**CSUTrike Use-Case Narrative for Passenger**

Author/s: Carrera, Dutdut, Los Baños, Ventura

Date: 31/3/25

Version: 1.0

Other types: passenger

|  |  |  |  |
| --- | --- | --- | --- |
| Use-Case Name : | **Login** | | Use-Case Type:  System Requirement |
| User-Case ID: | CSUTrike-UCAPassenger-1.00 | |
| Priority: | High | |
| Source: | Requirement – CSUTrike-SRv1 | |
| Primary Business Actor: | Passenger | | |
| Other Participating Actors: | Driver | | |
| Other Interested Actors: |  | | |
| Description: | This use-case describes the event of the user (passenger) logging into the system by providing their valid login credentials. And the user enable to access the services of the system. | | |
| Pre-condition: | * The user has a existing account. * The user has access to the internet. * The user has registered account. | | |
| Trigger: | This use case is executed if the user’s login. | | |
| Typical Course of Events: | Actor Action | System Action | |
| Step 1: The user (Passenger) enter the needed information in the login form. | Step 2: The System verify the credentials if its already save in the database of the system.  Step 3: If the credentials are valid, the user is Approved to access or redirect in the homepage | |
| Alternate Courses: | * If the credentials are incorrect, the user has receive a error notification. * If the user forget their password, they can change their password via “Forgot Password” option. | | |
| Conclusion: | Alt-Step 3: The user successfully logged in and it will redirect to the homepage/dashboard. | | |
| Post condition: | * If the credentials of the user are the same with the registered credentials it can automatically login, if it is not there is an error. * If the user are now in the dashboard of the system it can now book a ride. | | |
| Business Rules: | * The credentials must match the data stored in the database of the system. | | |
| Implementation Constraints and Specifications: | A mobile application that will access in android as well as the iOS that are provided to the users in order to successfully perform the features of the system such as book a ride, edit info, view history and more.. | | |
| Assumptions: | * The user must be familiar or memorize their credentials rigestered in the system. * The user must be correct credentials to easily logged in. | | |
| Open Issues: | * How long should the system allow a user to remain logged in without activity? | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Use-Case Name : | **Create Account** | | Use-Case Type:  System Requirement |
| User-Case ID: | CSUTrike-UCAPassenger-1.00 | |
| Priority: | High | |
| Source: | Requirement – CSUTrike-SRv1 | |
| Primary Business Actor: | Passenger | | |
| Other Participating Actors: | Driver | | |
| Other Interested Actors: |  | | |
| Description: | This use-case describes the event of the user (Passenger) to create a personal account by providing the necessary credencials and contact information, which is stored to the database of the system. | | |
| Pre-condition: | * User must be provide the information needed in the registration form. * The user has to access the registration form in order to login. | | |
| Trigger: | This use-case is triggered when the user are registered repeated. | | |
| Typical Course of Events: | Actor Action | System Action | |
| Step 1: The user must click the create account button from the landing page.  Step 2: The user must provide the information needed, Such as Full name, email, and password.  Step 3: The user will wait for the approval if the admin. | Step 4: The system verify the provided information  Step 5: If the verification’s correct, the system will create a new account and it will save it in the database.  Step 6: The system send a confirmation into the user.  Step 7: The user redirect to the login page. | |
| Alternate Courses: | Step 5: If the system detects the email is already used it will send a notification to the user and ask for another email to be use.  Step 6: If the user input a wrong information, it will receive a notification to correct the wrong information. | | |
| Conclusion: | If a user is no incorrect input it will successfully created an account and stored in the system and it will ready for login. | | |
| Post condition: | * If the new account approved the admin it will redirect to the login page. * The user’s account created is directed save in the database of the system. | | |
| Business Rules: | * The user must have a phone number cause it will required for registration. * The password must strongly password to ensure the security of the account. * The user cannot create a new account if it is already registered the credentials. | | |
| Implementation Constraints and Specifications: | A mobile application that will access in android as well as the iOS that are provided to the users in order to successfully perform the features of the system such as book a ride, edit info, view history and more.. | | |
| Assumptions: | * The user must correct and remember the password to login easily. * The user must did not forgot the credentials input ion the registration so there is no problem in the login page. | | |
| Open Issues: | What happen when the user cannot create new account even the credentials are not registered? | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Use-Case Name : | **Book Ride** | | Use-Case Type:  System Requirement |
| User-Case ID: | CSUTrike-UCAPassenger-1.00 | |
| Priority: | High | |
| Source: | Requirement – CSUTrike-SRv1 | |
| Primary Business Actor: | Passenger | | |
| Other Participating Actors: | Driver | | |
| Other Interested Actors: |  | | |
| Description: | This use-case describes the event of the user (Passenger) to book a ride using the CSUTrike system. The System process a booking and assigns a driver. And user has the ability to select the pickup point and drop-off locations. | | |
| Pre-condition: | * The user must logged in already before the booking. | | |
| Trigger: | The system will triggered if the booking is the simultaneously. | | |
| Typical Course of Events: | Actor Action | System Action | |
| Step 1: The user must click the book ride button.  Step 3: The user will confirm the booking, and it will choose the desired destination.  Step 5: The user receives a confirmation with a ride details after the system confirm. | Step 2: The System will prompt the user the capacity of the tricycle.  Step 4: The System process the booking and assign a driver. | |
| Alternate Courses: | * If the system are no available it will send a notification to the user and suggest try again. * If the passenger fails to confirm within a time limit the booking request will expired. | | |
| Conclusion: | The user successfully book a ride the system will assign a driver. | | |
| Post condition: | The user booking details will save and the assigned driver will notified. | | |
| Business Rules: | * The system riders must be efficient. | | |
| Implementation Constraints and Specifications: | A mobile application that will access in android as well as the iOS that are provided to the users in order to successfully perform the features of the system such as book a ride, edit info, view history and more.. | | |
| Assumptions: | The user must have a internet connection. | | |
| Open Issues: | Future integration with digital payment options. | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Use-Case Name : | **Cancel Ride** | | Use-Case Type:  System Requirement |
| User-Case ID: | CSUTrike-UCAPassenger-1.00 | |
| Priority: | High | |
| Source: | Requirement – CSUTrike-SRv1 | |
| Primary Business Actor: | Passenger | | |
| Other Participating Actors: | Driver | | |
| Other Interested Actors: |  | | |
| Description: | This use-case describes the event of the user (Passenger) to cancel the booked ride. | | |
| Pre-condition: | The user must a active booking. | | |
| Trigger: | The System triggered when the user cancel twice or three tries. | | |
| Typical Course of Events: | Actor Action | System Action | |
| Step 1: The user navigates t their booking details.  Step 2: The user click the Cancel ride button.  Step 4: The user will confirm the cancellation so that the driver will receive a notification that the booked ride are cancel. | Step 3: The system will notify that the ride are cancelled.  Step 5: The system will send a notification t the driver and updates the user’s booking history. | |
| Alternate Courses: | If the driver are on their way the booking will never be cancelled. | | |
| Conclusion: | If the user confirm the cancellation it will be successfully cancelled. | | |
| Post condition: | If the ride is cancelled it will automatically remove in the booking. | | |
| Business Rules: | Cancellations made after a certain time may incur a fee. | | |
| Implementation Constraints and Specifications: | A mobile application that will access in android as well as the iOS that are provided to the users in order to successfully perform the features of the system such as book a ride, edit info, view history and more.. | | |
| Assumptions: | The user must check always the internet connection to cancel the ride easily | | |
| Open Issues: | Should there be an automatic rebooking feature in case of cancellations?  It theirs a cancellation policies and cancelleation fee? | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Use-Case Name : | **View Current Booking Details** | | Use-Case Type:  System Requirement |
| User-Case ID: | CSUTrike-UCAPassenger-1.00 | |
| Priority: | High | |
| Source: | Requirement – CSUTrike-SRv1 | |
| Primary Business Actor: | Passenger | | |
| Other Participating Actors: | Driver | | |
| Other Interested Actors: |  | | |
| Description: | This use-case describes the event of the user (Passenger) to view the details of the active driver and also to view the available capacity seat. | | |
| Pre-condition: | The user must have an active booking in the system. | | |
| Trigger: | This triggered when the user didn’t have a booking already. | | |
| Typical Course of Events: | Actor Action | System Action | |
| Step 1: The user clicks the view current booking button. | Step 2: The system show the current booking details such as driver details, time, pickup point. | |
| Alternate Courses: | If there is no existing booking the system prompt the user that there is no current booking.  If cancellation is attempted too close to the ride time, the system prevents cancellation | | |
| Conclusion: | The user will successfully view the current booking. | | |
| Post condition: | The updated information is saved in the system. | | |
| Business Rules: | Only active users can be viewed the current booking. | | |
| Implementation Constraints and Specifications: | A mobile application that will access in android as well as the iOS that are provided to the users in order to successfully perform the features of the system such as book a ride, edit info, view history and more.. | | |
| Assumptions: | The user must have the active booking to make the system continues . | | |
| Open Issues: | What happen if the system fails to fetch booking details? | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Use-Case Name : | **Edit Information** | | Use-Case Type:  System Requirement |
| User-Case ID: | CSUTrike-UCAPassenger-1.00 | |
| Priority: | High | |
| Source: | Requirement – CSUTrike-SRv1 | |
| Primary Business Actor: | Passenger | | |
| Other Participating Actors: | Driver | | |
| Other Interested Actors: |  | | |
| Description: | This use-case describes the event where the user can modify the personal details such as name, contact number, or profile information while ensuring the integrity and security. | | |
| Pre-condition: | The user must logged into their account before modify the profile information. | | |
| Trigger: | This will triggered when the user wants to modify their personal information for accuracy or updates. | | |
| Typical Course of Events: | Actor Action | System Action | |
| Step 1: The user navigates to the profile settings.  Step 3: The user will modify only the necessary information.  Step 5: The user will receives a notification that thre update was successfully. | Step 2: The system will display the editable fields.  Step 4: The system will save the changes if the input are valid. | |
| Alternate Courses: | If the user entered an incorrect data the system will promp the user to input correct data. | | |
| Conclusion: | The Passenger successfully updates their personal details, ensuring accurate and up-to-date information within the system. | | |
| Post condition: | The Passenger's updated details are stored securely. | | |
| Business Rules: | Certain data may require verification before being updated. | | |
| Implementation Constraints and Specifications: | A mobile application that will access in android as well as the iOS that are provided to the users in order to successfully perform the features of the system such as book a ride, edit info, view history and more.. | | |
| Assumptions: | The user will make sure that the internet connection are stable. | | |
| Open Issues: | The need for an approval process for sensitive data changes. | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Use-Case Name : | **View Booking History** | | Use-Case Type:  System Requirement |
| User-Case ID: | CSUTrike-UCAPassenger-1.00 | |
| Priority: | High | |
| Source: | Requirement – CSUTrike-SRv1 | |
| Primary Business Actor: | Passenger | | |
| Other Participating Actors: | Driver | | |
| Other Interested Actors: |  | | |
| Description: | This use-case describes the event where the user wants to review their past rides for reference or record keeping. The system retrieves and display previous booking details. | | |
| Pre-condition: | The user must have at least 1 completed ride. | | |
| Trigger: | This use case triggered when the user wants to check past bookings. | | |
| Typical Course of Events: | Actor Action | System Action | |
| Step 1: The user will click the view booking history  Step 3: The user view the booking details such as ride date and time, face, and destination. | Step 2: The system will display all the past ride. | |
| Alternate Courses: | If no previous rides exist, the system notifies the user that there is no booking history available. | | |
| Conclusion: | The user successfully view the history of booking. | | |
| Post condition: | The booking history remains available for future reference | | |
| Business Rules: | Booking history should be stored for a specific period (e.g., 6 months). | | |
| Implementation Constraints and Specifications: | A mobile application that will access in android as well as the iOS that are provided to the users in order to successfully perform the features of the system such as book a ride, edit info, view history and more.. | | |
| Assumptions: | The user remembers their login credentials to access history | | |
| Open Issues: | The ability to export or print booking history may be required in the future. | | |