|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| |  |  |  | | --- | --- | --- | | **Logo_FPT_University_doc** | **MINISTRY OF EDUCATION AND TRAINING** | | | **FPT UNIVERSITY** | | |
| Capstone Project |
| Information System In Logistic Company  Software Design Description |
| |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | |  |  |  |  | | --- | --- | --- | --- | | **Group 6** | | | | | **Group Members** | Lê Anh Đảo |  | 60142 | | Nguyễn Bá Linh |  | 60153 | | Hồ Hữu Tài |  | 60267 | | Thân Văn Thành  Lê Quang Tú |  | 60277  60037 | | **Supervisor** | Lâm Hữu Khánh Phương | | | | **Capstone Project code** | HDMS | | | | |
| * Ho Chi Minh City, Oct / 2012 - |

# Table of Contents

[Table of Contents 3](#_Toc338283345)

[4.1 Design Overview 4](#_Toc338283346)

[4.2 System Architectural Design 5](#_Toc338283347)

[4.2.1 Choice of System Architecture 5](#_Toc338283348)

[4.2.2 Discussion of Alternative Designs 5](#_Toc338283349)

[4.3 Package Diagram 6](#_Toc338283350)

[4.4 Classes Detail Description 9](#_Toc338283351)

# Design Overview

This document describes the technical and UI design of the Management Information System for TicTac Delivery Company. It includes the architectural design and the detailed design of common functions and business functions. It also includes the design of database model, and overall design of user interface.

The architectural design describes the overall architecture of the system, and the architecture of each main component and subsystem. It will describe the patterns being used, the role of each component and the role of the system in the working environment.

The detailed design describes static and dynamic structure for each component and function. It includes class diagrams, class explanations, and sequence diagrams of the main use cases. The detailed design uses notations of .NET framework 4.0 and C# 4.0, as they are the framework and language for developing the system.

The database design describes the relationship between entities, and details of each entity. It uses notations of SQL Server 2008 R2, as it is the database server for developing the system.

The user interface design describes the layout of the system, and some design for the screens.

# System Architectural Design

## Choice of System Architecture



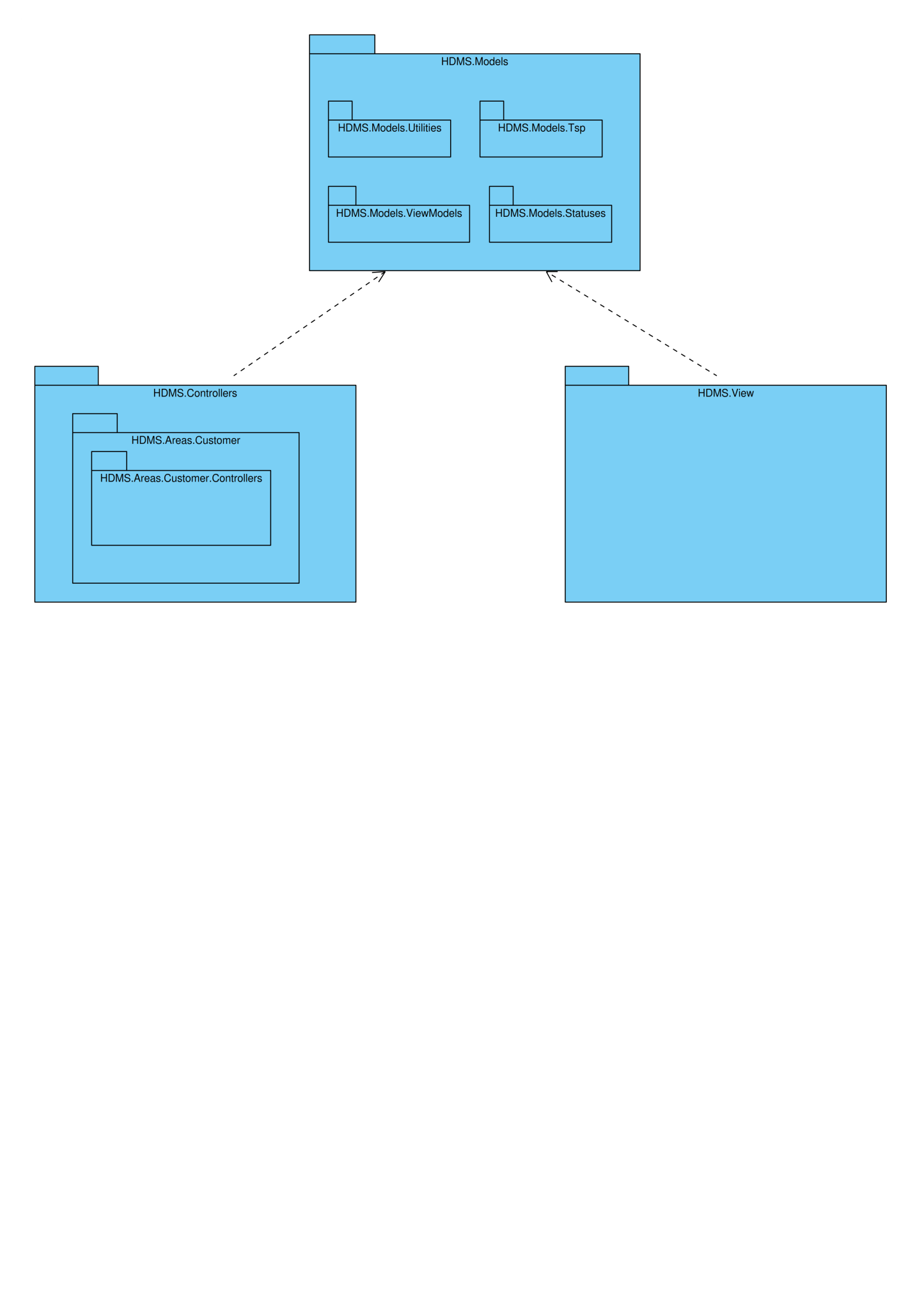
We use ASP.NET MVC 3, which is a framework for building scalable, standards-based web applications using well-established design patterns and the power of ASP.NET and the .NET Framework. MVC stands for model-view-controller. MVC is a pattern for developing applications that are well architected and easy to maintain. MVC-based applications contain:

