1 Individual steps and test description

In this section are described the tests that have to be performed to integrate components belonging to the same subsystem; each subsection corresponds to a subsystem. The way in which the tests are described is the following: for the components that expose methods to the extern, with inputs that comes from, there is a detailed specification of the inputs (like in a unit test some representative or critical inputs are specified), while for the internal interfaces is made the assumption that

1.1 Car communication subsystem

1.1.1 Car uses the CarlListener component callback methods

startEngineCallback(carId)	
INPUT	EFFECT
The car engine starts	The carOS get the event and call the carListener method. A timer (needed to compute the ride cost) is correctly initialized and starts in the server

stopEngineCallback(carId)	
INPUT	EFFECT
A null parameter	A null value exception is raised
An invalid parameter	An invalid parameter exception is raised
A valid parameter	The ride manager is notified that the engine of the given has stopped

An invalid parameter could be:

- A carId that doesn't exist
- An empty carId
- A carId of a car that is under maintenance
- A carId of a car that is not in a ride
- A carId of a car that have not yet called startEngineCallback

1.1.2 Call to car proxy from another subsystem

getCarStatus(carId)	
INPUT	EFFECT
A null parameter	A null value exception is raised
An invalid parameter	An invalid parameter exception is raised
The id of a car that have not been asked recently	The car API are correctly called
The id of a car that have been asked recently	The car API are not called, instead the db is interrogated

unlockCar(carId)	
INPUT	EFFECT
A null parameter	A null value exception is raised
An invalid parameter	An invalid parameter exception is raised
A valid carId	The car API are correctly called

lockCar(carId)	
INPUT	EFFECT
A null parameter	A null value exception is raised
An invalid parameter	An invalid parameter exception is raised
A valid carId	The car API are correctly called

An invalid parameter could be:

- An empty carId
- A carId that doesn't exist

1.2 Registration functionality subsystem

1.2.1 Registration manager uses Payment manager

verifyValidity (paymentInformation)	
INPUT	EFFECT
A null parameter	A null value exception is raised
A not valid parameter	A non valid input exception is raised
A valid parameter, that doesn't correspond to a real account	False is returned
A valid parameter	True is returned

A not valid paymentInformation could have some field with the wrong format (number of character, or doesn't satisfies a regular expression), or an empty field.

1.2.2 External call to Registration Manager

doRegistration(username, password, paymentInformation, driveLicense)	
INPUT	EFFECT
One, some, or all null parameter	A null value exception is raised, and it is reported in the response
One, some, or all not valid parameter	An invalid parameter exception is raised, and it is reported in the response
A username that is already present in the database	The response reports that the username is already in use
All well formed, but the paymentInformation doesn't correspond to a real account	The payment manager methods are correctly called, and the response reports that the paymentInformation are not accepted
All valid, but the DriveLicense doesn't correspond to a real account	The response reports that the driveLicense is not accepted
Username that doesn't exist yet, valid password, paymentInformation and driveLicense	The response confirm the validity, the database is updated with the information

A not valid list of parameter could contain:

- An empty username
- An empty password
- A password that doesn't satisfies the general criteria of passwords (length, contains certain character...)
- A drive license that is not valid, like with an empty field or a field with wrong format (number of character, or doesn't satisfies a regular expression)
- Not well formed payment information

1.3 Reservation functionality subsystem

1.3.1 Reservation manager uses Ride manager

rideInit(ride)	
INPUT	EFFECT
A null parameter	A null value exception is raised
An invalid parameter*	A not valid input exception is raised
A valid ride parameter	A CarListener instance is correctly created, and the car os is informed that the server is listening for events. The database is updated with a new ride instance

An invalid ride could contain:

- A user that doesn't exist
- A car that doesn't exist
- A car that hasn't been reserved by the user
- A car that has been reserved by another user
- A car that is under maintenance
- A date or time that is in the past

1.3.2 External call to Reservation manager

doReservation(userId, reservationInfo)	
INPUT	EFFECT
One or both null parameter	A null value exception is raised, and it is reported in the response
One or both invalid parameter	A not valid input exception is raised, and it is reported in the response
A valid userId, reservationInfo contains a car that have been already reserved	The response reports that the car is already reserved
A valid userId, reservationInfo contains a car that is under maintenance	The response reports that it is not possible to reserve that car
A valid userId, valid reservationInfo	rideInit is called and terminates correctly, the database is updated with a new reservation instance

