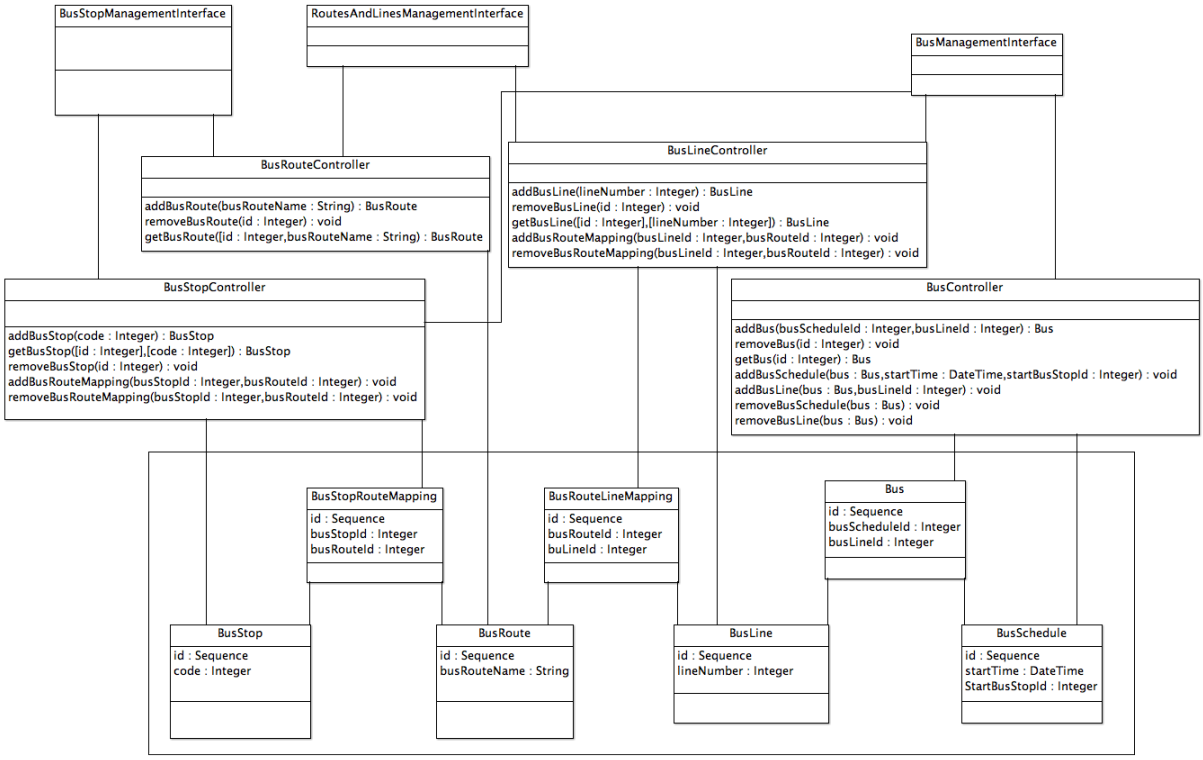
Bus Schedule Management System Design



Functional Requirements

- The system must allow adding and removing buses
- The system must allow defining stops
- The system must allow defining routes
- The system must allow defining lines
- The system must allow assigning a line and a time schedule to a bus

Design summary

I've chosen to supply a UML Class diagram of the Bus Schedule System as I felt it was important to communicate the overall design and decisions behind the system.

The system follows a MVC framework (models at the bottom, views at the top). I envision 3 main views into the system.

The first is the Bus Stop Management Interface. This interface would firstly allow the user to add or remove bus stops. It would also allow the user to search for and allocate existing routes to the bus stop. This means this view interacts with

two distinct controllers; the BusStopController (to create & modify the bus stops – including allocating routes to the stop) and the BusRouteController – to search for and retrieve existing BusRoutes.

The second 'RoutesAndLinesManagement' interface allows the user to create, edit and delete both bus lines and bus routes. It would also allow the user to assign routes to the lines. It's important to note the relationship between bus stops, routes and lines is not 1 to 1. I'll discuss this further below.

The final interface is the 'BusManagementInterface'. This allows the user to create Buses and assign a time schedule (which is driven by the id of the first bus stop and start time) and a bus line.

These three views reflect the logical 'front-end' portals to the system. A number of controllers drive these views and are better represented in the class diagram. The models that represent the various data sources are also shown in the class diagram.

Relationship mapping

The relationship representations in the class diagram exceed the functional requirements but are important as they represent potential expansion of the system. For example, I expect a bus stop to be able to belong to multiple routes. I also expect Bus Lines to potentially consist of multiple routes or bus routes to belong to multiple bus lines. In these cases, I have used mapping tables (and mapping models) to represent these many to many relationships.