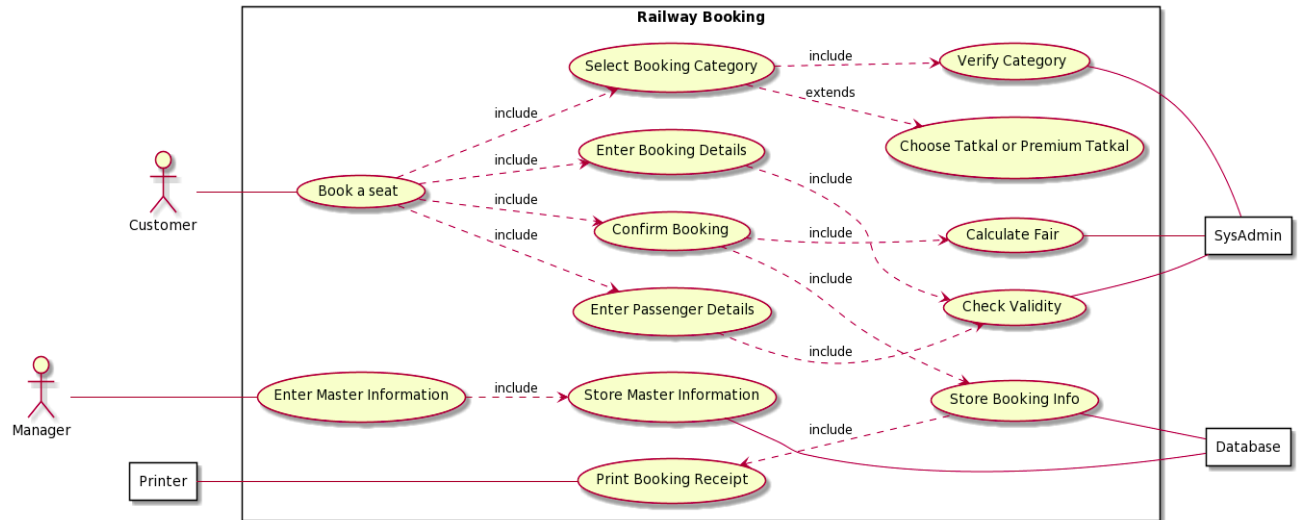


# UML DIAGRAMS

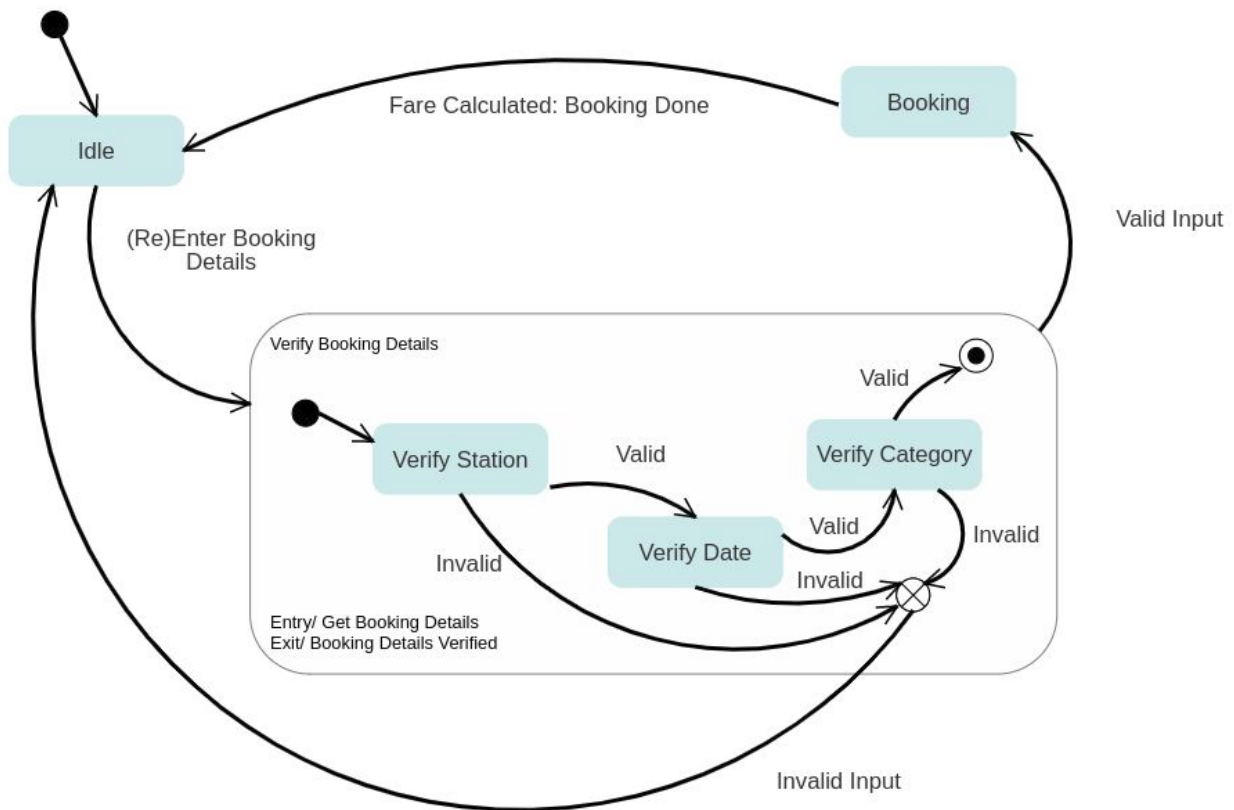
(All the diagrams have been separately attached for better visibility)

