1 Section 1

1.1 Subsection 1

1.1.1 A guest registers to PowerEnJoy.

Actor	Guest
Goal	[G1]
Preconditions	The guest had never been registered before
Execution Flow	1. The guest on the home page clicks on register button to start the registration process. 2. The guest chooses among "Sign up with Google", "Sign up with Facebook" or insert manually the data. 3. In the case of manually inserting, the guest fills in at least all mandatory fields with the required informations(name, surname, username, email address, DOB). 4. In the other case, the guest presses the button and system will take the data from the option he chose. 5. The guest uploads a photo of the driving license or inserts manually the informations. 6. The guest inserts the number of the credit card and the relative CVV. 7. The system verifies the correctness of the inserted data. 8. The guest clicks on confirm button. 9. The system generates a password and provides it to the user. 10. The system will save the data in the DB. 11. The system notifies the registration and sends the user to the profile management page.
Postconditions	The guest successfully ends registration process and become a user. From now on he can log in to the application using his credential and start using PowerEnjoy.
Exceptions	1. The guest is already registered. 2. The guest inserts invalid information. 3. The guest inserts a username used by another user. 4. The guest inserts an email used by another user. 5. The guest doesn't confirm the registration. Each exception is handled warning the guest of the problem and the Execution Flow comes back to the point 2.

1.1.2 A user logs in the PowerEnjoy application.

Actor	Guest
Goal	[G2]
Preconditions	The user must be registered in the system.
Execution Flow	1. The guest opens the PowerEnjoy application and presses on the login button. 2. The guest inserts the username or email and password received during registration. 3. The system checks the couple inserted by the user. 4. The guest, from now user, is redirected to the page where he can search a car.
Postconditions	The guest is now a user, he is logged in and can use all the functionality of the system.
Exceptions	1. The guest inserts invalid credentials.

1.1.3 A user searches an available car near his position.

Actor	User
Goal	[G3a]
Preconditions	The user is logged in to the system and he has activated the GPS.
Execution Flow	1. The user presses the button to be localized on the map. 2. The system receives the user's position and checks in the DB all the available cars nearby the user. 3. The system shows on the application all the available cars. 4. The user navigates on the map to search a car.
Postconditions	The user finds a car most suitable for him.
Exceptions	1. There aren't any available cars and the system suggests to the user to search in another location.

1.1.4 A user searches an available car in a specific position.

Actor	User
Goal	[G3b]
Preconditions	The user is logged in to the system
Execution Flow	1. The user presses the search bar to insert a location. 2. The user inserts an address (street, building, place (vorrei intendere pub, bar, discoteche)) 3. The system receives the address inserted by the user. 4. The system interprets the address and converts it into a position. 5. The system checks in the DB all the available cars nearby the location. 6. The system shows on the application all the available cars. 7. The user navigates on the map to search a car.
Postconditions	The user finds a car most suitable for him.
Exceptions	1. The address inserted by the user doesn't exist. 2. There aren't any available cars and the system suggests to the user to search in another location.

1.1.5 A user reserves a car.

Actor	User
Goal	[G4]
Preconditions	The user is logged and there is at least an available car.
Execution Flow	1. The user selects a car in the map. 2. The system shows to the user the battery remaining charge. 3. The user confirms to reserve the car.
Postconditions	The car is reserved for the user for an hour.
Exceptions	1. The car is reserved by an another user before the user confirm the reservation. 2. The user is suspended by the system. In this case, the application remembers him to pay his pending bill.

1.1.6 A user cancels a reservation for a car.

Actor	User
Goal	[G4]
Preconditions	The user is logged and he reserved a car.
Execution Flow	1. The user goes to the main page of the application. 2. The user presses the "Cancel reservation" button. 3. The user confirms to cancel reservation. 4. The system set the car available again.
Postconditions	The car is ready to be used.
Exceptions	1. The reservation is expired.

