# **System operation**

1. **As an unauthorized user , I want to create a new account.**

Name: create new account

Description: This operation allows an unauthorized user to create a new account.

Operation type: Command

Roles: Unauthorized user

FR id: 1.1

Pre-condition:

user is unauthorized

user with same email and username doesn't exist

Post-condition:

new account is created

Request:

Response:

Success or failure message

REST API endpoint:

POST /user

Request:

{

"username": String,

"email": String,

"password": String,

}

Response:

201

1. **As an unauthorized user , log into an existing account.**

Name: log into an existing account

Description: This operation allows an unauthorized user to log into an existing account.

Operation type: Command

Roles: Unauthorized user

FR id: 1.2

Pre-condition:

user is unauthorized

user with this email and password exist

Post-condition:

user is logged into the system

Request:

user\_id

email

password

Response:

Success or failure message

REST API endpoint:

POST /login

Request:

{

"email": String,

"password": String,

}

Response:

201

1. **As a user, I want to add a new car.**

Name: Add a new car

Description: This operation allows a user to add a new car.

Operation type: Command

Roles: User

FR id: 2.1

Pre-condition:

user is logged into the system

car with the same name does not exists

Post-condition:

new car is added to the system

Request:

car\_id

name

Response:

Success or failure message

REST API endpoint:

POST /car

Request:

{

"name": String,

"mileage": int,

}

Response:

201

1. **As a user, I want to change info about the added car.**

Name: Change info about the added car

Description: This operation allows a user to change the information about an added car.

Operation type: Command

Roles: User

FR id: 2.2

Pre-condition:

user is logged into the system

car with this id exists and belongs to the user

Post-condition:

information about the car is updated

Request:

car\_id

Response:

Success or failure message

REST API endpoint:

PUT /car/{id}

Request:

{

"name": String,

"mileage": int,

}

Response:

200

1. **As a user, I want to delete a car.**

Name: Delete a car

Description: This operation allows a user to delete a car.

Operation type: Command

Roles: User

FR id: 2.3

Pre-condition:

user is logged into the system

car with this id exists and belongs to the user

Post-condition:

car is deleted from the system

Request:

car\_id

Response:

Success or failure message

REST API endpoint:

DELETE /car/{id}

Response:

204

1. **As a user, I want to view my cars.**

Name: View all cars.

Description: This operation allows a user to view all cars.

Operation type: Query

Roles: User

FR id: 2.4

Pre-condition:

user is logged into the system

user has added car

Post-condition:

cars are shown

Request:

Response:

Success or failure message

REST API endpoint:

GET /cars

Response:

200 OK or 400 Bad Request

1. **As a user, I want to choose between several added cars.**

Name: Choose between several added cars

Description: This operation allows a user to choose between several added cars.

Operation type: Command

Roles: User

FR id: 2.5

Pre-condition:

user is logged into the system

user has added multiple cars

Post-condition:

selected car is set as the active car

Request:

car\_id

Response:

Success or failure message

REST API endpoint:

GET /cars

GET /cars/{id}

Response:

200 OK or 400 Bad Request

1. **As a user, I want to create a new note about a car.**

Name: Create a new note about a car

Description: This operation allows a user to create a new note about a car.

Operation type: Command

Roles: User

FR id: 2.6

Pre-condition:

user is logged into the system

car with this id exists and belongs to the user

Post-condition:

new note is created and associated with the car

Request:

car\_id

Response:

Success or failure message

REST API endpoint:

POST /notes/

Request:

{

“carID” : int,

"name": String,

"current\_mileage": int,

"date": Date,

"parts\_costs": int,

"works\_costs": int,

"comment": String,

"remind": int,

}

Response:

200

1. **As a user, I want to delete a note about a car.**

Name: Delete a note about a car

Description: This operation allows a user to delete a note about a car.