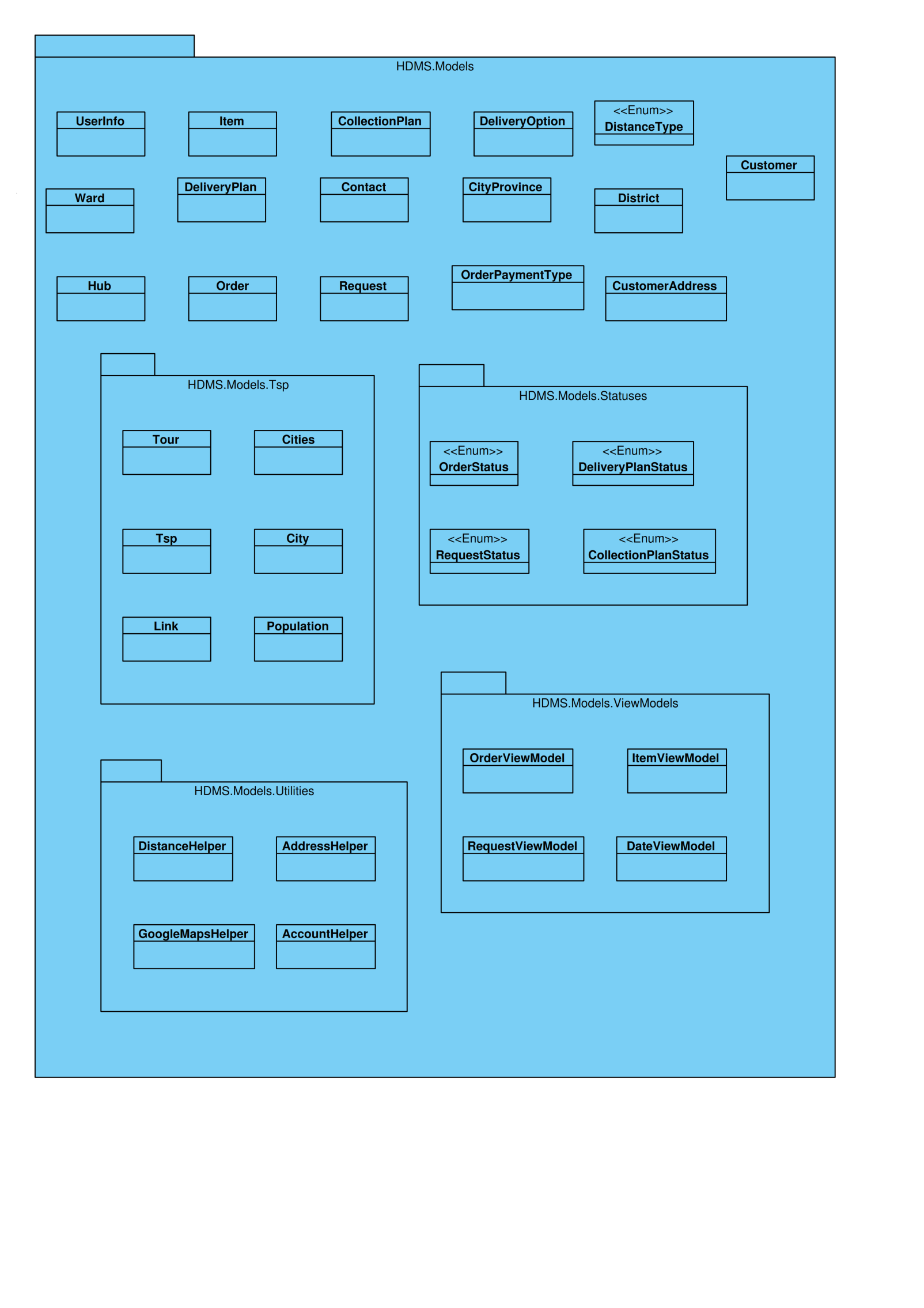
* **View**s are template files that your application uses for dynamically generating HTML responses.
* **Model**s are classes that represent the data of the application and that use validation logic to enforce business rules for that data.
* **Controller**s are classes that handle incoming requests to the application, retrieve model data, and then specify view templates that return a response to the client.

