**Use case Narrative**

**Motorcycle rider**

|  |  |
| --- | --- |
| **Uses Case** | View location of the mechanic |
| **Actor** | Motorcycle User |
| **Description** | The motorcycle user will view the location of the mechanic the system will help find a motorcycle mechanic. |
| **Precondition** | * The user must be registered to the system * The user must be signed in to the system * The user must have motorcycle to repair |
| **Trigger** | The use case initiated when the motorcycle user want a help from motorcycle mechanic. |
| **Typical Course of Events:**   |  |  | | --- | --- | | **Actor Action** | **System Response** | | **Step 1:** The motorcycle user will click on the location button | **Step 2:** The system responds to the user by giving the location of the mechanic  **Step 3:** The system will help the motorcycle user to locate a mechanic or to book.  **Step 4:** Once the user booked a mechanic the system or the request will pending.  **Step 5:** For each pending mechanic the request will grant when the motorcycle mechanic confirms the request. | | |
| **Alternate Courses** | **Alt-Step 2:** if the motorcycle user don’t have any response of the mechanic the user can also check the steps on how to repair the motorcycle.  . |
| **Conclusion** | This use case concludes when the club member receives confirmation of the order. |
| **Postcondition** | The order has been recorded and if the ordered products were available, they were released. For any product not available a back order has been created. |
| **Exceptions** | * The motorcycle user is no longer approved by double request. * The system asked the mechanics if it is done or to exit. * The Customer request another type of problem. * The Customer asked to exit   Appropriate message is displayed for every exception. |

|  |  |
| --- | --- |
| **Uses Case** | Edit Profile |
| **Actor** | Motorcycle User |
| **Description** | The motorcycle user will edit and update their profile. |
| **Precondition** | * The user must be registered to the system * The user must be signed in to the system * The user must have motorcycle to repair |
| **Trigger** | The use case initiated when the motorcycle user want to update or edit their personal information. |
| **Typical Course of Events:**   |  |  | | --- | --- | | **Actor Action** | **System Response** | | **Step 1:** The motorcycle user will click on the edit profile. | **Step 2:** The system responds to the user by giving the information given before.  **Step 3:** The system will let the motorcycle rider to edit information.  **Step 4:** The information that the motorcycle rider gives it records to the database. | | |
| **Alternate Courses** | **Alt-Step 2:** If the user will not update information the current information will stay the same.  . |
| **Conclusion** | This usecase initiated when the motorcycle rider will edit their profile . |
| **Postcondition** | The update has been recorded and if the update were available, the information will be updated. |
| **Exceptions** | * The motorcycle user is no longer approved by double request. * The system asked the mechanics if it is done or to exit. * The Customer request another type of problem. * The Customer asked to exit   Appropriate message is displayed for every exception. |

|  |  |
| --- | --- |
| **Uses Case** | View Pending |
| **Actor** | Motorcycle User |
| **Description** | The motorcycle rider will view the pending mechanic that they are booked. |
| **Precondition** | * The user must be registered to the system * The user must be signed in to the system * The user must have motorcycle to repair |
| **Trigger** | The use case initiated when the motorcycle user got a pending request. |
| **Typical Course of Events:**   |  |  | | --- | --- | | **Actor Action** | **System Response** | | **Step 1:** The motorcycle user will click on the pending. | **Step 2:** The system responds to the user by giving the pending request.  **Step 3:** The system will help the motorcycle user to see the pending request. | | |
| **Alternate Courses** | **Alt-Step 2:** if the motorcycle user don’t have any response of the mechanic the user can check the pending mechanics.  . |
| **Conclusion** | This use case concludes when the club member receives confirmation of the order. |
| **Postcondition** | The order has been recorded and if the ordered products were available, they were released. For any product not available a back order has been created. |
| **Exceptions** | * The motorcycle user is no longer approved by double request. * The system asked the mechanics if it is done or to exit. * The Customer request another type of problem. * The Customer asked to exit   Appropriate message is displayed for every exception. |

