REQUEREMENT ANALYSIS SPECIFICATIONS DOCUMENT

Version 1.0 (only use cases)

UseCaseName: RegisterUserInSystem

Actors: Unregistered user, admin, bank

Entry conditions: The unregistered user has initiated the registration process

Flow of events:

* The user inputs his credentials
* The user inputs his payment information
* The bank replies if the payment info is valid
* The system sends the letter of the confirmation to the user’s e-mail
* The user confirms registration by e-mail

Exit conditions:

* The user has successfully confirmed the registration

Exceptions:

* The bank has denied the user’s payment information

UseCaseName: FindAvailableCarsLocation

Actors: registeredUser, cars

Entry Conditions:

* The user has pressed the button “Find an available car”

Flow of Events:

* The user selects if he wants to use his car location or to specify address
* If the user has chosen specified address, he inputs it
* If the user has chosen the current location, the system gets his location via GPS
* The system performs search for available cars
* Cars send their locations on the system request

Exit conditions:

* The user has successfully received the list of the available cars according to the chosen location
* There are no available cars in the chosen area

Exceptions:

* If GPS in the user’s device is not available, the system send a message “GPS is not available. Switch on your location searcher”
* If the incorrect address has been inputted, the system sends a message “Input a correct address”

UseCaseName: ReserveCar

Actors: registeredUser, cars

Entry Conditions:

* The list of available cars is displayed to user
* The user initiated reserving process

Flow of Events:

* User sends a request to reserve a car
* A car sends a signal to the system that it’s available
* The system confirms the reservation
* The system tags the car as unavailable
* The system sends the car’s number to the user’s email
* The system outputs the warning “You are able to reserve a car for up to one hour before you pick it up”
* The system warns the user needs to pay a fee of 1 EUR if the reservation expires

Exit conditions:

* The car has been successfully reserved

Exceptions:

UseCaseName: CancelTheReservation

Actors: the dispatcher, cars

Entry Conditions:

* The car has not been picked up

Flow of Events:

* The system tags the car as available
* The system sends a request to the bank
* The bank confirms the payment
* The system discounts 1 EUR
* The system sends a notification to the user’s e-mail

Exit conditions:

* The reservation has successfully canceled
* The fee was not paid

Exceptions:

* If the bank has not confirmed the request, the system forbids the user make any other operations and sends him the appropriate message.

UseCaseName: UnlockTheCar

Actors: registeredUser, dispatcher, cars

Entry Conditions: The user has reached the reserved car

Flow of Events:

* The user tells the system that he’s nearby
* The system checks his GPS location
* The system unlocks the reserved car
* User enters the car

Exit conditions:

* The user has successfully entered the car
* The user has messed up the car location

Exceptions:

* If GPS in the user’s device does not work, the system send a message “GPS is not available. Switch on your geo-location searcher”
* If the user does not want to enter the car, he has to choose the button “Search another car” and write to Dispatcher what is wrong with the car

UseCaseName: InitiateChargingProcess

Actors: registeredUser, car

Entry Conditions:

* The engine of the car has ignited

Flow of Events:

* The car sends a message “The engine has ignited”
* The system sets the timer
* The system starts charging
* The system outputs a message “Your trip has started. Good luck and have a nice day!” through a screen on the car
* The user puts the button “OK”
* The system outputs the current charges through the screen

Exit conditions:

* The trip has started

Exceptions:

If the user did not press “OK”, then the system continues charging until the user parks the car in a safe area and exists it.

