Description of the problem

PowerEnjoy is a car sharing service that exclusively employs electric cars. The company is in need of a system that allows users to find the locations of available cars located in the geographical areas they choose to explore. Then, they may complete the reservation of a car among those found with the research tool. After that, if the user who made the reservation launches the car doors unlock command and PowerEnjoy detects he is nearby, then the car doors are actually unlocked.

During the ride, the driver is notified of the current bill in real-time through the screen. At the of the ride expenses are automatically charged on the driver’s count. Since all the vehicles are electric, it is fundamental to keep them properly charged, so the users will be incentivized in being virtuous through discounts and penalty fees. A saving option will be provided in order to help users make the right choices both to save money and to help the PowerEnjoy maintain its high quality service.

Goals

PowerEnjoy has to provide these main features:

* [G1] Let users drive a car when needed.
* [G2] When a user wants to use a car, nobody else can use that car at the same time.
* [G3] Let users find and reach car.
* [G4] Let drivers find and reach locations.
* [G5] Encourage the share of a single car.
* [G6] Most of the time cars have battery level such that they can be used.
* [G7] Cars are always well distributed on the territory.
* [G8] Make sure that most of the cars are ready to use and in a good condition.

Domain properties

We suppose that these properties hold in the analyzed world:

* [D1] Sign up personal information is true.
* [D2] Cars are always connected to PowerEnjoy VPN (Virtual Private Network).
* [D3] Cars’ GPS is never neither switched off nor damaged.
* [D4] Accurate cars’ locations are known by GPS.
* [D5] Users who have an open reservation are properly located by GPS through their mobile devices.
* [D6] Available cars are ready to serve.
* [D7] Payment issues are delegated to a third party company.
* [D8] All technical sensors report correct information and they are neither switched off nor damaged.
* [D9] All possible technical issues are detected by the sensors.
* [D10] Each power plug is linked to exactly one car at the same time.
* [D11] Power grid stations always work.
* [D12] Cars’ screens are never switched off nor damaged.
* [D13] A driver always ignites the engine.
* [D14] The number of available slots in a special parking area is always known by PowerEnJoy though his VPN.
* [D15] The user who made the reservation is the same person who actually drive the car.

Glossary

In this paragraph, we go through some recurrent terms that deserve a complete definition to avoid misunderstandings along the discussion:

* **Budget**: maximum amount of money acquirable from the payment information.
* **Car**: the electric vehicles considered available by the stakeholders.
* **Company**: people who work to\*.
* **Discount**:
  + **A**: PowerEnJoy detects the user took at least two other passengers onto the car -> 10% discount.
  + **B**: the car is left no more than 50% of the battery empty -> 20% discount.
  + **C**: the car is left in a special parking area and it is plugged to a power grid station before ending the ride -> 30% discount.
* **Driver**: user that has made the reservation, boarded and then drives the car.
* **Fee**:
  + **A**: a user makes a reservation but he does not unlock the car doors within one hour from the reservation.
  + **B**: the car is left at more than 3 km from the nearest power grid station or with more than 80% of the battery empty.
* **Guest**: we use this term when referring to an unlogged user.
* **Operator**: person who works for the company in order to give support to the drivers\*.
* **Passenger**: person who enter the car.
* **Payment information**:
  + Payment method name
  + Surname and name of the owner
  + Card number
  + Valid through date
  + CVV
* **Personal information**:
  + Surname and name
  + Nationality
  + Date of birth
  + City of birth
  + Personal ID
  + Phone number
  + E-mail address
  + Username to log in
* **Power grid station**: the energy turrets where the users can leave cars to refill their batteries. These stations are located only in special parking areas.
* **Power plug**: it is about a power plug located in a power grid station.
* **Ready to serve car**: a car which has at least 10% of the battery charged and there is not any technical issue.\*\*\*
* **Reservation**: service concerning the possibility to exclusively reserve a car for a user.
* **Ride**: time between the car doors’ unlock due to the fact that the user is close to the car and the car doors’ lock due to the fact that the driver stops the engine and all the passengers exit the car.
* **Safe parking area**: a parking area included by PowerEnJoy in a list of legal places to leave cars at. Safe parking areas are the only places where a car can be parked in order to end the ride.
* **Screen**: system terminal connected to the central one used in order to communicate to the driver.
* **Special parking area**: we use this expression to point out a subset of safe parking areas where a power grid station is present.
* **Technical issue**: every kind of issues sensor detectable.
* **User**: person who provided personal and payment information which have been verified and recognized. User has access to the full set of PowerEnJoy’s services.