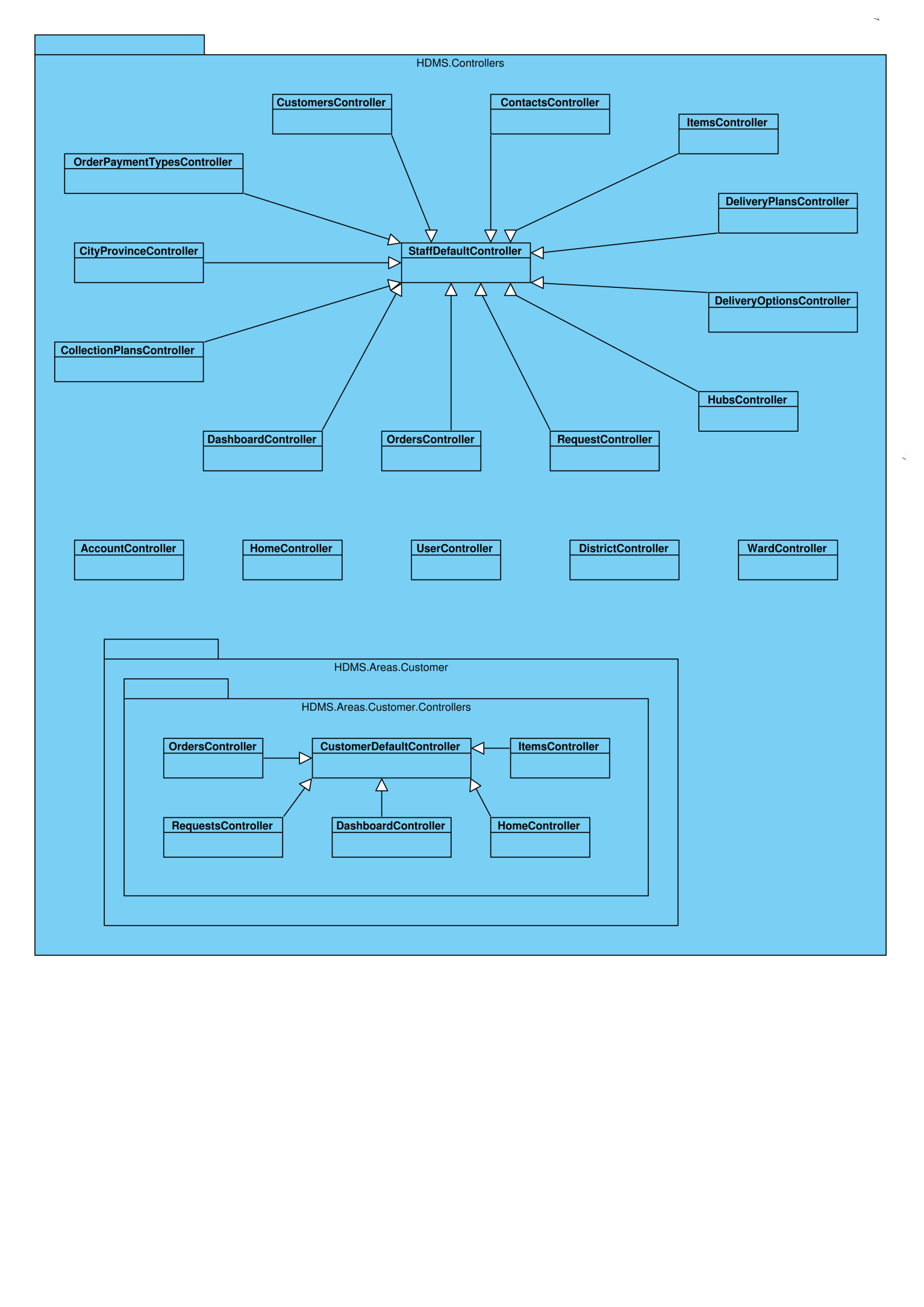
## Discussion of Alternative Designs

At first glance, the three tiers may seem similar to the model-view-controller (MVC) concept; however, topologically they are different. A fundamental rule in three-tier architecture is that the client tier never communicates directly with the data tier; in a three-tier model, all communication must pass through the middle tier. Conceptually the three-tier architecture is linear. However, the MVC architecture is triangular: the view sends updates to the controller, the controller updates the model, and the view gets updated directly from the model.

# Package Diagram







# Classes Detail Description

#### City/Province

|  |  |  |
| --- | --- | --- |
| Field | Type | Description |
| CityProvinceId | int | Unique ID of each City/Province |
| Name | string | Name of City/Province |
| IsActive | bool | Status of City/Province |
| Districts | virtual ICollection<District> | Districts associate with each City/Province |

#### Contact

|  |  |  |
| --- | --- | --- |
| Field | Type | Description |
| ContactId | int | Unique ID of each Contact |
| CustomerId | int | ID of Customer associates with Contact |
| Title | string | Title of Contact person |
| Name | string | Name of Contact person |
| Position | string | Position of Contact person |
| EmailAddress | string | Email address of Contact person |
| PhoneNumber | string | Phone number of Contact person |
| IsActive | bool | Status of Contact |
| Customer | virtual Customer | Customer associates with Contact |

#### Customer

|  |  |  |
| --- | --- | --- |
| Field | Type | Description |
| CustomerId | int | Unique ID of each Customer |
| CompanyName | string | Name of Customer’s company |
| DisplayName | string | A more familiar name of Customer’s company |
| PhoneNumber | string | Phone number of Customer |
| CurrentContractNumber | string | Number of current contract with Customer |
| IsActive | bool | Status of Customer |
| Note | string | Optional information |
| Contacts | virtual ICollection<Contact> | Contacts associate with each Customer |
| CustomerAddresses | virtual ICollection<CustomerAddress> | Customer Addresses associate with each Customer |
| Requests | virtual ICollection<Request> | Requests associate with each Customer |
| UserInfoes | virtual ICollection<UserInfo> | UserInfoes associate with each Customer |

#### Customer Address

|  |  |  |
| --- | --- | --- |
| Field | Type | Description |
| CustomerAddressId | int | Unique ID of each Customer Address |
| CustomerId | int | ID of Customer associates with Customer Address |
| Address | string | Address and street name |
| WardId | Nullable<int> | ID of Ward associate with Customer Address |
| DistrictId | int | ID of district in each Customer Address |
| Latitude | Nullable<decimal> | Latitude of each Customer Address in the map |
| Longitude | Nullable<decimal> | Longitude of each Customer Address in the map |
| IsActive | bool | Status of Customer Address |
| Note | string | Optional information |
| Customer | virtual Customer | Customer associates with Customer Address |
| Ward | virtual Ward | Ward associates with Customer Address |
| District | virtual District | District associates with Customer Address |
| Requests | virtual ICollection<Request> | Requests associate with each Customer Address |

#### Delivery Option

|  |  |  |
| --- | --- | --- |
| Field | Type | Description |
| DeliveryOptionId | int | Unique id of each delivery option |
| Name | string | Name of delivery option |
| Description | string | Describe about delivery option |
| IsActive | bool | Status of delivery option |
| Orders | virtual ICollection<Order> | Order associate with delivery option |

#### District

|  |  |  |
| --- | --- | --- |
| Field | Type | Description |
| DistrictId | int | Unique id of each District |
| CityProvinceId | int | Unique id of each City/Province |
| Name | string | Name of District |
| IsActive | bool | Status of District |
| CityProvince | virtual CityProvince | City/province associate with District |
| CustomerAddresses | virtual ICollection<CustomerAddress> | Customer Address associate with District |
| Hubs | virtual ICollection<Hub> | Hubs associate with District |
| Orders | virtual ICollection<Order> | Order associate with District |
| Wards | virtual ICollection<Ward> | Ward associate with District |