|  |  |
| --- | --- |
| **Uses Case** | Booked Mechanic |
| **Actor** | Motorcycle User |
| **Description** | The motorcycle user will booked a request to the motorcycle mechanic. |
| **Precondition** | * The user must be registered to the system * The user must be signed in to the system * The user must have motorcycle to repair |
| **Trigger** | The use case initiated when the motorcycle user will booked a request or help to the motorcycle mechanic. |
| **Typical Course of Events:**   |  |  | | --- | --- | | **Actor Action** | **System Response** | | **Step 1:** The motorcycle user will locate the motorcycle mechanic and the system will booked a request. | **Step 2:** The system responds by booking a motorcycle mechanic  **Step 3:** The system verifies the booked request information from the motorcycle user. | | |
| **Alternate Courses** | **Alt-Step 2:** The booked request has not provided all the information of the motorcycle mechanic.  **Alt-Step 3:** If the user cannot do the request; The motorcycle have also another option the steps on how to repair motorcycle problems. |
| **Conclusion** | This use case concludes when the motorcycle user booked a request from the motorcycle mechanic. |
| **Postcondition** | The booked request recorded and if the request were available, the booked request were released. |
| **Exceptions** | * The motorcycle user is no longer approved by double request. * The system asked the mechanics if it is done or to exit. * The Customer request another type of problem. * The Customer asked to exit   Appropriate message is displayed for every exception. |

|  |  |
| --- | --- |
| **Uses Case** | Mechanic feedback |
| **Actor** | Motorcycle user |
| **Description** | The motorcycle user will give feedback to the motorcycle mechanics service. |
| **Precondition** | * The user must be registered to the system * The user must be signed in to the system * The user must have motorcycle to repair |
| **Trigger** | The use case initiated when the motorcycle user will booked a request or help to the motorcycle mechanic |
| **Typical Course of Events:**   |  |  | | --- | --- | | **Actor Action** | **System Response** | | **Step 1:** The motorcycle user will give feedback of what they have experience of the motorcycle mechanic’s service, | **Step 2:** The system responds by giving the user to give feedback to the mechanic.  **Step 3:** The system verifies the feedback information against to the motorcycle mechanic. | | |
| **Alternate Courses** | **Alt-Step 2:**If the motorcycle user gives feedback from the user the information of the motorcycle mechanic will be have a feedback. |
| **Conclusion** | This use case concludes when the motorcycle mechanic receive from the motorcycle user. |
| **Postcondition** | The feedback has been recorded and if the feedback were available, they were released. For any mechanic does not have feedback the mechanic their information would stay stable. |
| **Exceptions** | * The motorcycle user is no longer approved by double request. * The system asked the mechanics if it is done or to exit. * The Customer request another type of problem. * The Customer asked to exit.   Appropriate message is displayed for every exception. |

**Motorcycle Mechanic**

|  |  |
| --- | --- |
| **Uses Case** | View location of the motorcycle user |
| **Actor** | Motorcycle Mechanic |
| **Description** | The motorcycle mechanic will view the location of the motorcycle user. |
| **Precondition** | * The user must be registered to the system * The user must be signed in to the system * The user must have motorcycle to repair |
| **Trigger** | The use case initiated when the motorcycle mechanic will locate the motorcycle user. |
| **Typical Course of Events:**   |  |  | | --- | --- | | **Actor Action** | **System Response** | | **Step 1:** The motorcycle mechanic will locate the motorcycle user | **Step 2:** The system responds by giving the location of the motorcycle user.  **Step 3**: The system verifies the motorcycle user information of what the user gives.  **Step 4**: For each location of the motorcycle user, the system validates the user’s identity. | | |
| **Alternate Courses** | **Alt-Step 2:** The motorcycle mechanic will get the information and the location of the motorcycle user.  **Alt-Step 3**: If the motorcycle user booked a request the location will be shown by the mechanic. If the mechanic can’t locate the system has another option which the steps on to repair their motorcycle problem. |
| **Conclusion** | This use case concludes when the motorcycle mechanic receives confirmation of the motorcycle user. |
| **Postcondition** | The location has been recorded and if the location were available, the will response. For any request the mechanic will not get the user may use the steps on how to repair the motorcycle problems. |
| **Exceptions** | * The motorcycle user is no longer approved by double request. * The system asked the mechanics if it is done or to exit. * The Customer request another type of problem. * The Customer asked to exit   Appropriate message is displayed for every exception. |

