1 Integration Strategy

1.1 Elements to be integrated

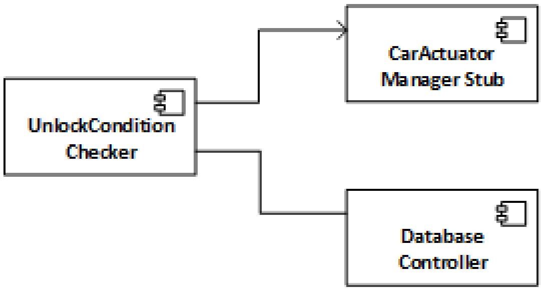
1.1.1 Client System Strategy

In order to test the client-related part of our system, we will follow a bottom-up strategy. First, we will test each functionality-related subsystem, ordering them in this way, from the most critical for the application:

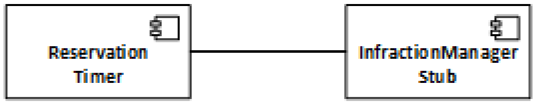
1. ReservationController
2. BalanceController
3. AccountController
4. RegistrationController
5. ReportController

For each of this subsystem, first we are going to test all the components separatly, then integrate them.

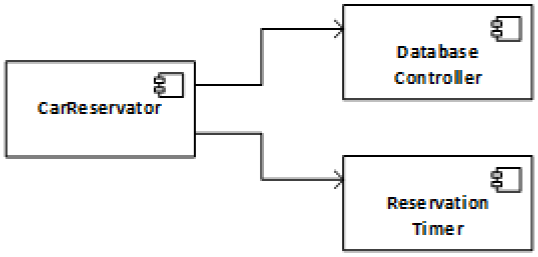
ReservationController
First we will test the UnlockConditionChecker with the CarActuatorManager Stub and the DatabaseController.



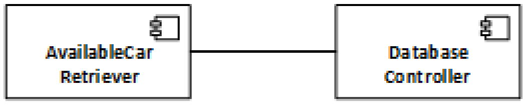
Then we will integrate the ReservationTimer with the InfractionManager Stub,



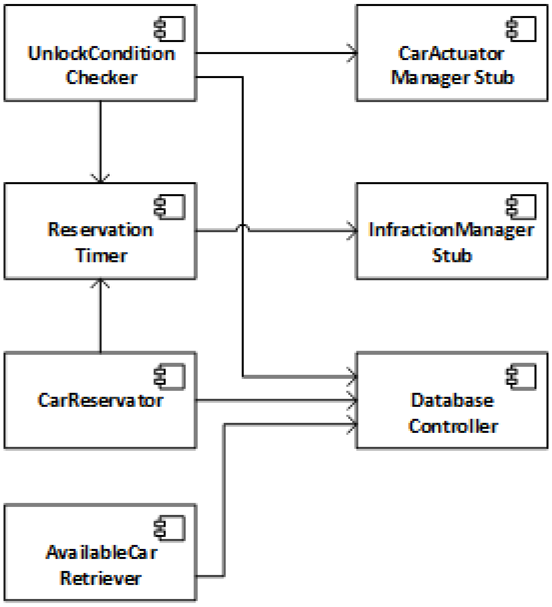
followed by the integration between the CarReservator with both DatabaseController and ReservationTimer.



Lastly we will integrate the AvailableCarRetriever with the DatabaseController.

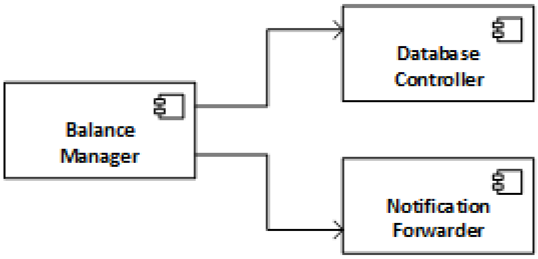


At this point we can integrate togheter all the components to test the reservation functionalities.

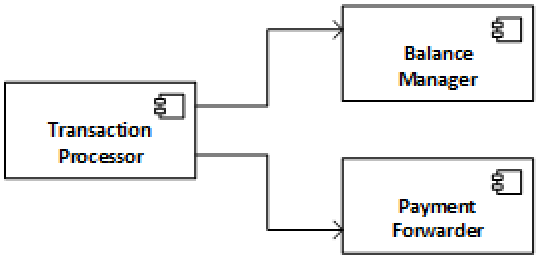


BalanceController

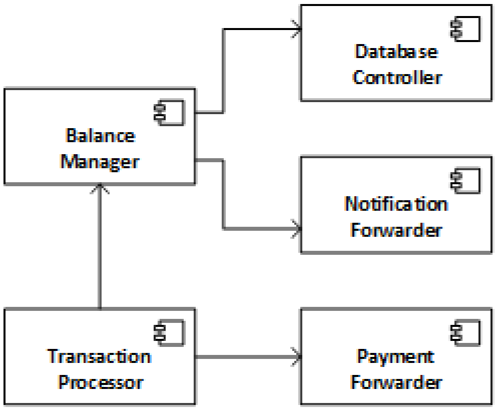
First we will integrate the BalanceController with the DataBaseController and the NotificationForwarder.



Then we will integrate the TransactionProcessor with the PaymentForwarder and the BalanceController.



In the end we will integrate all the components to test the balance-related functionalities.



AccountController For this subsystem we first integrate the InfractionManager with the BalanceManager and the DatabaseController.

Then we have to integrate the other components with the DatabaseController