System changes:

Transport Module:

Business Layer:

1) We added a new package called bingAPI - now we connect to the maps of bing using API and get the distances and durations between all the sites in the system. This information is updated in the following databases: DeliveryRoutesDAO, SitesRoutesDAO.
2) As a result, to the API connection, we had to open a new controller named: SitesRoutesController, it's role is to take care of everything related to the distances and durations between all the sites.

Data Access Layer:

1) We renamed the object DistanceBetweenSites to SiteRoute. And we added a new field named: duration – it's saves the duration between 2 sites.
2) We added two new databases:
   DeliveryRoutesDAO - His job is to give information about the estimated time of arrival according to transport number and it's destinations .
   SitesRoutesDAO-His job is to keep all the sites in the system and calculate the distance and travel time between 2 sites.
3) We deleted the TransportDestinationsDAO and the object TransportDestinations. There were no longer used because of the addition that occurs in section 2.

Objects:

1) In the DeliveryRoute object we deleted the source field, and instead it was added to the list of destinations. This change was made because it was more convenient in the implementation we used.

Employees Module:

Presentation Layer:

1) We designed the HRManagerMenu and EmployeeMenu to be more user friendly by allowing the user to choose each command before providing each one of its parameters one after the other, instead of a long string of a parameterized command.
2) We added the option to generate the system data from the initial menu.

Business Layer:

1) We changed the shift approval check to also enforce all of the needed employees specified by the HRManager are also assigned to the shift.

Data Access Layer:

1) We made sure all the DAO objects inherit from the same base class and use the same foreign key constraints as the rest of the system.
2) We added a few small record classes to the DAL layer to help with the generic caching in the DAO and to improve the readability of the classes.

General:

1) We reorganized the order of initialization of the system using the ServiceFactory.