UseCaseName: FindSafeArea

Actors: registeredUser, car, dispatcher

Entry Conditions:

* The user has pressed the button “Search the nearest safe area for parking”
* The set of safe areas is pre-defined

Flow of events:

* The car sends its GPS-coordinates to the system
* The system searches safe areas in the set of safe areas by the GPS coordinates
* The system outputs the list of the safe areas located near the car through the screen of the car

Exit conditions:

* The safe areas are determined
* There are no safe areas near the car

Exceptions:

* If the system does not find any safe areas, it outputs a message “Sorry, there are no safe areas for parking here” through the screen of the car

UseCaseName: StopChargingProcess

Actors: registeredUser, car

Entry Conditions:

* The car was parked

Flow of Events:

* The car sends a message “The car is parked” to the system
* The system checks by the car’s GPS if this area belongs to safe areas
* The system confirms that this area is safe
* The user exits the car
* The system stops charging
* The system locks the car

Exit conditions:

* The car is locked

Exceptions:

* If the area where the RegisteredUser has come does not belong to safe ones, the system outputs a message “This area is not safe for parking. Leave the car at a safe area!” and continues charging

UseCaseName: ApplyDiscountForPassengers

Actors: registeredUser, car, dispatcher

Entry Conditions:

* the user entered the car

Flow of Events:

* The system asks the user to input the number of other passengers
* The user inputs the number
* The car counts the number of the passengers except the user
* The car sends the number to the system
* The system compares the received information from the user and the car
* If the number of passengers is at least equal two, then the system applies a discount of 10% on the last ride.
* The system sends a message about the discount to the User’s e-mail

Exit conditions:

* The discount of 10% was applied
* The user took one passenger or nobody

Exceptions:

* If the number received from the user does not equal the number received from the car, the system outputs the message “The received data does not match. Please, input the correct number of other passengers or call the Dispatcher”

UseCaseName: ApplyDiscountForBattery

Actors: registeredUser, car

Entry Conditions:

* The system locked the car

Flow of Events:

* The car sends the information of the battery charge to the system
* If the battery charge is at least 50 %, then the system applies a discount of 20% on the last ride
* The system sends a message about the discount to the User’s e-mail

Exit conditions:

* The discount of 20% was applied
* The user left the car with more than 50% of the battery empty

Exceptions:

UseCaseName: ApplyDiscountForRechargingArea

Actors: registeredUser, car

Entry Conditions:

* The system locked the car

Flow of Events:

* The system checks if the safe area is a special one where the car can be recharged
* The system checks if the car is plugged into the power grid
* If the system’s checks are successful, then the system applies a discount of 30% on the last ride
* The system sends a message about the discount to the User’s e-mail

Exit conditions:

* The discount of 30% was applied
* The user left the car at non-special parking area
* The user left the car at the special area but did not plug it into the power grid

Exceptions:

UseCaseName: ChargeAdditionalPaymentForDestination

Actors: registeredUser, car

Entry Conditions:

* The car was parked at the safe area

Flow of Events:

* The system sends a request of the coordinates of the nearest power grid station by GPS
* The system calculates the distance between the safe area and the station
* If the destination is more than 3 KM, the system charges 30% more on the last ride
* The system sends the user a message about the payment of compensating for the cost required to recharge the car on-site

Exit conditions:

* The payment for recharging was assigned
* The user left the car less than 3 KM away from the nearest power grid station

Exceptions:

UseCaseName: ChargeAdditionalPaymentForDestination

Actors: registeredUser, car

Entry Conditions:

* The system locked the car

Flow of Events:

* The system checks the battery charge
* If the battery charge is less than 20 %, then the system charges 30% more on the last ride
* The system sends a message about the payment to compensate for the cost required to recharge the car on-site to the User’s e-mail

Exit conditions:

* The payment for recharging was assigned
* The user left the car with less than 80% of the battery empty

Exceptions:

UseCaseName: ProvideMoneySavingOption

Actors: registeredUser

Entry Conditions:

* The user pressed the button “Enable money saving option”

Flow of Events:

* The user inputs his final destination
* The system sends the data to the set of safe areas
* The system selects a station to ensure a uniform distribution of cars
* The system sends a request of the availability of power plugs at the selected station
* The system outputs the information about the station where to leave the car to get a discount

Exit conditions:

* The user got the required information about the station
* There are no available stations with a discount near the user’s final destination

Exceptions: