#### **2.2.1.3. Ride Seat Booking Management - RSB**

* **FR-RSB-1:** Users will be able to see the available rides.

1. The system will show a button for the user to interact and will display all available rides.
2. The available rides button will show on the main screen.
3. For users to see the rides they will need to be authenticated.

* **FR-RSB-1.1:** To see the rides passengers must indicate the pick-up point, date and time, and the arrival point.

1. In the rides display screen a filter interaction system to find rides, with a field to choose date, time origin and arrival.
2. To change the filters the user will need to press the filter screen.
3. Once these filters are used the system will only display rides that satisfy the search.

* **FR-RSB-1.2:** In addition, they will be able to define filters for ride details

1. The system will also show filters for details such as smoke and car type.
2. Users need to be logged in and have accessed the rides page.
3. User will use search filters to change pick-up point, date and time,

and the arrival point. At the same time the user can modify these details filters.

1. The application searches for the rides that satisfy the conditions.

* **FR-RSB-1.3:** When filters are applied the system will display all the results that match the specified criteria.

1. Users need to be authenticated.
2. Application search for rides that satisfy the search.
3. Users see the available rides displayed on the screen with the corresponding drivers dates and times.

* **FR-RSB-2:** Passengers will be able to see the ride details.

1. When the user is authenticated the system will display the option to see the ride detail of a specific ride the user selects on the rides menu.
2. The user needs to be authenticated.
3. The user has to enter the rides menu. The system will display the available rides.
4. The user must press a ride to open the rides detail menu.
5. The system will access the database and show all the ride information to the user.

* **FR-RSB-3:** Passengers can make seat bookings when they are seeing the ride details.

1. If a user is on a ride details screen a button can be pressed to book a seat on the desired car. The system will add the passenger to the respective ride.

* **FR-RSB-3.1:** For recurring journeys, the system will require passengers to confirm if the reservation is for a particular ride or it is intended to be a recurring booking (in this case the final date must be provided).

1. When the book seat button is pressed by the user a pop up message will be shown asking if the ride is recurrent. If so, the system will repeat the process for all the recurrent rides of that trip.

* **FR-RSB-3.2:** The system will calculate and show the user the total cost of the book.

1. A user that has been previously authenticated
2. The user is in the process of booking a ride.
3. The system will calculate the total cost of the booking and display it to the user.

* **FR-RSB-3.3:** The system will verify that the user has the required credit in his/her virtual payment account.

1. The system will access the database to find the user balance and check if the user booking has enough credits to fulfill the book cost.
2. If the user has enough credits to pay the book, the system will change the user balance account in the database, where the cost in credits will change from available credit to blocked credit.
3. After making a successful booking by a user, the system will show a reservation request on the request menu of the driver client of the application.
4. If a reservation request is done, it will show to the driver application.
   1. A request for a ride has been made by a user.
   2. The system shows the rider a new request popup on the request menu icon in the main page.
   3. The driver enters the request menu and the system shows all reservation requests pending for the driver.
   4. The driver accepts the request. The system adds the user to the corresponding rides in the database.
   5. The system generates a booking voucher with a unique code and sends it to the passenger.
5. If the driver does not accept, the passenger credit that has been blocked for that ride will be unblocked.
   1. If a reservation request is done, it will show to the driver application.
   2. A request for a ride has been made by a user.
   3. The system shows the rider a new request popup on the request menu icon in the main page.
   4. The driver enters the request menu and the system shows all reservation requests pending for the driver.
   5. The driver rejects the request. The system removes the request and unblocks the previously blocked money of the passenger in the database.
6. If there are not enough credits, the user cannot book the ride(s)
   1. The user is authenticated
   2. The user access a ride detail page
   3. The user tries to book a ride with the book ride button.
   4. The system checks in the database if there is enough credit.
   5. If there is not the system would not allow the booking. Feedback of the cancellation of the process is shown in the user interface.

* **FR-RSB-3.4:** The system will indicate the occupation of the booked seat(s) in the ride.

1. When a user sees a ride detail screen of a specific ride, the occupation of the seats will be shown.
2. When a book is validated, the system will add the user to the ride.
3. The system will check for the number of users of a ride on the database and will display them showing some visual feedback on how many seats are left on the ride detail page.

* **FR-RSB-3.5:** Passengers cannot reserve seats on different rides that overlap in time.

1. A passenger reserves a seat on one ride at a specific hour.
2. The passenger tries to reserve another seat on another ride in that same hour.
3. The system cancels the operation and shows to the user some feedback response indicating why the procedure could not be achieved.

* **FR-RSB-4:** Passengers may cancel any reservations made before the beginning of the ride. A passenger can press the cancel button on the details of the my rides page before the beginning of the ride.

1. The user accesses the my rides button in the main page.
2. The system shows all current rides booked or done by the user.
3. The user selects the ride desired to cancel. The system displays the ride details with a cancel button. The user can press the cancel button.
4. The system will access the database to change the user balance and the available seats.
5. If the cancellation has been made within 24-3 hours before the ride a 30% of the price would be not returned.
6. If the cancellation has been made within 3-0 hours before the ride there would not be a refund.

