**CSUTrike Use-Case Narrative for Passengers**

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

Date: 01/04/25

Version: 1.0

Other types: user story, business need

|  |  |  |  |
| --- | --- | --- | --- |
| Use-Case Name : | Book a Driver | | Use-Case Type:  System Requirement |
| User-Case ID: | CSUTrike-UCPassenger-1.00 | |
| Priority: | High | |
| Source: | Requirement – CSUTrike-SRv1 | |
| Primary Business Actor: | Passengers | | |
| Other Participating Actors: | •Driver  •Automated Booking Management | | |
| Other Interested Actors: |  | | |
| Description: | This use case allows a student to book a trike without selecting a specific driver. The system automatically assigns the ride to the first available driver in the queue. The driver who logs in and is first in the queue will be assigned the booking request. | | |
| Pre-condition: | * The student must be logged into the system. * The system must have at least one available trike driver logged in. | | |
| Trigger: | This use-case is triggered when a **student initiates a trike booking request** by selecting the **"Book Trike"** option in the web or mobile application. | | |
| 1. Typical Course of Events: | Actor Action | System Action | |
| Step 1: Passenger selects the **"Book Trike"** option in the application.  Step 5: The driver accepts the booking and proceeds to the pickup location.  Step 7: The driver picks up the passenger and completes the ride. | Step 2: The system checks the queue for available trike drivers.  Step 3: The system assigns the booking to the **first driver** in the queue.  Step 4: The system sends a notification to the assigned driver.  Step 6: The system notifies the passenger of the assigned driver’s details and estimated arrival time. Step 8: The system updates the ride status to **"Completed"** and logs the transaction. | |
| Alternate Courses: | Alt-Step 2: If no drivers are available, the system displays a message informing the passenger to try again later.  Alt-Step 6: If the passenger cancels the booking before pickup, the system updates the queue and notifies the drivers. | | |
| Conclusion: | The use-case concludes when the passenger successfully books and completes the ride, or when the request is canceled. | | |
| Post condition: | * Passenger cannot choose a specific driver; assignments are based on queue order. * Drivers must be logged in to receive bookings. * If a driver logs out or becomes unavailable before accepting a ride, they are removed from the queue. | | |
| Business Rules: | * By default, when the request is sent, it is pending. * If an existing account is detected, a warning alert will be issued to the Admin. | | |
| Implementation Constraints and Specifications: | A web-based and mobile application will be provided to students to successfully book a trike. The application can be accessed using modern browsers such as Google Chrome, Microsoft Edge, or Mozilla Firefox on desktops, and via native mobile apps on smartphones. | | |
| Assumptions: | * The Passenger provides all necessary information for the request. * The system maintains an accurate, real-time queue of available drivers with valid credentials. | | |
| Open Issues: | 1. If a passenger submits a new booking request before the previous request is processed, should the new request overwrite the pending request or be queued separately? | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Use-Case Name : | Cancel Rides | | Use-Case Type:  System Requirement |
| User-Case ID: | CSUTrike-UCpassenger-1.01 | |
| Priority: | Medium | |
| Source: | Requirement – CSUTrike-SRv1 | |
| Primary Business Actor: | Passenger | | |
| Other Participating Actors: | Drivers | | |
| Other Interested Actors: |  | | |
| Description: | The use-case describes the process by which a passenger cancels a booked ride before the driver accept the request. The cancellation request updates the system, notifies the assigned driver, and releases the driver back into the queue for other bookings. | | |
| Pre-condition: | * The passenger must have an active booking. * Drivers must have accept the booking (cancellation is possible only if the driver accept the booking ). | | |
| Trigger: | A passenger initiates a ride cancellation request by selecting the cancellation button in the web or mobile application. | | |
| Typical Course of Events: | Actor Action | System Action | |
| Step 1: The passenger selects the cancellation button in the application.  Step 5: The passenger receives a confirmation that the ride has been successfully canceled. | Step 2: The system verifies that the ride is still in the pending or assigned state.  Step 3: The system updates the booking status to “Cancelled”.  Step 4: The system notifies the assigned driver about the cancellation. | |
| Alternate Courses: | Alt-Step 2: If the ride has already started or accepted, the system prevents cancellation and notifies the student that cancellation is no longer available.  Alt-Step 4: If the driver has already arrived at the pickup location, the system may notify the passenger of potential penalties for late cancellation (if applicable hehe). | | |
| Conclusion: | The use-case concludes when the passenger’s ride is successfully cancelled, and the system updates the ride status to “**cancelled**”. | | |
| Post condition: | * The ride is removed from the student’s active bookings. * The driver’s status is updated to “available” in the queue. * A cancellation record is stored for future reference. | | |
| Business Rules: | * The passenger may only cancel a ride before the driver accept the booking. * The system may impose a cancellation limit or penalty if passenger frequently cancel rides. * Drivers must be notified immediately when a ride is canceled. | | |
| Implementation Constraints and Specifications: | A web-based and mobile application will be provided to passenger to cancel their ride requests. The application can be accessed using modern browsers such as Google Chrome, Microsoft Edge, or Mozilla Firefox on desktop, and via a mobile app on smartphones. | | |
| Assumptions: | * The passenger has a valid internet connection to process the cancellation request. * The driver receives notifications instantly upon ride cancellation. | | |
| Open Issues: | 1. Should passenger have limited number of cancellations per day to prevent misuse? | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Use-Case Name : | Edit Information | | Use-Case Type:  System Requirement |
| User-Case ID: | CSUTrike-UCpassenger-1.02 | |
| Priority: | medium | |
| Source: | Requirement – CSUTrike-SRv1 | |
| Primary Business Actor: | Passenger | | |
| Other Participating Actors: |  | | |
| Other Interested Actors: |  | | |
| Description: | The use-case describes the event where a passengers can update their personal information, such as name, contact details, and profile picture, within the system. | | |
| Pre-condition: | * The passenger must be logged into their account. * The system must have the existing student profile data. | | |
| Trigger: | This use-case is triggered when a Passenger initiates a profile update by selecting the “Edit profile” option in the web or mobile application. | | |
| Typical Course of Events: | Actor Action | System Action | |
| Step 1: The passenger navigates to the “Edit profile” selection in the application.  Step 2: The passenger updates their personal details (name, email, phone number, profile picture). | Step 3: The system validates the updated information (checks for prorper formatting and required fields).  Step 4: If all inputs are valid, the system updates the student’s profile.  Step 5: The system displays a confirmation message indicating that the changes were saved successfully. | |
| Alternate Courses: | Alt-Step 3: If the input validations fails (invalid format or missing required fields), the system prompts the passenger to correct the errors before proceeding.  Alt-Step 4: If the system encounters an error while saving the changes (database failure), it notifies the passenger and suggest retrying later. | | |
| Conclusion: | The use-case concludes when the passenger’s updated profile information is successfully saved in the system. | | |
| Post condition: | * The passenger’s profile information is updated in the system. * The changes are reflected in the future interactions with the applications. | | |
| Business Rules: | * The system should ensure that email addresses and phone numbers follow a valid format. * The passenger cannot edit restricted fields such as their Student ID. | | |
| Implementation Constraints and Specifications: | A web-based and mobile application will be provided to the passenger in order to edit their profile which can be accessed using Google Chrome, Microsoft Edge, or Mozilla Firefox. | | |
| Assumptions: | * The passenger has a stable internet connections to submit the updated information. * The system accurately retrieves and updates the passenger’s profile data. | | |
| Open Issues: | 1. Should there be restrictions on how often passengers can edit their information within a specific timeframe. | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Use-Case Name : | View (recent bookings) | | Use-Case Type:  System Requirement |
| User-Case ID: | CSUTrike-UCpassenger-1.03 | |
| Priority: | Medium | |
| Source: | Requirement – CSUTrike-SRv1 | |
| Primary Business Actor: | Passenger | | |
| Other Participating Actors: |  | | |
| Other Interested Actors: |  | | |
| Description: | The use-case describes the event where passengers are allowed to view their recent trike booking history, including past and ongoing ride details. The system retrieves and displays the relevant booking records, providing information such as ride status, date, time, and assigned driver. | | |
| Pre-condition: | * The student must be logged into their account. * The system must have at least one past or ongoing booking associated with the student. | | |
| Trigger: | This use-case trigger occurs when a passenger initiates a request to view the recent booking history. | | |
| Typical Course of Events: | Actor Action | System Action | |
| Step 1: The passenger navigates to the recent bookings section in the application.  Step 4: The passenger reviews the list of bookings and selects a specific booking if they want to view more details. | Step 2: The system retrieves the passenger’s booking history from the database.  Step 3: The system displays a list recent bookings, including details such as ride date, time status(completed, ongoing, Cancelled), and assigned driver information.  Step 5: The system displays the full details of the selected booking. | |
| Alternate Courses: | Alt-Step 2: If the system encounters an error retrieving booking history (e.g., database failure), it displays an error message and prompts the student to retry later.  Alt-Step 3: If no booking history is found, the system displays a message indicating that no recent bookings are available. | | |
| Conclusion: | The use-case concludes when the passenger’s recent booking history is successfully retrieved and displayed in the system. | | |
| Post condition: | * The passenger's booking history is successfully retrieved and displayed. * The student gains access to details of past and ongoing rides. | | |
| Business Rules: | * Passengers can only view their own booking histpry. * The system must protect sensitive driver details and ensure compliance with privacy policies. | | |
| Implementation Constraints and Specifications: | A web-based application will be provided to the passengers in order to view their activities history which can be accessed using Google Chrome, Microsoft Edge, or Mozilla Firefox. | | |
| Assumptions: | * The passenger has a stable internet connection to retrieve booking history. * The system accurately records and updates booking details in real time. | | |
| Open Issues: | 1. How long should completed and canceled bookings remain visible before being archived? | | |