|  |  |
| --- | --- |
| **Name** | Sign up |
| **Actors** | * Guest |
| **Entry conditions** | * The guest isn’t registered * The guest has opened the mobile app or the web site home page |
| **Flow of events** | * The guest click on the “Sign Up” button * The system shows him the form to be filled * The guest add all requested information * The guest click on the “Confirm” button * The system verifies the information, confirms the registration and stores new user’s information |
| **Exit conditions** | * The guest is correctly registered |
| **Exceptions** | * If the guest is already registered, the system notifies the user with a pop-up message * If guest information are incomplete or incorrect, the system notifies the guest and ask him to rewrite them |

|  |  |
| --- | --- |
| **Name** | Log In |
| **Actors** | * Guest |
| **Entry conditions** | * The guest is registered * The guest has opened the mobile app or the web site home page |
| **Flow of events** | * The guest click on the “Log in” button * The system shows him the form to be filled * The guest write username and password * The guest click on the “Confirm” button * The system verifies the information and shows user’s home page |
| **Exit conditions** | * The user is correctly logged in |
| **Exceptions** | * If validation process doesn’t terminate well, the system ask the user to rewrite username and password |

|  |  |
| --- | --- |
| **Name** | New reservation |
| **Actors** | * Passenger |
| **Entry conditions** | * The passenger is logged in * The system is showing user’s home page |
| **Flow of events** | * The passenger click on the “New reservation” button * The system shows him a form to be filled with all trip information (starting address, destination address, leaving date and time, number of passengers) * The user adds all information and click on the “Confirm” button * The system checks the information, analyse the request and confirm with a message * The system generates a “Request for service – real time” ten minutes before the meeting time |
| **Exit conditions** | * The reservation is correctly stored |
| **Exceptions** | * If the information are wrong the system asks to rewrite them * If the reservation can’t be satisfied the system shows an error message |

|  |  |
| --- | --- |
| **Name** | Modify reservation |
| **Actors** | * Passenger |
| **Entry conditions** | * The passenger is logged in * The system is showing user’s home page |
| **Flow of events** | * The passenger click on the “Reservation history” button * The system shows him all his reservations * The user click on the reservation he want to modify * The system shows him a pop up window with two options: “Modify” or “Delete” * The user click on the “Modify” button * The system shows him a form to be filled with all trip information (starting address, destination address, leaving date and time, number of passengers) * The user adds all information and click on the “Confirm” button * The system checks the information, analyse the request and confirm with a message * The system generates a “Request for service – real time” at the right moment to satisfy user reservation |
| **Exit conditions** | * The reservation is correctly modified and stored |
| **Exceptions** | * If there is no reservation in the history, the system shows an error message * If it is too late to modify the reservation (less then two hours before the ride), the system shows an error message * If the information are wrong the system asks to rewrite them * If the reservation can’t be satisfied the system shows an error message |

|  |  |
| --- | --- |
| **Name** | Delete reservation |
| **Actors** | * Passenger |
| **Entry conditions** | * The passenger is logged in * The system is showing user’s home page |
| **Flow of events** | * The passenger click on the “Reservation history” button * The system shows him all his reservations * The user click on the reservation he want to delete * The system shows him a pop up window with two options: “Modify” or “Delete” * The user click on the “Delete” button * The system delete the reservation and shows reservation history |
| **Exit conditions** | * The reservation is correctly deleted |
| **Exceptions** | * If there is no reservation in the history, the system shows an error message |