## Use Case Diagram



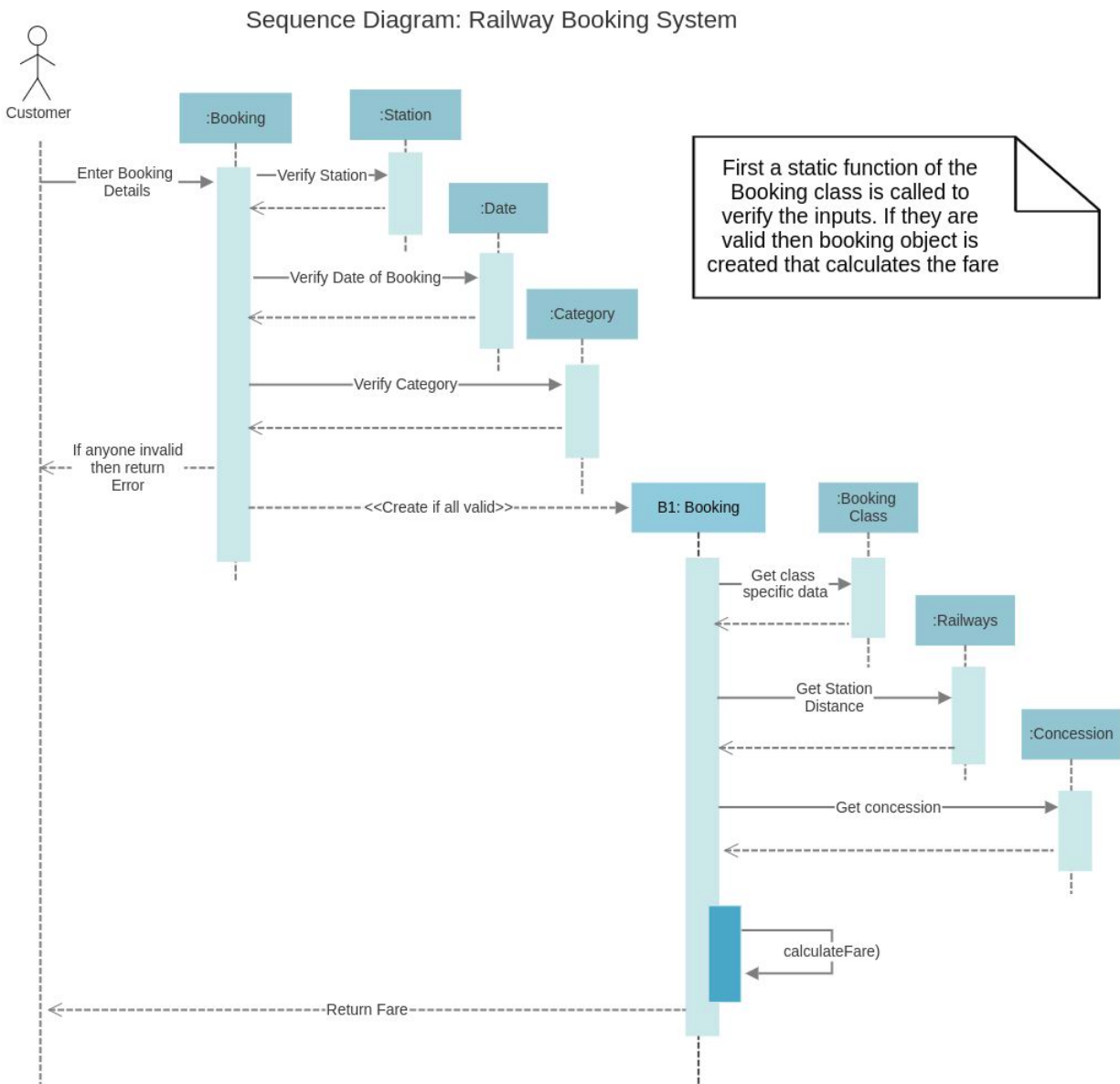
Customer has a book seat use case that includes all the other internal workings. The Manager enters master information that is stored in the database. The database also stores the bookings. The SystemAdmin does all the verification tasks and fare calculation.

# State Machine Diagram



Three basic stages defined. Idle, Validation and Booking. Validation shown as a submachine as it had internal states too. Booking not shown as a sub machine as although it has internal steps they are not necessarily states and all constitute calculating fare.

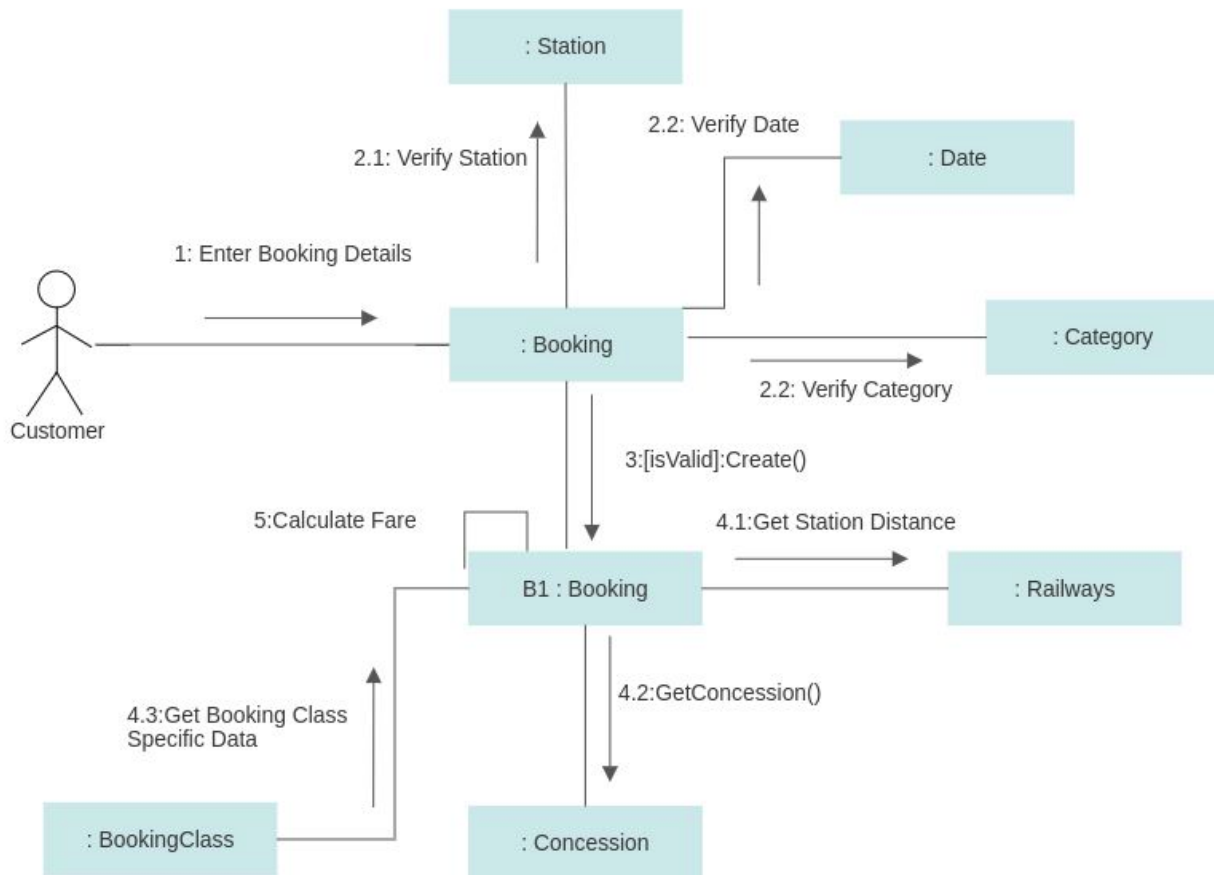
# Sequence Diagram



As written in the note, it involves 2 steps. One, calling the static function to check validity of the inputs and then create a booking object to calculate the fare after taking the data form various classes. Error handling is also shown.