Operation type: Command

Roles: User

FR id: 2.7

Pre-condition:

user is logged into the system

note with this id exists and belongs to the user

Post-condition:

note is deleted from the system

Request:

note\_id

Response:

Success or failure message

REST API endpoint:

DELETE /notes/{id}

Response:

204

1. **As a user, I want to change a note about a car.**

Name: Change a note about a car

Description: This operation allows a user to change a note about a car.

Operation type: Command

Roles: User

FR id: 2.8

Pre-condition:

user is logged into the system

note with this id exists and belongs to the user

Post-condition:

note is updated

Request:

note\_id

Response:

Success or failure message

REST API endpoint:

PUT /notes/{id}

Request:

{

"name": String,

"current\_mileage": int,

"date": Date,

"parts\_costs": int,

"works\_costs": int,

"comment": String,

"remind": int,

}

Response:

200

1. **As a user, I want to view notes about a car.**

Name: View notes about a car**.**

Description: This operation allows a user to view notes about added car.

Operation type: Query

Roles: User

FR id: 2.9

Pre-condition:

user is logged into the system

user has added car

Post-condition:

notes are shown

Request:

Response:

Success or failure message

REST API endpoint:

GET /notes?carId=

Response:

200 OK or 400 Bad Request

1. **As a user, I want to view note about a car.**

Name: View note about a car**.**

Description: This operation allows a user to view note about added car.

Operation type: Query

Roles: User

FR id: 2.10

Pre-condition:

user is logged into the system

user has added note about a car

Post-condition:

note is shown

Request:

Response:

Success or failure message

REST API endpoint:

GET /notes/{id}

Response:

200 OK or 400 Bad Request

1. **As a user, I want to create a new reminder about a car.**

Name: Create a new reminder about a car

Description: This operation allows a user to create a new reminder about a car.

Operation type: Command

Roles: User

FR id: 2.11

Pre-condition:

user is logged into the system

car with this id exists and belongs to the user

Post-condition:

new reminder is created and associated with the car

Request:

car\_id

Response:

Success or failure message

REST API endpoint:

POST /remind/{id}

Request:

{

"deadline": Date,

"comment": String,

}

Response:

201

1. **As a user, I want to delete a reminder about a car.**

Name: delete a reminder about a car

Description: This operation allows a user to delete a reminder about a car.

Operation type: Command

Roles: User

FR id: 2.12

Pre-condition:

user is logged into the system

reminder with this id exists and belongs to the user

Post-condition:

reminder is deleted from the system

Request:

remind\_id

Response:

Success or failure message

REST API endpoint:

DELETE /remind/{id}

Response:

204

1. **As a user, I want to change a reminder about a car.**

Name: Change a reminder about a car

Description: This operation allows a user to change a reminder about a car.

Operation type: Command

Roles: User

FR id: 2.13

Pre-condition:

user is logged into the system

reminder with this id exists and belongs to the user

Post-condition:

reminder is updated with new information

Request:

remind\_id

Response:

Success or failure message

REST API endpoint:

PUT /remind/{id}

Request:

{

"deadline": Date,

"comment": String,

}

Response:

200

1. **As a user, I want to view reminds about a car.**

Name: View **reminds** about a car**.**

Description: This operation allows a user to view **reminds** about added car.

Operation type: Query

Roles: User

FR id: 2.14

Pre-condition:

user is logged into the system

user has added car

Post-condition:

**reminds** are shown

Request:

Response:

Success or failure message

REST API endpoint:

GET /reminds?carId=

Response:

200 OK or 400 Bad Request

1. **As a user, I want to view remind about a car.**

Name: View remind about a car**.**

Description: This operation allows a user to view remind about added car.

Operation type: Query

Roles: User

FR id: 2.15

Pre-condition:

user is logged into the system

user has added remind about a car

Post-condition:

remind is shown

Request:

Response:

Success or failure message

REST API endpoint:

GET /notes/{id}

Response:

200 OK or 400 Bad Request