|  |  |
| --- | --- |
| **Uses Case** | Edit Profile |
| **Actor** | Motorcycle Mechanic |
| **Description** | The motorcycle Mechanic will edit and update their profile. |
| **Precondition** | * The user must be registered to the system * The user must be signed in to the system * The user must have motorcycle to repair |
| **Trigger** | The use case initiated when the motorcycle Mechanic want to update or edit their personal information. |
| **Typical Course of Events:**   |  |  | | --- | --- | | **Actor Action** | **System Response** | | **Step 1:** The motorcycle Mechanic will click on the edit profile. | **Step 2:** The system responds to the user by giving the information given before.  **Step 3:** The system will let the motorcycle mechanic to edit information.  **Step 4:** The information that the motorcycle mechanic gives it records to the database. | | |
| **Alternate Courses** | **Alt-Step 2:** If the user will not update information the current information will stay the same.  . |
| **Conclusion** | This usecase initiated when the motorcycle mechanic will edit their profile . |
| **Postcondition** | The update has been recorded and if the update were available, the information will be updated. |
| **Exceptions** | * The motorcycle user is no longer approved by double request. * The system asked the mechanics if it is done or to exit. * The Customer request another type of problem. * The Customer asked to exit   Appropriate message is displayed for every exception. |

|  |  |
| --- | --- |
| **Uses Case** | Accept Request |
| **Actor** | Motorcycle Mechanic |
| **Description** | The motorcycle mechanic will accept a booked request from the motorcycle rider. |
| **Precondition** | * The user must be registered to the system * The user must be signed in to the system * The user must have motorcycle to repair |
| **Trigger** | The use case initiated when the motorcycle mechanic get a booked request from the motorcycle rider. |
| **Typical Course of Events:**   |  |  | | --- | --- | | **Actor Action** | **System Response** | | **Step 1:** The motorcycle mechanic will decide to accept the booked request. | **Step 2:** The system responds by giving the location of the motorcycle rider.  **Step 3**: The system verifies the motorcycle rider’s location. | | |
| **Alternate Courses** | **Alt-Step 3**: If the motorcycle rider booked a request the mechanic will accept the booked request. |
| **Conclusion** | This use case concludes when the motorcycle mechanic gets booked request. |
| **Postcondition** | The request has been recorded and if the location were available, the will response. For any request the mechanic will not get the user will not granted. |
| **Exceptions** | * The motorcycle user is no longer approved by double request. * The system asked the mechanics if it is done or to exit. * The Customer request another type of problem. * The Customer asked to exit   Appropriate message is displayed for every exception. |

**ADMIN**

|  |  |
| --- | --- |
| **Uses Case** | Add mechanic |
| **Actor** | Admin |
| **Description** | The motorcycle admin will add a motorcycle mechanic. |
| **Precondition** | * The user must be registered to the system * The user must be signed in to the system * The user must have motorcycle to repair |
| **Trigger** | The use case initiated when the motorcycle mechanic wants to register. |
| **Typical Course of Events:**   |  |  | | --- | --- | | **Actor Action** | **System Response** | | **Step 1:** The admin will add mecahnic | **Step 2:** The system responds by adding new mechanic | | |
| **Alternate Courses** | **Alt-Step 3**: If the motorcycle admin will add a mechanic the mechanic will be add on the database. |
| **Conclusion** | This use case concludes when the motorcycle admin will add a new mechanic. |
| **Postcondition** | The request has been recorded and if the add were available, the will response. For any request the admin will not get the user will not granted. |
| **Exceptions** | * The motorcycle user is no longer approved by double request. * The system asked the mechanics if it is done or to exit. * The Customer request another type of problem. * The Customer asked to exit   Appropriate message is displayed for every exception. |