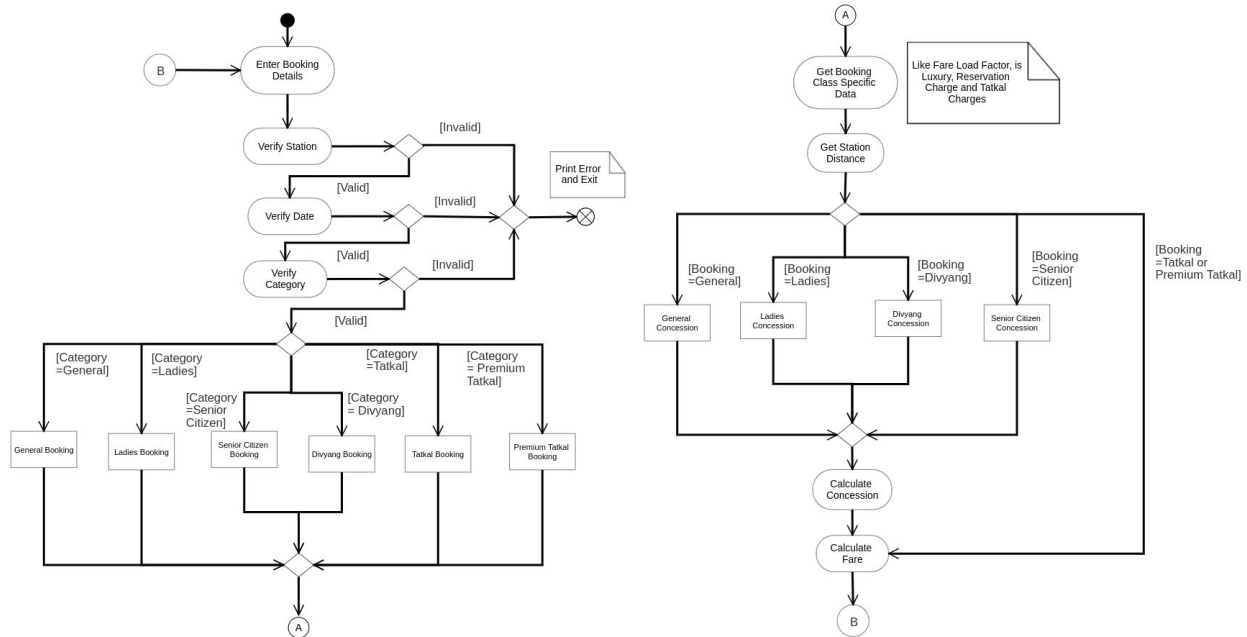
# Communication Diagram

Communication Diagram: Railway Booking System



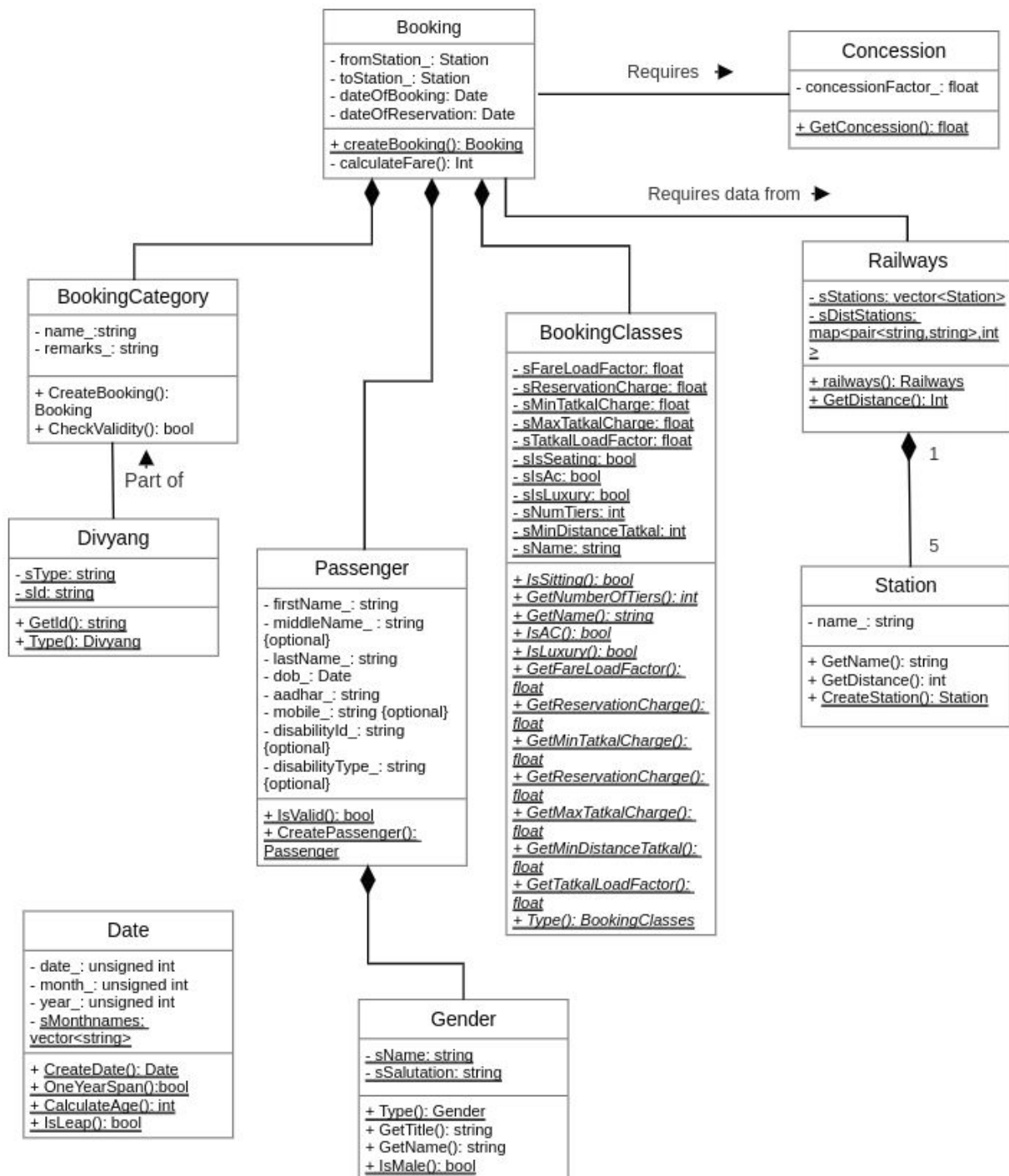
The idea is similar to that of sequence diagram although here the focus is on messages that are transferred.

# Activity Diagram



In the activity diagram, I have shown all the internal working too. First the validity of the input is checked. Not done parallelly as the code will be written in sequential fashion. Then depending on the category booking object is transferred to the next level. Then the data is received from the different classes. Depending on the booking object, a specific concession object is created that is used to calculate concession. Then the fare is calculated and system resets and users can re-enter booking details.

# Class Diagram



The hierarchies have been shown separately. The attributes and methods have not been re-written to avoid clutter.