#### Hub

|  |  |  |
| --- | --- | --- |
| Field | Type | Description |
| HubId | int | Unique id of each Hub |
| Name | string | Name of each Hub |
| Address | string | Address and street name of each Hub |
| WardId | Nullable<int> | ID of Ward associate with Hub |
| DistrictId | int | ID of district in each Hub |
| Latitude | Nullable<decimal> | Latitude of each Customer Address in the map |
| Longitude | Nullable<decimal> | Longitude of each Customer Address in the map |
| IsActive | bool | Status of Hub |
| District | virtual District | District associates with Hub |
| Ward | virtual Ward | Ward associates with Hub |

#### Item

|  |  |  |
| --- | --- | --- |
| Field | Type | Description |
| ItemId | int | Unique id of each item |
| OrderId | int | Unique id of each order |
| Name | string | Name of items |
| Quantity | int | Quantity of each item |
| IsFragile | bool | It is fragile or not? |
| HasHighValue | bool | It has high value or not? |
| Size | string | Size of item |
| Weight | string | Weight of item |
| Note | string | Comment about items |
| Order | virtual Order | Order associate with item |

#### Order Payment Type

|  |  |  |
| --- | --- | --- |
| Field | Type | Description |
| OrderPaymentTypeId | int | Unique id of each order payment type |
| Name | string | Name of order payment type |
| Description | string | Describe about order payment type |
| IsActive | bool | Status of order payment type |
| Orders | virtual ICollection<Order> | Order associate with order payment type |

#### User Info

|  |  |  |
| --- | --- | --- |
| Field | Type | Description |
| UserId | System.Guid | Unique id of each User |
| FullName | string | Full name of each user |
| Customer | virtual Customer | Customer associate with UserInfo |

#### Ward

|  |  |  |
| --- | --- | --- |
| Field | Type | Description |
| WardId | int | Unique id of each Ward |
| DistrictId | int | Unique id of each District |
| Name | string | Name of Ward |
| IsActive | bool | Status of Ward |
| CustomerAddresses | virtual ICollection<CustomerAddress> | Customer address associate with Ward |
| District | virtual District | District associate with Ward |
| Hubs | virtual ICollection<Hub> | Hubs associate with Ward |
| Orders | virtual ICollection<Order> | Order associate with Ward |

#### Order

|  |  |  |
| --- | --- | --- |
| Field | Type | Description |
| OrderId | int | Unique id of each order |
| RequestId | int | Id of the request that the order belong to |
| DeliveryOptionId | int | Id of the Delivery Option that apply to this order |
| OrderPaymentTypeId | int | Id of the Order Payment Type that apply to this order |
| DeliveryPlanId | Nullable<int> | Id of the Delivery Plan that the order belong to |
| DueDate | Nullable<DateTime> | Deadline for the order to be delivered |
| CollectedDate | Nullable<DateTime> | Date of collection |
| DeliveryDate | Nullable<DateTime> | Date of delivery |
| ReceiverName | string | Name of the receiver |
| ReceiverPhone | string | Phone number of the receiver |
| ReceiverAddress | string | Address of the receiver |
| ReceiverAddressWardId | Nullable<int> | Id of the ward that the receiver address belong to |
| ReceiverAddressDistrictId | int | Id of the district that the receiver address belong to |
| AmountToBeCollectedFromReceiver | int | The amount to be collected from receiver |
| Latitude | Nullable<int> | Latitude of the receiver address |
| Longitude | Nullable<int> | Longitude of the receiver address |
| Fee | Nullable<int> | Amount of fee to be charged for the order |
| OrderStatus | int | Status Id of the order |
| Note | string | Comment about order |
| Request | virtual CollectionRequest | Request associate with order |
| DeliveryPlan | virtual Delivery | Delivery Plan associate with order |
| DeliveryOption | virtual DeliveryOption | Delivery option associate with order |
| Items | virtual ICollection<Item> | Item associate with order |
| Ward | virtual Ward | Ward associate with order |
| District | virtual District | District associate with order |
| OrderPaymentType | virtual OrderPaymentType | Order payment type associate with order |
| Status | OrderStatus | Status of the order |
| FullAddress | string | Full address of the receiver |
| AddToRequest(HDMSEntities context, Request request) | bool | Add the order to a request |
| Approve(HDMSEntities context, DateTime dueDate, int fee) | bool | Approve and set due date, fee to the order |
| Reject(HDMSEntities context) | bool | Reject order |
| Disapprove(HDMSEntities context) | bool | Disapprove order |
| Cancel(HDMSEntities context) | bool | Cancel order |
| Cancel(HDMSEntities context, bool commit) | bool | Cancel order |
| CustomerCancel(HDMSEntities context) | bool | Cancel order by customer |
| CustomerCancel(HDMSEntities context, bool commit) | bool | Cancel order by customer |
| MarkAsCollected(HDMSEntities context) | bool | Mark order as collected |
| MarkAsCollected(HDMSEntities context, bool commit) | bool | Mark order as collected |
| MarkAsDelivered(HDMSEntities context, List<Order> orders) | static bool | Mark orders as delivered |
| MarkAsDelivered(HDMSEntities context) | bool | Mark order as delivered |
| MarkAsDelivered(HDMSEntities context, bool commit) | bool | Mark order as delivered |
| MarkAsReturned(HDMSEntities context) | bool | Mark order as returned |
| AddToDeliveryPlan(HDMSEntities context, DeliveryPlan deliveryPlan) | bool | Add order to delivery plan |
| AddToDeliveryPlan(HDMSEntities context, DeliveryPlan deliveryPlan, bool commit) | bool | Add order to delivery plan |
| Delete(HDMSEntities context) | bool | Delete order |
| Delete(HDMSEntities context, bool commit) | bool | Delete order |
| AddToDeliveryPlan(HDMSEntities context, DeliveryPlan deliveryPlan, List<Order> orders) | static bool | Add orders to delivery plan |
| AddToDeliveryPlan(HDMSEntities context, DeliveryPlan deliveryPlan, List<Order> orders, bool commit) | static bool | Add orders to delivery plan |
| RemoveFromDeliveryPlan(HDMSEntities context, List<Order> orders) | static bool | Remove orders from delivery plan |
| RemoveFromDeliveryPlan(HDMSEntities context) | bool | Remove order from delivery plan |
| RemoveFromDeliveryPlan(HDMSEntities context, bool commit) | bool | Remove order from delivery plan |
| UpdateLocation(HDMSEntities context, decimal latitude, decimal longitude) | bool | Update location of the receiver address |
| BackToDraft(HDMSEntities context) | bool | Switch order back to draft |
| BackToDraft(HDMSEntities context, bool commit) | bool | Switch order back to draft |

