstimateDepartureTime | time |  | - |  | The estimate time for the trip departures in each day |
| RealDepartureTime | time |  | - |  | The real time that the trip departures in a specific day |
| EstimateArrivalTime | time |  | - |  | The estimate time for the trip departures in each day |
| RealArrivalTime | float |  |  |  | The real time for the trip arrival to the station in a specific day |
| AvailableVolume | float |  |  |  | The volume that the coach on this trip can deliver |
| Date | date |  |  |  | The date that trip start. |
| RouteID |  |  |  | FK | The route that the trip traveling on |
| CoachID |  |  |  | FK | The coach of this trip. |

Unique: RouteID.

Foreign key: RouteID (table Route), CoachID (table Coach).

#### Station

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Column Name** | **Type** | **Allow Null** | **Default Value** | **P/F Key** | **Description** |
| StationID | int |  | Auto generated | PK | Uniquely identifies a route |
| StationName | nvarchar(50) |  | - |  | The name of the station |
| StationLocation | nvarchar(50 |  | - |  | The address of location |
| BreakTime | float |  | - |  | The default break time that for each trip come to this station |
| StationPhone | nvarchar(50) |  | - |  | The telephone number of this station |
| Description | nvarchar(500) |  | - |  | The desciption for this station |
| Longitude | float |  | - |  | The longitude of the station location |
| Latitude | float |  | - |  | The latitude of the station location |
| Province | nvarchar |  | - |  | The province of the statino location |

Unique: StationID.

Foreign key: N/A.

#### Schedule

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Column Name** | **Type** | **Allow Null** | **Default Value** | **P/F Key** | **Description** |
| ScheduleID | int |  | Auto generated | PK | Uniquely identifies the volume |
| ArrangeID | float |  | - |  | The maximum volume the coach can be deliver on the route |
| EstimateDepartureTime | time |  | - |  | The minimum volume the coach can be deliver on the route |
| EstimateArrivalTime | time |  | - | FK | The route in this volume define |

Unique: N/A.

Foreign key: ArrangeID (table CoachArrangement).

#### Request

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Column Name** | **Type** | **Allow Null** | **Default Value** | **P/F Key** | **Description** |
| ScheduleID | int |  | Auto generated | PK | Uniquely identifies the volume |
| ArrangeID | float |  | - |  | The maximum volume the coach can be deliver on the route |
| EstimateDepartureTime | time |  | - |  | The minimum volume the coach can be deliver on the route |
| EstimateArrivalTime | time |  | - | FK | The route in this volume define |

Unique: N/A.

Foreign key: ArrangeID (table CoachArrangement).

#### DeliveryStatus

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Column Name** | **Type** | **Allow Null** | **Default Value** | **P/F Key** | **Description** |
| DeliveryStatusID | Int |  | Auto generated | PK | Uniquely identifies the delivery status |
| StatusName | nvarchar(50) |  | - |  | The name of delivery status of the package |
| Note | nvarchar(500) | X | - |  | Some more description for the status |

Unique: DeliveryStatusID.

Foreign key: N/A.

#### Invoice

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Column Name** | **Type** | **Allow Null** | **Default Value** | **P/F Key** | **Description** |
| InvoiceID | Int |  | Auto generated | PK | Uniquely identifies a invoice |
| Weight | float | X | - |  | Weight of package |
| Volume | int | X | - |  | Volume of package |
| Price | float |  | - |  | Price of delivery service |
| RequestID | int |  | - | FK | The request that the invoice point to |

Unique: InvoiceID.

Foreign key: RequestID (table Request).

#### ManageFee

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Column Name** | **Type** | **Allow Null** | **Default Value** | **P/F Key** | **Description** |
| FeeID | int |  | Auto generated | PK | Uniquely identifies a fee rule |
| Fee | float |  | - |  | The fee of this weight and volume range |
| MinWeight | float |  | - |  | The minimum weight of the weight range |
| MaxWeight | float |  | - |  | The maximum weight of the weight range |
| MinVolume | float |  | - |  | The minimum volume of the volume range |
| MaxVolume | float |  | - |  | The maximum volume of the volume range |

Unique: FeeID.

Foreign key: N/A.

#### Comment

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Column Name** | **Type** | **Allow Null** | **Default Value** | **P/F Key** | **Description** |
| CommentID | int |  | Auto generated | PK | Uniquely identifies the comment |
| Username | nvarchar(50) |  | - | FK | The user post comment |
| CommentContent | nvarchar(4000) |  | - |  | Content of comment |
| DatePost | date |  | - |  | Date that user post the comment |
| Status | bit |  | - |  | Approve or reject the comment |
| StationID | int |  | - | FK | The station that customer comment for |

Unique: CommentID.

Foreign key: Username (table User), StationID (table Station).

#### Rating

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Column Name** | **Type** | **Allow Null** | **Default Value** | **P/F Key** | **Description** |
| RatingID | int |  | Auto generated | PK | Uniquely identifies the rating |
| Username | nvarchar(50) |  | - | FK | The user post comment |
| RateLevel | int |  | - |  | The level of rate: from 1 to 5 |
| StationID | int |  | - | FK | The station that user rating for |

Unique: RatingID.

Foreign key: Username (table User), StationID (table Station).

#### Assigning

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Column Name** | **Type** | **Allow Null** | **Default Value** | **P/F Key** | **Description** |
| CoachID | int |  | - | PK | The ID of coach for the specific request |
| RequestID | int |  | - | The ID of request |
| IndicateOrder | int |  | - |  | The indicated order of the assigning the package of this request |

Unique: (CoachID, RequestID).

Foreign key: N/A.

#### CoachArragement

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Column Name** | **Type** | **Allow Null** | **Default Value** | **P/F Key** | **Description** |
| CoachID | int |  | - | PK | The ID of coach |
| Route | int |  | - | The ID of route |

Unique: N/A.

Foreign key: N/A.

#### Stage

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Column Name** | **Type** | **Allow Null** | **Default Value** | **P/F Key** | **Description** |
| CoachID | int |  | - | PK | The ID of coach |
| StartPoint | int |  | - | FK | The start position (station) of the stage |
| EndPoint | int |  | - | FK | The end position (station) of the stage |
| Duration | float |  | - |  | The estimate duration (in hours) for moving on this route |
| Distance | float |  | - |  | The estimate distance (in km) between start and end location |
| Container | float |  | - |  | The percentage of coach that allow to travel on the stage |
| Price | float |  | - |  | The fee for traval on the stage |

Unique: N/A.

Foreign key: StartPoint, EndPoint.

#### RouteStage

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Column Name** | **Type** | **Allow Null** | **Default Value** | **P/F Key** | **Description** |
| RouteID | int |  | - | PK | The ID of route |
| StageID | int |  | - | The ID of stage |
| StageIndex | int |  | - |  | The index of stage in the list stage of the route |

Unique: N/A.

Foreign key: N/A.

#### Notification

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Column Name** | **Type** | **Allow Null** | **Default Value** | **P/F Key** | **Description** |
| ID | int |  | **-** | PK | The index of notification |
| Username | Int |  | - |  | Username of user |
| StationID | int |  | - |  | The ID of station the user manges |
| isView | bit |  | - |  | Status of notification, it is seen by user or not. |

Unique: N/A.

Foreign key: StationID, Username.

#### Province

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Column Name** | **Type** | **Allow Null** | **Default Value** | **P/F Key** | **Description** |
| ProvinceID | int |  | **-** | PK | The index of province |
| ProvinceName | nvarchar(50) |  | - |  | Name of province |

Unique: N/A.

Foreign key: N/A.

### Physical database design



Figure 11 Physical database design