1.1.7 A user unlocks the car with the QR code printed on the car.

Actor	User
Goal	[G6]
Preconditions	The user is nearby the car he reserved.
Execution Flow	1. The user presses on the camera button, takes a picture of the QR code and submits it to the system. 2. The system identifies the car with the QR code and checks the reservation. 3. The system enables the button to unlock the car on the application. 4. The user presses the button.
Postconditions	The car is ready to be ignite.
Exceptions	1. The user sent a QR code of a car he didn't reserve.

1.1.8 A user unlocks the car using his position.

Actor	User
Goal	[G6]
Preconditions	The user is nearby the car he reserved and has the localization activated.
Execution Flow	1. The user presses on the localization button and sends to the system his position. 2. The system checks the user's position and the reservation. 3. The system enables the button to unlock the car on the application. 4. The user presses the button.
Postconditions	The car is ready to be turned on.
Exceptions	1. The user is far from the car he reserved

1.1.9 A user ends the ride.

Actor	User
Goal	[G9]
Preconditions	The user is using the car.
Execution Flow	1. If the user is driving he stops the car and turns it off. 2. The display shows to the user two options: **a)** to end the ride; **b)** to park the car temporarily. 3. The user presses the button a). 4. The user exits the car and closes the doors. 5. The system locks the car automatically. 6. The system locks the car automatically. 7. The system stops charging the user. 8. The system sets the car available. 9. The system carries out the payment transaction.
Postconditions	The car is stopped in a safe area, ready to be used again.
Exceptions	• The user drives away instead of exiting the car. In this case the stop is canceled. The user drives away instead of exiting the car. In this case

1.1.10 A user parks the car without ending the ride.

Actor	User
Goal	[G10]
Preconditions	The user is using the car.
Execution Flow	1. If the user is driving he stops the car and turns it off. 2. The display shows to the user two options: **a)** to end the ride; **b)** to park the car temporarily. 3. The user presses the button b). 4. The user exits the car and closes the doors 5. The system locks the car automatically.
Postconditions	The car is stopped in a safe area, ready to be used again.
Exceptions	• The user drives away instead of exiting the car. In this case the stop is canceled. The user drives away instead of exiting the car. In this case

1.1.11 The system suggests to the user a PGS to park the car and save money.

Actor	System, user
Goal	[G12]
Preconditions	The user is using the car.
Execution Flow	1. The user inserts the destination. 2. The user starts the ride. 3. The user chooses the option "save money!" 4. The system calculates the best solution, so it searches the nearest PGS to the destination keeping in mind a uniform distribution of the cars. 5. The system suggests the PGS to the user through the display. 6. The user drives to the PGS, parks the car and ends the ride. 7. The user plugs in the car in the power grid. 8. The system detects that the car is charging at the right PGS. 9. The system applies a discount on the amount the user must pay.
Postconditions	The car is parked, ready to be used again and the user receives a discount.
Exceptions	1. There isn't any available PGS. 2. The user doesn't park the car at the PGS suggested by the system. 3. The user doesn't plug the car in the power grid.

1.1.12 Payment.

Actor	System, Payment API
Goal	–
Preconditions	The user ended a ride and there is a bill pending on him.
Execution Flow	1. The user inserts the destination. 2. The system begins the transaction. 3. The system sends the user's account data to the Payment API. 4. The system sends the amount of the bill to the Payment API. 5. The API acknowledges the payment. 6. The system closes the transaction.
Postconditions	The bill is paid
Exceptions	1. The Payment API is not available. In this case, the system the bill is still marked as "pending". 2. The transaction is rejected. In this case, the system suspends the user until he pays the pending bill.

1.1.13 User pays a pending bill.

Actor	User, System, Payment API
Goal	[G18]
Preconditions	The user has a pending bill.
Execution Flow	1. The user goes to his profile page. 2. The user selects the pending bill. 3. The user inserts the payment method. 4. The user inserts all the mandatory data. 5. The user submits the data to the system. 6. The system carries out the payment. 7. The API acknowledges the payment. 8. The system cancels the suspension.
Postconditions	The bill is paid
Exceptions	1. The Payment API is not available. In this case, the system the bill is still marked as "pending". 2. The transaction is rejected. In this case, the users remains suspended.