#### Request

|  |  |  |
| --- | --- | --- |
| Field | Type | Description |
| RequestId | int | Unique ID of each Request |
| CreatedByUserId | Guid | Id of the user that create the request |
| CustomerId | int | ID of Customer associates with Collection Request |
| CollectionAddressId | int | ID of address where goods will be collected |
| RequestedDate | Nullable<DateTime> | Date Collection Request was created |
| CollectionPlanId | Nullable<int> | ID of Collection Plan associates with Collection Request |
| RequestStatus | int | Status of the Request |
| Note | string | Optional information |
| CollectionPlan | virtual Collection | Collection plan associates with Collection Request |
| Customer | virtual Customer | Customer associates with Collection Request |
| CustomerAddress | virtual CustomerAddress | Customer Address associates with Collection Request |
| Orders | virtual ICollection<Order> | Orders associate with each Collection Request |
| Status | RequestStatus | Status of the Request |
| CollectionAddress | string | Collection address of the request |
| TotalFee | int | Total fee of orders in the request |
| TotalAmountToBeCollected | int | Total amount to collect of orders in the request |
| NumberOfItems | int | Number of items in the request |
| ValidOrders | List<Order> | Valid orders of the request at a specific time |
| AddToCollectionPlan(HDMSEntities context, CollectionPlan collectionPlan, List<Request> requests) | static bool | Add requests to a collection plan |
| AddToCollectionPlan(HDMSEntities context, CollectionPlan collectionPlan, bool commit) | bool | Add request to a collection plan |
| RemoveFromCollectionPlan(HDMSEntities context, List<Request> requests) | static bool | Remove requests from a collection plan |
| RemoveFromCollectionPlan(HDMSEntities context) | bool | Remove request from a collection plan |
| RemoveFromCollectionPlan(HDMSEntities context, bool commit) | bool | Remove request from a collection plan |
| Create(HDMSEntities context) | bool | Create new request |
| Confirm(HDMSEntities context) | bool | Submit request from processing |
| Approve(HDMSEntities context) | bool | Approve request |
| Reject(HDMSEntities context) | bool | Reject request |
| Cancel(HDMSEntities context) | bool | Cancel request |
| Cancel(HDMSEntities context, bool commit) | bool | Cancel request |
| CustomerCancel(HDMSEntities context) | bool | Cancel request by customer |
| CustomerCancel(HDMSEntities context, bool commit) | bool | Cancel request by customer |
| MarkAsCollected(HDMSEntities context, List<Request> requests) | static bool | Mark requests as collected |
| MarkAsCollected(HDMSEntities context) | bool | Mark request as collected |
| MarkAsCollected(HDMSEntities context, bool commit) | bool | Mark request as collected |
| Delete(HDMSEntities context) | bool | Delete request |
| Delete(HDMSEntities context, bool commit) | bool | Delete request |
| BackToDraft(HDMSEntities context) | bool | Switch request back to draft |
| BackToDraft(HDMSEntities context, bool commit) | bool | Switch request back to draft |

#### CollectionPlan

|  |  |  |
| --- | --- | --- |
| Field | Type | Description |
| CollectionPlanId | int | Unique ID of each Collection Plan |
| CreatedDate | DateTime | Date collection was created |
| CreatedByUserId | Guid | ID of User created Collection |
| AssignedDeliveryStaffId | Nullable<Guid> | ID of assigned staff |
| FinishedDate | Nullable<DateTime> | Finished date |
| CollectionPlanStatus | int | Status id of Collection Plan |
| Note | string | Note |
| Requests | virtual ICollection<Request> | Requests that are included in the Collection Plan |
| CollectionOrder | string | Order of the collection locations |
| Status | CollectionPlanStatus | Current status of collection plan |
| CreatedByUserInfo | UserInfo | User that created the collection plan |
| AssignedUserInfo | UserInfo | User that is assigned to the collection plan |
| CreatedByUser | MembershipUser | User that created the collection plan |
| AssignedUser | MembershipUser | User that is assigned to the collection plan |
| Create(HDMSEntities context, int[] requestIds) | bool | Create new Collection Plan |
| Cancel(HDMSEntities context) | bool | Cancel the collection plan |
| MarkAsFinished(HDMSEntities context, bool removeUncollectedRequests) | bool | Mark the collection plan as finished |

#### Delivery Plan

|  |  |  |
| --- | --- | --- |
| Field | Type | Description |
| DeliveryPlanId | int | Unique ID of each Delivery Plan |
| CreatedDate | DateTime | Date delivery was created |
| CreatedByUserId | Guid | ID of User created Delivery |
| AssignedDeliveryStaffId | Nullable<Guid> | ID of assigned staff |
| FinishedDate | Nullable<DateTime> | Finished date |
| DeliveryPlanStatus | int | Status id of Delivery Plan |
| Note | string | Note |
| Orders | virtual ICollection<Order> | Orders that are included in the Delivery Plan |
| DeliveryOrder | string | Order of the delivery locations |
| Status | DeliveryPlanStatus | Current status of delivery plan |
| CreatedByUserInfo | UserInfo | User that created the delivery plan |
| AssignedUserInfo | UserInfo | User that is assigned to the delivery plan |
| CreatedByUser | MembershipUser | User that created the delivery plan |
| AssignedUser | MembershipUser | User that is assigned to the delivery plan |
| Create(HDMSEntities context, int[] orderIds) | bool | Create new Delivery Plan |
| Cancel(HDMSEntities context) | bool | Cancel the delivery plan |
| MarkAsFinished(HDMSEntities context, bool removeUnfinishedOrders) | bool | Mark the delivery plan as finished |
| CalculateRoute(HDMSEntities context) | void | Calculate the optimal route for the delivery plan |
| CalculateRoute(HDMSEntities context, bool commit) | void | Calculate the optimal route for the delivery plan |
| CalculateOrderList(HDMSEntities context, List<Distance> distances, bool commit) | void | Calculate the optimal order of locations for the delivery plan |