An invalid list of parameter could contain:

- An empty userId
- A userId that doesn't exist
- Invalid reservationInfo (date or time in the past, car that doesn't exist)

1.4 Ride functionality subsystem

1.4.1 Ride manager uses Payment manger

doPayment(paymentInformation, price)	
INPUT	EFFECT
One or both null parameter	A null value exception is raised
One or both invalid parameter	A not valid input exception is raised
Valid paymentInformation and price	The payment is carried out

An invalid list of parameter could contain:

- Not well formed payment information (what does not well formed paymentInformation means is defined in 2.1.1 section)
- PaymentInformation that doesn't correspond to a real account
- Negative price

1.4.2 External call to Ride manager

unlockRequest(userId, carId, userPosition)	
INPUT	EFFECT
One, some, or all null parameter	A null value exception is raised, and it is reported in the response
One, some, or all invalid parameter	A not valid parameter exception is raised, and it is reported in the response
Valid userId, carId, but a position that is distant from the car position	The unlock method of car proxy is not called, the response reports that the car can't be unlock
Valid userId, carId, and a position that is near to the car position	The unlock method of car proxy is correctly called, and if it terminates correctly the response confirm, otherwise the response reports that it is not possible to unlock the car

An invalid list of parameter could contain

- An empty userId
- An empty carId
- A userId that doesn't exist
- A carId that doesn't exist
- UserId and carId that are not in the same ride
- Position in wrong format

lockRequest(userId, carId)	
INPUT	EFFECT
One, some, or all null parameter	A null value exception is raised, and it is reported in the response
One, or both invalid parameter	A not valid parameter exception is raised, and it is reported in the response
Valid userId and carId	The lock method of car proxy is correctly called, and if it terminates correctly the response confirm, otherwise the response reports that it is not possible to lock the car

An invalid list of parameter could contain:

- An empty userId
- An empty carId
- A userId that doesn't exist
- A carId that doesn't exist
- UserId and carId that are not in the same ride

pauseRide(ride)	
INPUT	EFFECT
A null parameter	A null value exception is raised, and it is reported in the response
An invalid parameter	A not valid parameter exception is raised, and it is reported in the response
A valid parameter	The ride state is set to pause, a new car listener is correctly initialized and the response confirm the success of the operation

An invalid parameter could be

- A ride that doesn't exist
- A ride with the engine that is running
- Like said in 1.3.1 section

terminateRide(ride)	
INPUT	EFFECT
A null parameter	A null value exception is raised, and it is reported in the response
An invalid parameter	A not valid parameter exception is raised, and it is reported in the response
A valid parameter	The procedure to compute the cost starts and the payment component methods are correctly called. The response contains the result of the payment.

An invalid parameter is the same as an invalid parameter for pauseRide

1.5 Car management subsystem

setCarStatus(carId, status)	
INPUT	EFFECT
One or both null parameter	A null value exception is raised, and it is reported in the response
One or both invalid parameter	A not valid parameter exception is raised, and it is reported in the response
CarId corresponds to a car that is in a ride, status is reserved	

An invalid list of parameter could contains:

- An empty carId
- A carId that doesn't exist
- A not valid status (the valid status are listed in the RASD document)

getCarStatus(carId)	
INPUT	EFFECT
A null parameter	A null value exception is raised, and it is reported in the response
An invalid parameter	A not valid parameter exception is raised, and it is reported in the response

unlockCar(carId)	
INPUT	EFFECT
A null parameter	A null value exception is raised, and it is reported in the response
An invalid parameter	A not valid parameter exception is raised, and it is reported in the response

lockCar(carId)	
INPUT	EFFECT
A null parameter	A null value exception is raised, and it is reported in the response
An invalid parameter	A not valid parameter exception is raised, and it is reported in the response

1.6 Search functionality subsystem

1.7 Car monitoring subsystem

1.8 Customer comnumication subsystem