* **FR-RSB-6:** Passengers will be able to check the rides they have reserved (both those from the past and the ones for the future)

1. User is authenticated.
2. The user presses the my rides button in the main page.
3. The system displays past history of rides. A button is also displayed to change to the future history rides.
4. The system will display an icon next to each future ride showing the state of the book (pending, accepted or rejected).

* **FR-RSB-7:** Riders will receive notifications related to bookings when the driver edits the ride.

1. Drivers can change the details of the rides or cancel the ride. For changes that affect the ride such as changing the date the users will get a notification.
   1. If the ride is canceled by the driver, the system will automatically refund the passenger’s credits and emit a notification.
2. For recurring journeys, passengers will be notified before rides that a scheduled ride is approaching.
3. When the driver accepts the booking request.

* **FR-RSB-8:** Drivers will receive notifications related to bookings when the passenger books a ride.

1. When a request for a ride has been made by a user.

---------------------------------------------------------------------------------------------------------------------------

#### **2.2.1.3. Ride Seat Booking Management - RSB**

* **FR-RSB-1:**

**EI-See-available-rides:**

DET:

* + Button to see available rides
  + Pick-up point
  + Date
  + Time
  + Arrival point
  + Box for the filters

FTR:

* + Rides information
  + Users information

Conclusion: The complexity is Medium with 6 DET and 2 FTR

**EO-See-available-rides:**

DET:

* + Available rides
  + Available rides according to filter

FTR:

* + Rides information
  + Users information

Conclusion: The complexity is Low with 2 DET and 2 FTR

* **FR-RSB-2:**

**EO-See-ride-details:**

DET:

* + Button to see ride details
  + Id of ride
  + Description of the ride

FTR:

* + Rides information

Conclusion: The complexity is Low with 3 DET and 1 FTR

* **FR-RSB-3:**

**EI-Book-ride:**

DET:

* + Button to make booking
  + Pick-up point
  + Date
  + Time
  + Arrival point
  + Filters
  + Button of confirmation
  + Button to make recurring booking

FTR:

* + Users information
  + Rides information
  + Bookings information

Conclusion: The complexity is High with 8 DET and 3 FTR

**EO-Book-ride:**

DET:

* + Message indication booking made successfully or not

FTR:

* + Users information
  + Rides information
  + Bookings information

Conclusion: The complexity is Low with 1 DET and 3 FTR

**FR-RSB-3:**

**EO-Total-Cost:**

DET:

* Message indicating total cost

FTR:

* Users information
* Rides information
* Bookings information

Conclusion: The complexity is Low with 1 DET and 3 FTR

**FR-RSB-3:**

**EO-Verify-Credit:**

DET:

* Message indicating true or false

FTR:

* Users information

Conclusion: The complexity is Low with 1 DET and 1 FTR

**FR-RSB-3:**

**EO-Occupation:**

DET:

* Message indicating true or false

FTR:

* Rides information

Conclusion: The complexity is Low with 1 DET and 1 FTR

* **FR-RSB-4:**

**EI-Cancel-Booking:**

DET:

* + Button to cancel booking.

FTR:

* + Users information
  + Rides information
  + Bookings information

Conclusion: The complexity is Medium with 1 DET and 3 FTR

**EO-Cancel-Booking:**

DET:

* + Message indicating cancelation of a booking made successfully or not

FTR:

* + Users information
  + Rides information
  + Bookings information

Conclusion: The complexity is Low with 1 DET and 3 FTR

* **FR-RSB-3:** Passengers cannot reserve seats on different rides that overlap in time.

**EO-Reserve-Booking-NotValid:**

DET:

* + Text indicating error

FTR:

* + Users information
  + Bookings information

Conclusion: The complexity is Low with 1 DET and 2 FTR

* **FR-RSB-6:**

**EO-See-rides-reserved:**

DET:

* + Rides reserved box

FTR:

* + User information
  + Booking information

Conclusion: The complexity is Low with 1 DET and 2 FTR

* **FR-RSB-7:**

**EO-Receive-notification-booking:**

DET:

* + Notification related to a booking

FTR:

* + User information
  + Booking information

Conclusion: The complexity is Low with 1 DET and 2 FTR

ILFs:

**ILF-RSB:**

DET:

* Id of booking
* Id of ride
* Id of user
* Pick-up point
* Date
* Time
* Arrival point
* Cost
* VPA
* Filters (Optional)

RET:

* Mandatory attributes
* Optional attributes

Conclusion: The complexity is Low with 10 DET and 3 RET

* **FR-RSB-6:**

**ILF-See-rides-reserved:**

DET:

* + Id of user
  + Id of rides

RET:

* + Mandatory attributes

Conclusion: The complexity is Low with 2 DET and 1 FTR