#### Account controller

|  |  |  |
| --- | --- | --- |
| Method | Return type | Description |
| LogOn() | ActionResult | Display a form for login |
| LogOn(LogOnModel model, string returnUrl) | ActionResult | Allow user to login to system |
| LogOff() | ActionResult | Allow user to logout from system |
| Register() | ActionResult | Display a form for register |
| Register(RegisterModel model) | ActionResult | Allow user to create new account |
| ChangePassword() | ActionResult | Display a form for changing password |
| ChangePassword (ChangePasswordModel model) | ActionResult | Allow user to change their password |
| ChangePasswordSuccess() | ActionResult | Display a form to show that password was change successfully. |
| ErrorCodeToString(MembershipCreateStatus createStatus) | ActionResult | Display a message about creating account unsuccessfully. |

#### CityProvince controller

|  |  |  |
| --- | --- | --- |
| Method | Return type | Description |
| Index() | ViewResult | Display all city/province |
| Details(int id) | ViewResult | Display city/province based on City/Province Id |
| Create() | ActionResult | Display form for creating city/ province |
| ConfirmCreate(CityProvince cityprovince) | ActionResult | Add a city/province to Database |
| Edit(int id) | ActionResult | Display form for editing city/ province |
| ConfirmEdit(CityProvince cityprovince) | ActionResult | Save change of a city/province to Database |
| Remove(int id) | ActionResult | Remove a city/province based on City/Province Id |

***CollectionPlans Controller***

|  |  |  |
| --- | --- | --- |
| Method | Return type | Description |
| Index() | ViewResult | Display all collection plans |
| Details(int id) | ViewResult | Display collection plan based on Collection Plan Id |
| Create() | ActionResult | Display form for creating a collection plan |
| ConfirmCreate() | ActionResult | Add a collection plan to database |
| GetUnplannedRequestsByCityProvince(int id, int[] selectedRequestIds) | ActionResult | Get unplanned request by city/ province |
| GetUnplannedRequestsByDistrict(int id, int[] selectedRequestIds) | ActionResult | Get unplanned request by district |
| GetUnplannedRequestsByWard(int id, int[] selectedRequestIds) | ActionResult | Get unplanned request by ward |
| Assign(int collectionPlanId, Guid staffUserId) | ActionResult | Assign collection plan to Staff for collecting |
| Cancel(int id) | ActionResult | Cancel collection plan |
| MarkAsFinished(int id, bool removeUncollectedRequests) | ActionResult | Mark a collection plan as finished |

#### Contact Controller

|  |  |  |
| --- | --- | --- |
| Method | Return type | Description |
| Create() | ActionResult | Display form for creating contact |
| ConfirmCreate(Contact  contact) | ActionResult | Add a contact to Database |
| Edit(int id) | ActionResult | Display form for editing contact |
| ConfirmEdit(Contact  contact) | ActionResult | Save change of a contact to Database |
| Delete(int id) | ActionResult | Display form for deleting contact |
| DeleteConfirmed(int id) | ActionResult | Delete a contact based on Contact Id |
| ToggleActivate(int id) | ActionResult | Change the active status that display on screen. |

#### CustomerAddress controller

|  |  |  |
| --- | --- | --- |
| Method | Return type | Description |
| Details(int id) | ViewResult | Display customer address based on Customer Address Id |
| Create() | ActionResult | Display form for creating customer address |
| ConfirmCreate(CustomerAddress  customeraddress) | ActionResult | Add a customer address to Database |
| Edit(int id) | ActionResult | Display form for editing customer address |
| ConfirmEdit(CustomerAddress  customeraddress) | ActionResult | Save change of a customer address to Database |
| GetAddressesOfCustomer(int id) | ActionResult | Get address of a customer based on Customer Id |
| UpdateLocation(int id, decimal latitude, decimal longitude) | ActionResult | Update location of customer on map. |
| Remove(int id) | ActionResult | Remove an address of customer based on Customer Address Id |

#### CustomerController

|  |  |  |
| --- | --- | --- |
| Method | Return type | Description |
| Index() | ViewResult | Display all customer |
| Details(int id) | ViewResult | Display customer based on Customer Id |
| Create() | ActionResult | Display form for creating customer |
| ConfirmCreate(Customer  customer) | ActionResult | Add a customer to Database |
| Edit(int id) | ActionResult | Display form for editing customer |
| ConfirmEdit(Customer  customer) | ActionResult | Save change of a customer to Database |
| GetAddressTable(int customerId) | ActionResult | Display form for insert customer addresses |
| GetContactTable(int customerId) | ActionResult | Display form for insert customer contacts |
| Remove(int id) | ActionResult | Remove customer based on Customer Id |

***DashboardController***

|  |  |  |
| --- | --- | --- |
| Method | Return type | Description |
| Index() | ViewResult | Display main page |

#### DeliveryOptionsController

|  |  |  |
| --- | --- | --- |
| Method | Return Type | Description |
| Index() | ViewResult | Display list of delivery options |
| Create() | ActionResult | Display form for create new delivery option |
| ConfirmCreate(DeliveryOption deliveryoption) | ActionResult | Add new delivery option to Database |
| Edit(int id) | ActionResult | Display form for edit delivery option information |
| ConfirmEdit(DeliveryOption deliveryoption) | ActionResult | Add new delivery option information to Database |
| Remove(int id) | ActionResult | Remove delivery option based on Delivery Option Id |