Assumption

There are few points that are not very clear in the specification document, so we will have to assume some facts. We assume that:

* A guest can only register, surf the list of available cars on the map and consult the service’s rules.
* \*discount a\* can be applied if and only if the number of passengers is greater or equal to three, both at the beginning and at the end of the ride.
* Each special parking area is associated to one and only one power grid station.
* Insurance renewal is company’s administration competence.
* Road fines are affair between traffic corps and company’s legal department.
* The validity of the driving license number associated with the user’s profile will have to be verified by the driver licensing authority.
* The validity of the payment information associated with the user’s profile will have to be verified by the payment company.
* Once a request is made, the user may cancel it.
* Payments are charged at the end of each ride.
* Despite the possibility of encouraging users to behave well, periodically an operator will take care about cars’ repositioning.
* A driver cannot temporary park his car and exit from it [1].
* A discount regarding power grid stations is applied if and only if the car is in a special parking area and the car is actually charging.
* The set of special parking areas is pre-defined in PowerEnJoy.
* Cars’ batteries are considered almost empty when they reach 10%.

[1] It is important to remark that a possible improvement to PowerEnjoy would provide this functionality. However it may be though as a future extension to the one with all the necessary functionalities.\*

Proposed system

The best solution to carry on the project is to develop a web platform, both in the form of a website and of a mobile application. The requisite of portability is fundamental, since one of the wanted features is the that cars should be found opened and ready to serve when they are near enough. A plugin for wearable devices could also serve to this functionality. \*what will we use to make it possible to use the system? / hardware and software of machines\*

Stakeholders

The one and only stakeholder for this project is represented by the professor who defined the assignment. The deadline for the submission of the complete document is the end of the current semester. The final submission should provide a clear and complete documentation for the development of the system, along with use cases analysis and tests.

The completeness of the document will be compromised by the need of focusing on the major features of the system. Nonetheless we’ll try to maintain a high level of consistency all along the dissertation and to develop as many aspects as possible.

Concerning the target of the application, we can think of the standard user as a stable user: the aim of PowerEnJoy should be to convince the customers to use it more than just once.

Other considerations about the system

Some considerations based on the last sentence in the previous paragraph can be made. If the aim of PowerEnJoy is to be used daily, a great effort must be spent on the user experience, in particular we require the following characteristics:

• Usability: since the set of functionalities usable by the user is quite limited it should not be difficult to make them so intuitive that no documentation will be necessary to fully understand them at a first glance

• Security: the manipulation of sensible data like payment methods requires a strong focus on security matters and the customers must be aware that they are placing their money in good hands

• Stability: it’s a major characteristic since the service must be available h24 7/7dd

• Look & feel: an essential design is necessary to catch the customers’ attention and bound them to the service

Actors identifying

The actors involved in PowerEnJoy are:

• Guest: a person who can access a limited number of the PowerEnJoy’s features, e.g. the research tool. They can neither make reservations nor access to any features that require the possibility of online payments\*.

• User: a person who has registered and therefore has provided his personal and payment information.

• Driver: a user who has made a reservation for a car and now is driving it.

• Operator: a person who takes care of cars’ maintainability both for technical and legal issues.

Requirements

Thinking that the domain properties, written in paragraph 1.3, holds, from the goals, written in paragraph 1.2, we can derive our requirements. We write below, for each goal, what we can derive:

Thinking that the domain properties, written in paragraph 1.3, holds, from the goals, written in paragraph 1.2, we can derive our requirements. We write below, for each goal, what we can derive:

1. **[G1] Let users drive a car for a fare when needed.**
   1. [R1] Sign up functionality.
   2. [R2] Verification of driving license and payment information.
   3. [R3] Log in functionality.
   4. [R4] Prevent car doors from unlocking unless the user is nearby.
   5. [R5] Start counting expenses for a given amount of money per minute, as soon as the engine ignites.
   6. [R6] Charge expenses on the user’s account after the end of each ride.
   7. [R7] Consult last rides’ invoices.
2. **[G2] When a user wants to use a car, nobody else can use that car at the same time.**
   1. [R1] Make reservations valid for a single car at the same time.
   2. [R2] The reservation expires by one hour and the A fee is charged, then the car is available again.
   3. [R3] Cars in use are marked as unavailable.
   4. [R4] Lock car doors after each ride.
   5. [R5] Stop counting expanses when a ride terminates and generate the invoice.
   6. [R6] Restore cars’ availability after each ride.
   7. [R7] Restore cars’ availability once at least 10% battery level is reached.
3. **[G3] Let users find and reach cars.**
   1. [R1] Find cars located nearby or specifying an address.
   2. [R2] Navigate to a reserved car.
4. **[G4] Let drivers find and reach safe parking areas and locations.**
   1. [R1] Guide each driver to a chosen location.
   2. [R2] If money saving option is enabled, then guide the driver to a special parking area according to driver’s final destination.
5. **[G5] Encourage the share of a single car.**
   1. [R1] Apply the A discount on the last ride.
6. **[G6] Most of the time cars have battery level such that they can be used.**
   1. [R1] Show the battery level during the ride.
   2. [R2] View special parking areas.
   3. [R3] Apply the B discount on the last ride.
   4. [R4] Apply the C discount on the last ride.
   5. [R5] Periodically, if there is a low battery level car, a notification will be sent to operators who will intervene.
   6. [R6] Charge the B fee on the last ride.
   7. [R7] Enable money saving option.
7. **[G7] Cars are always well distributed on the territory.**
   1. [R1] Charge the B fee on the last ride.
   2. [R2] Periodically, if there is a non-uniform cars’ distribution, a notification will be sent to operators who will intervene.
   3. [R3] View special parking areas.
   4. [R4] Enable money saving option.
8. **[G8] Make sure that most of the cars are ready to use and in a good condition.**
   1. [R1] When an issue is detected, a notification will be sent to operators who will intervene.
   2. [R2] Make a report whether an issue has been discovered by the driver.
   3. [R3] Periodically, an operator will take care of maintain the cars clean according to drivers reported issue.

Scenario identifying

**Scenario 1**

Sebastian always reach his office by bus but today is scheduled a strike. So, the day before he discovers PowerEnJoy and he signs up. The day of the strike, he uses PowerEnJoy’s app to make a reservation. He finds a car around and reaches it. Once near to the reserved car, PowerEnJoy unlocks car doors and then Sebastian enter it.

Use Case

|  |  |
| --- | --- |
| **Actor** | Guest |
| **Goal** | [G1] [R1] [R2] |
| **Precondition** | - |
| **Event Flow** | 1. A guest on the home page presses the “sign up” button to start the registration process. 2. The guest fills the form with personal, payment and driving license information. 3. The guest verifies and confirms information entered. 4. PowerEnJoy forwards driving license information to the driver licensing authority which will validate them. 5. PowerEnJoy forwards payment information to the payment company which will validate them. |
| **Post-Condition** | The guest receives a password to log in PowerEnJoy via e-mail.  The guest can now log in PowerEnJoy.  All his information is stored. |
| **Exception** | 1. A guest fills the form with information regarding another user. 2. The driving license information has not been validated. 3. The payment information has not been validated. |
| **Exception Handling** | (a), (b), (c) The signing up process is aborted and the guest is notified via e-mail. |

|  |  |
| --- | --- |
| **Actor** | Guest |
| **Goal** | [G1] [R3] |
| **Precondition** | - |
| **Event Flow** | 1. A guest on the home page presses the “log in” button to use PowerEnJoy’s features. 2. The guest fills basic personal information and the password given by PowerEnJoy. |
| **Post-Condition** | The guest is now logged in PowerEnJoy.  The guest becomes a user. |
| **Exception** | 1. Either basic personal information or password is wrong. |
| **Exception Handling** | (a) Ask the user to refill both basic personal information and the password. |

|  |  |
| --- | --- |
| **Actor** | User |
| **Goal** | [G1] [R4] [R5] |
| **Precondition** | The user has an active reservation for a car. |
| **Event Flow** | 1. The user launches the car doors unlock command. 2. PowerEnJoy detects the user’s position using the GPS. 3. The driver ignites the engine. |
| **Post-Condition** | PowerEnJoy unlocks car doors.  The ride starts.  The user becomes a driver.  The driver can drive the car.  Start counting expenses for a given amount of money per minute. |
| **Exception** | 1. The user is too far from the car. |
| **Exception Handling** | (a) Prevent car doors unlock. |

|  |  |
| --- | --- |
| **Actor** | User |
| **Goal** | [G1] [R6] [R7] |
| **Precondition** | The user has done at least one ride. |
| **Event Flow** | 1. PowerEnJoy charges expenses based on user’s payment information at the end of each ride. 2. PowerEnJoy creates an invoice related to last ride’s expenses. |
| **Post-Condition** | The user consults last rides’ invoices. |
| **Exception** | - |
| **Exception Handling** | - |

|  |  |
| --- | --- |
| **Actor** | User |
| **Goal** | [G2] [R1] [R2] [R3] |
| **Precondition** | The user has found an available car. |
| **Event Flow** | 1. The user presses the “make a reservation” button. 2. Reservation expiring timer starts. 3. The car is marked as unavailable. |
| **Post-Condition** | The user has reserved a car. |
| **Exception** | 1. Another user makes a reservation for the same car in the very short moment before so the car is actually unavailable. 2. The user does not start the ride within one hour. |
| **Exception Handling** | (a) The user may find another available car.  (b) The user has to pay the A fee and the reservation for the car expires. |

|  |  |
| --- | --- |
| **Actor** | Driver |
| **Goal** | [G2] [R4] [R5] [R6] |
| **Precondition** | - |
| **Event Flow** | 1. The driver parks into a safe parking area. 2. The driver turns off the engine. 3. Every passenger exits the car. |
| **Post-Condition** | PowerEnJoy locks car doors.  The ride ends.  Stop counting expenses.  PowerEnJoy generates the invoice.  The car is marked as available. |
| **Exception** | 1. The driver parks in a non-safe parking area. 2. The driver does not turn off the engine. 3. Every passenger does not exit the car. 4. The car has almost empty battery. |
| **Exception Handling** | (a), (b), (c) The car doors does not unlock and the ride does not terminate.  (d) Car doors get locked, the ride ends but the car is marked as unavailable. |

|  |  |
| --- | --- |
| **Actor** | Guest / User |
| **Goal** | [G3] [R1] |
| **Precondition** | - |
| **Event Flow** | 1. A guest or a user on the home page presses the “look for a car” button to find a car. 2. The guest or the user fill the search filter in order according to places of interest, address or actual position. 3. The guest or the user may consult cars’ status. |
| **Post-Condition** | PowerEnJoy finds out all the cars which match the search filter.  Cars’ status is available to be consulted. |
| **Exception** | - |
| **Exception Handling** | - |

|  |  |
| --- | --- |
| **Actor** | User |
| **Goal** | [G3] [R2] |
| **Precondition** | The user has an active reservation for a car. |
| **Event Flow** | 1. The user on the recap of the reservation presses the “navigate to the car” button to reach the car. 2. The user selects a path according to his necessity. |
| **Post-Condition** | PowerEnJoy provides navigation tips to the user in order to reach the reserved car. |
| **Exception** | - |
| **Exception Handling** | - |

|  |  |
| --- | --- |
| **Actor** | Driver |
| **Goal** | [G4] [R1] |
| **Precondition** | - |
| **Event Flow** | 1. A driver on the screen presses the “look for a destination” button in order to navigate to a safe parking area near to a specific location. 2. The driver fills the search filter according to places of interest or a specific address. 3. The driver selects a path according to his necessity. |
| **Post-Condition** | PowerEnJoy provides navigation tips to the driver in order to reach the target location. |
| **Exception** | - |
| **Exception Handling** | - |

|  |  |
| --- | --- |
| **Actor** | Driver |
| **Goal** | [G4] [R2] |
| **Precondition** | - |
| **Event Flow** | 1. A driver, on the screen presses the “money saving option” button. 2. The driver fills the search filter according to places of interest or specific address. 3. The driver selects a special parking area among the ones nearer to the target location. 4. The driver selects a path according to his necessity. |
| **Post-Condition** | PowerEnJoy provides navigation tips to the driver in order to reach the nearest special parking area from the target location. |
| **Exception** | 1. While reaching the special parking area, it may happen that the target special parking area becomes full. |
| **Exception Handling** | (a) PowerEnJoy will notify the driver about what happened. Then, it will evaluate another special parking area target. |

