Requirements:

**User Assessment - UA**

### **Functional Requirements - FR**

* **FR-UA-1:** The driver of a ride must be able to rate the passengers of his rides from one to five (lowest to highest respectively)

1. Driver must be logged in.
2. The user has to select the ride menu where the user to be reviewed is.
3. The user has to select the user to review.
4. The ride must have been finished.

* **FR-UA-2:** The driver must be able to specify the qualities that the passengers had from a quality list (for example, talkative, reserved, timely, etc.) And also will have an open field for additional observations on the passenger.

1. The driver must select the ride.
2. The driver must select the passenger to review.
3. Options with qualities to be assigned to the reviewed user will be displayed.
4. A textbox field will be displayed for additional observations.

* **FR-UA-3:** The passengers must also be able to make assessments and observations to the other passengers of the shared ride.

1. The user must select the ride.
2. The user must select the passenger to review.
3. Options with qualities to be assigned to the reviewed user will be displayed.

* **FR-UA-4:** The system will block a user's account if they receive a big number of really poor scores.

1. If a big number of poor scores is received, the user account will be blocked.
2. The user will not be able to access the blocked account (log in).
3. If login is attempted, a message will be displayed showing the account is blocked.

* **FR-UA-5:** The users must be able to file a claim for the account blocked if they think the app has made a mistake and their case will be reviewed

1. If login is attempted and the user account is blocked a file claim button will be displayed.
2. If the file claim button is pressed, the user will be able to write the claim.
3. If the file submit button is pressed, a claim will be made to the administrator.

* **FR-UA-6:** The users should be notified when the claim on the account blocking has been reviewed

1. The user must have filed a claim.
2. If the user logs in and the claim has been denied a message will display showing the claim was denied.
3. If the user logs in and the claim has been accepted the user will be able to login.
4. An email will automatically be sent to the user when the account blocking review has been done.

* **FR-UA-7:** The administrator will receive account unblock claims and will be able to accept them or deny them.

1. A notification will be shown to the administrator when a claim is filed.
2. The administrator can see all the filed claims and their corresponding users.
3. The administrator can press on the claim’s user to see their poor reviews.
4. The administrator can press an “accept claim” button to accept each claim or “deny claim” button to deny it.

# UFP Details

### **Functional Requirements - FR**

* **FR-UA-1:**

**EI-Rate passengers being driver**

DET:

* + User reviewed information
  + Ride information
  + Rate input box
  + Confirmation button

FTR:

* + Rider profile
  + Passenger profile
  + Passenger rating

Conclusion: The complexity is Medium with 4 DET and 3 FTR

* **FR-UA-2:**

**EI-Passenger qualities and observations**

DET:

* + Profile of the passenger selected
  + Dropdown with a set of qualities to choose from
  + Open text field
  + Submit button

FTR:

* + Rider profile information
  + Passenger profile information
  + Ride information

Conclusion: The complexity is Medium with 4 DET and 3 FTR

* **FR-UA-3:**

**EI-Review fellow passengers**

DET:

* + Profile of the passenger selected
  + Open text field
  + Submit button

FTR:

* + User information
  + Other passengers profile information
  + Ride information

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

* **FR-UA-4:**

**EO-Block poor users**

DET:

* + Message describing the reason for the block of the account.
  + Button to make a claim for the blocking.
  + A link to the apps FAQ section.

FTR:

* + User profile information
  + Blocking information

Conclusion: The complexity is Low with 3 DET and 2 FTR

* **FR-UA-5:**

**EI-File claim for blocked account**

DET:

* + Username
  + Description of the block
  + Open text field to explain the claim
  + Submit button

FTR:

* + User profile information
  + Blocking information

Conclusion: The complexity is Low with 4 DET and 2 FTR

* **FR-UA-6:**

**EO-Notification about the resolution of the blocked account claim**

DET:

* + Message with the verdict of the claim
  + Continue button

FTR:

* + User account information
  + Blocking information

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

* **FR-UA-7:**

**EI-Admin can manage the claims**

DET:

* + Profiles of the claimants
  + List with all the claims

FTR:

* + Profile information of the claimants
  + Blocking information

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

**ILF**

* **ILF-Reviews-And-Observations**

DET:

* + Reviewer user
  + Reviewed user
  + Rating
  + Observations text
  + Qualities

RET:

* + Mandatory subgroup
  + Optional subgroup: Reviewed user, Rating
  + Optional subgroup: Reviewed user, Observations text, Qualities

Conclusion: The complexity is Low with 5 DET and 3 RET

* **ILF-Blocked-User**

DET:

* + Blocked user

RET:

* + Mandatory subgroup

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

* **ILF-Claims**

DET:

* + Reviewer user
  + Claim text
  + Status

RET:

* + Mandatory subgroup

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