***DeliveryPlansController***

|  |  |  |
| --- | --- | --- |
| Method | Return Type | Description |
| Index() | ViewResult | Display all customer |
| Details(int id) | ViewResult | Display customer based on Customer Id |
| PdfDetails(int id) | ActionResult |  |
| Create() | ActionResult | Display form for create new delivery plan |
| ConfirmCreate(DeliveryOption deliveryoption) | ActionResult | Add new delivery plan to Database |
| Assign(int deliveryPlanId, Guid staffUserId) | ActionResult | Assign plan to staff for delivering |
| GetUnplannedOrdersByCityProvince(int id, int[] selectedOrderIds) | ActionResult | Get unplanned orders by city/province |
| GetUnplannedOrdersByDistrict(int id, int[] selectedOrderIds) | ActionResult | Get unplanned orders by district |
| GetUnplannedOrdersByWard(int id, int[] selectedOrderIds) | ActionResult | Get unplanned orders by ward |
| MarkAsFinished(int id, bool removeUnfinishedOrders) | ActionResult | Mark a collection plan as finished |
| Cancel(int id) | ActionResult | Cancel collection plan |
| RecalculateRoute(int id) | ActionResult | Recalculate the route for delivery plan |

#### DistrictController

|  |  |  |
| --- | --- | --- |
| Method | Return Type | Description |
| Index() | ViewResult | Display list of districts |
| Create() | ActionResult | Display form for create new district |
| ConfirmCreate(District district) | ActionResult | Add new district to Database |
| Edit(int id) | ActionResult | Display form for edit district information |
| ConfirmEdit(District district) | ActionResult | Add new district information to Database |
| GetDistrictsOfCityProvince(int id) | ActionResult | Use to get districts of city/province based on City/Province Id |
| Remove(int id) | ActionResult | Remove district based on District Id |

***HomeController***

|  |  |  |
| --- | --- | --- |
| Method | Return type | Description |
| Index() | ViewResult | Redirect to appropriate page |

***HubController***

|  |  |  |
| --- | --- | --- |
| Method | Return Type | Description |
| Index() | ActionResult | Display list of hubs |
| Create() | ActionResult | Display form for create new hub |
| ConfirmCreate(Hub hub) | ActionResult | Add new hub to database |
| Edit(int id) | ActionResult | Display form for edit hub information |
| ConfirmEdit(Hub hub) | ActionResult | Add new hub’s information to Database |
| UpdateLocation(int id, decimal latitude, decimal longitude) | ActionResult | Update hub’s location |
| Remove(int id) | ActionResult | Remove hub based on hub Id |

***ItemsController***

|  |  |  |
| --- | --- | --- |
| Method | Return Type | Description |
| Create(int orderId) | ActionResult | Display form for create new item |
| ConfirmCreate(Item item) | ActionResult | Add new item to Database |
| Edit(int id) | ActionResult | Display form for edit |
| ConfirmEdit(Item item) | ActionResult | Update new item’s information to database |
| Cancel(int id) | ActionResult | Cancel this item |

#### OrderPaymentTypesController

|  |  |  |
| --- | --- | --- |
| Method | Return Type | Description |
| Index() | ViewResult | Display list of order payment types |
| Create() | ActionResult | Display form for create new order payment type |
| ConfirmCreate(OrderPaymentType orderpaymenttype) | ActionResult | Add new order payment type to Database |
| Edit(int id) | ActionResult | Display form for edit order payment type information |
| ConfirmEdit(OrderPaymentType orderpaymenttype) | ActionResult | Add new order payment type information to Database |
| Remove(int id) | ActionResult | Display form for delete order payment type information |

#### OrdersController

|  |  |  |
| --- | --- | --- |
| Method | Return Type | Description |
| Index() | ActionResult | Display list of orders |
| Details(int id) | ActionResult | Display detail of an order based on Order Id |
| Create(int requestId) | ActionResult | Display form for create new order |
| ConfirmCreate(Order order) | ActionResult | Add new order to database |
| Edit(int id) | ActionResult | Display form for edit order’s information |
| ConfirmEdit(Order order) | ActionResult | Add new order’s information to Database |
| Cancel(int id) | ActionResult | Cancel this order |
| Approve() | ActionResult | Approve this order |
| Reject(int id) | ActionResult | Reject this order |
| Disapprove(int id) | ActionResult | Disapprove this order |
| MarkAsDelivered() | ActionResult | Mark this order as delivered |
| MarkAsReturned(int id) | ActionResult | Mark this order as returned |
| UpdateLocation(int id, decimal latitude, decimal longitude) | ActionResult | Update location of this order |
| RemoveFromDeliveryPlan(int id) | ActionResult | Remove this order from delivery plan |

***RequestController***

|  |  |  |
| --- | --- | --- |
| Method | Return Type | Description |
| Index() | ViewResult | Display list of requests |
| Details(int id) | ViewResult | Display detail of a request based on Request Id |
| Create(int requestId) | ActionResult | Display form for create new request |
| ConfirmCreate(int CustomerId, int CollectionAddressId, string Note, HttpPostedFileBase excelFile) | ActionResult | Add new request to database |
| AddOrders(int id) | ActionResult | Add Orders to request |
| Submit(int id) | ActionResult | Submit this request |
| RemoveFromDeliveryPlan(int id) | ActionResult | Remove this order from delivery plan |
| GetTotalFee(int id) | ActionResult | Get total fee of this request |
| ApproveOrders(int id) | ActionResult | Approve orders of this request |
| Approve(int id) | ActionResult | Approve this request |
| Reject(int id) | ActionResult | Reject this request |
| Cancel(int id) | ActionResult | Cancel this request |
| MarkAsCollected() | ActionResult | Mark this request as collected |
| UpdateLocation(int id, decimal latitude, decimal longitude) | ActionResult | Update location of this request |