|  |  |
| --- | --- |
| **Actor** | Operator |
| **Goal** | [G6] [R5] |
| **Precondition** | PowerEnJoy detects an almost empty battery level car and it notifies the operator. |
| **Event Flow** | 1. The operator accepts the notification. 2. The operator opens a new paperwork to manage the issue. 3. PowerEnJoy shows cars to be potentially low battery level. 4. The operator selects a car and manually switches it to unavailable. 5. PowerEnJoy suggests a special parking area where to place the car in order to recharge the battery. 6. The operator picks the car through a by-pass. 7. The operator moves the car to the suggested special parking area. 8. The operator plugs the car to a power grid station. 9. PowerEnJoy will restore car’s availability when a certain level of battery is reached. 10. The operator closes the paperwork. |
| **Post-Condition** | A car is placed in a special parking area and it is charging. |
| **Exception** | 1. The car has a very low battery level and the operator can neither move it to a special parking area. |
| **Exception Handling** | (a) The operator reaches the car bringing with him a supply battery and plugs it. After an enough battery charge, the operator takes care of moving the car to a special parking area and to restore the car’s availability. |

|  |  |
| --- | --- |
| **Actor** | Operator |
| **Goal** | [G7] [R2] |
| **Precondition** | PowerEnJoy detects a non-uniform cars’ distribution and it notifies the operator. |
| **Event Flow** | 1. The operator accepts the notification. 2. The operator opens a new paperwork to manage the issue. 3. PowerEnJoy shows cars to be potentially re-distributed. 4. The operator selects a car and manually switches it to unavailable. 5. PowerEnJoy suggests a safe parking area where to place the car. 6. The operator picks the car through a by-pass. 7. The operator moves the car to the suggested safe parking area. 8. The operator manually switches the car as available. 9. The operator closes the paperwork. |
| **Post-Condition** | A car is placed in a better safe parking area. |
| **Exception** | 1. A user makes a reservation for a car which has been shown in the list of cars to be potentially re-distributed before the operator selects it. |
| **Exception Handling** | (a) PowerEnJoy warn the operator that the selected car has been reserved by a user. So, the operator is prompted to the list of cars to be potentially re-distributed. |

|  |  |
| --- | --- |
| **Actor** | Guest / User / Driver |
| **Goal** | [G7] [R3] |
| **Precondition** | - |
| **Event Flow** | 1. An actor on the home page clicks on the “show special parking area” button. |
| **Post-Condition** | PowerEnJoy provides a list of all special parking areas. |
| **Exception** | - |
| **Exception Handling** | - |

|  |  |
| --- | --- |
| **Actor** | Operator |
| **Goal** | [G8] [R1] |
| **Precondition** | PowerEnJoy detects a technical issue and it sends a notification an operator. |
| **Event Flow** | 1. The operator accepts the notification. 2. The operator opens a new paperwork to manage the issue. 3. The operator evaluates the issue relevance. 4. The operator manually switches it to unavailable. 5. The operator picks the car through a by-pass. 6. The operator, once in the car, may either launch the external tow truck service through the screen or move the car to the affiliate mechanical workshop according to the issue relevance. 7. After fixing up, the operator brings the car to a safe parking area. 8. The operator restores car availability. 9. The operator closes the paperwork. |
| **Post-Condition** | The car is fixed up. |
| **Exception** | 1. Actually, the technical issue is a minor issue. 2. The technical issue is detected while a user has reserved the car or someone is driving it. 3. A user makes a reservation shortly before the operator was marking that car as unavailable. |
| **Exception Handling** | (a) The operator just stores the issue report and the car availability is not changed.  (b), (c) If the operator does not consider that issue as a minor issue, then PowerEnJoy notifies the user/driver about it and it exhorts to cancel the reservation or end the ride. |

|  |  |
| --- | --- |
| **Actor** | Driver |
| **Goal** | [G8] [R2] |
| **Precondition** | A driver is in the car. |
| **Event Flow** | 1. The driver pushes the “report an issue” button through the screen. 2. The driver fills relevant fields regarding the observed issue. |
| **Post-Condition** | The report is saved. |
| **Exception** | - |
| **Exception Handling** | - |

|  |  |
| --- | --- |
| **Actor** | Operator |
| **Goal** | [G8] [R3] |
| **Precondition** | - |
| **Event Flow** | 1. The operator pushes the “show reported issues” button. 2. The operator selects all issues regarding a single car. 3. The operator opens a new paperwork to manage the car issues. 4. The operator switches the car as unavailable. 5. The operator picks the car through a by-pass. 6. The operator will take care of solving the issues. 7. The operator brings the car to a safe parking area. 8. The operator restores car availability. 9. The operator closes the paperwork. |
| **Post-Condition** | The car is clean and in a good condition. |
| **Exception** | - |
| **Exception Handling** | - |