***UserController***

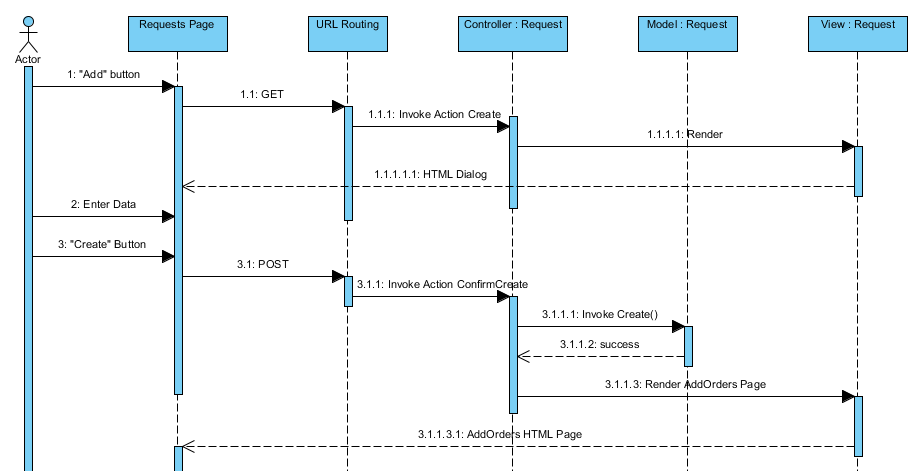
|  |  |  |
| --- | --- | --- |
| Method | Return Type | Description |
| Index() | ActionResult | Display users table |
| LoadUser(string searchUser, string roleName, JQueryDataTableParamModel param, bool showDisabledUsers) | ActionResult | List all users based on parameter or list all active user |
| getRoleUserList(string username) | string | Get roles of users |
| getCustomerName(string username) | string | Get customer name of users |
| ResetPassword(string username) | ActionResult | Reset password of this user |
| Update(string username) | ActionResult | Display form for update user role |
| UpdateUserRole(string username) | ActionResult | Update user role |
| DeleteUser(string username, string rolename) | ActionResult | Disable/enable user |

#### WardController

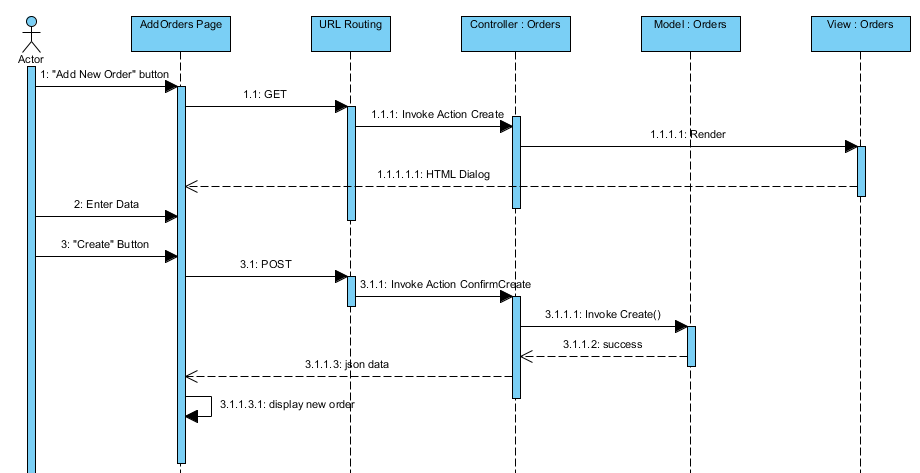
|  |  |  |
| --- | --- | --- |
| Method | Return Type | Description |
| Index() | ViewResult | Display list of wards |
| Create() | ActionResult | Display form for create new ward |
| ConfirmCreate(Ward ward) | ActionResult | Add new ward to Database |
| Edit(int id) | ActionResult | Display form for edit ward information |
| ConfirmEdit(Ward ward) | ActionResult | Add new ward information to Database |
| GetWardsOfDistrict(int id) | ActionResult | Use to get wards of district based on District Id |
| Remove(int id) | ActionResult | Remove this ward |

# Sequence Diagrams

#### Create Request



#### Add New Order



#### Edit Order

#### 

#### Cancel Order

#### Add New Item

#### Approve Request

#### Reject Request

#### Create Collection Plan

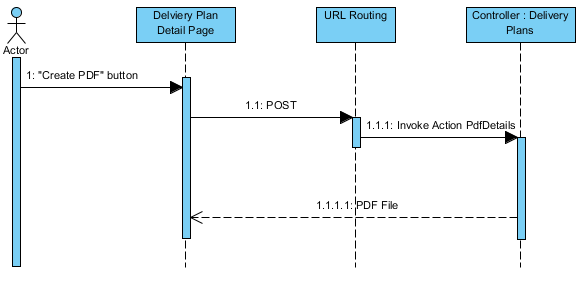
#### Assign Delivery Staff to Collection Plan

#### Mark Collection Plan as Finished

#### Create Delivery Plan

#### Assign Delivery Staff to Delivery Plan

#### Mark Delivery as Finished

Create Delivery Plan PDF

#### Mark Requests As Collected

#### Mark Orders As Delivered

#### Mark Order As Returned

#### Cancel Request

#### Create Customers

#### Delete Customer

#### Edit Customer Information

#### Create Customer Address

#### Delete Customer Address

#### Edit Customer Address

#### Create Contact

#### Delete Contact

#### Edit Contact

#### Create Option

#### Delete Option

#### Edit Option

#### Set Location for Hub

# Databse Design Diagram

# D:\Study\FU\Semester 9\Capstone\HDMS\Design\Database\DB_Design_001.png

# State Diagrams

# Request State Diagram

# D:\Study\FU\Semester 9\Capstone\HDMS\Design\State Diagrams\RequestStateDiagram.png

# Order State Diagram

# D:\Study\FU\Semester 9\Capstone\HDMS\Design\State Diagrams\OrderStateDiagram.png

# Algorithms

To help reduce time and fuel cost, the system should find the best route for delivery mans to deliver all packages in a delivery plan. This problem is similar to a common problem named **Traveling Salesman Problem** (<http://en.wikipedia.org/wiki/Travelling_salesman_problem>).

There are many algorithms to solve this problem. In this project, we use a **Genetic Algorithm** published at <http://www.lalena.com/AI/TSP/>.