**Use Cases**

**for**

**<**SC2006 **Project>**

**Version 2.0 approved**

**Prepared by <A27>**

**<Nanyang Technological University>**

**<12 Feb 2023>**

**Revision History**

| **Name** | **Date** | **Reason For Changes** | **Version** |
| --- | --- | --- | --- |
| A27 | 29 Jan 2023 | - | 1.0 |
| A27 | 12 Feb 2023 | Finalise the use case descriptions | 2.0 |

# **Guidance for Use Case Template**

Document each use case using the template shown in the Appendix. This section provides a description of each section in the use case template.

# **Use Case Identification**

## **Use Case ID**

Give each use case a unique numeric identifier, in hierarchical form: X.Y. Related use cases can be grouped in the hierarchy. Functional requirements can be traced back to a labeled use case.

## **Use Case Name**

State a concise, results-oriented name for the use case. These reflect the tasks the user needs to be able to accomplish using the system. Include an action verb and a noun. Some examples:

* View part number information.
* Manually mark hypertext source and establish link to target.
* Place an order for a CD with the updated software version.

## **Use Case History**

### **Created By**

Supply the name of the person who initially documented this use case.

### **Date Created**

Enter the date on which the use case was initially documented.

### **Last Updated By**

Supply the name of the person who performed the most recent update to the use case description.

### **Date Last Updated**

Enter the date on which the use case was most recently updated.

# **Use Case Definition**

## **Actor**

An actor is a person or other entity external to the software system being specified who interacts with the system and performs use cases to accomplish tasks. Different actors often correspond to different user classes, or roles, identified from the customer community that will use the product. Name the actor(s) that will be performing this use case.

## **Description**

Provide a brief description of the reason for and outcome of this use case, or a high-level description of the sequence of actions and the outcome of executing the use case.

## **Preconditions**

List any activities that must take place, or any conditions that must be true, before the use case can be started. Number each precondition. Examples:

1. User’s identity has been authenticated.
2. User’s computer has sufficient free memory available to launch task.

## **Postconditions**

Describe the state of the system at the conclusion of the use case execution. Number each postcondition. Examples:

1. Document contains only valid SGML tags.
2. Price of item in database has been updated with new value.

## **Priority**

Indicate the relative priority of implementing the functionality required to allow this use case to be executed. The priority scheme used must be the same as that used in the software requirements specification.

## **Frequency of Use**

Estimate the number of times this use case will be performed by the actors per some appropriate unit of time.

## **Flow of Events**

Provide a detailed description of the user actions and system responses that will take place during execution of the use case under normal, expected conditions. This dialog sequence will ultimately lead to accomplishing the goal stated in the use case name and description. This description may be written as an answer to the hypothetical question, “How do I <accomplish the task stated in the use case name>?” This is best done as a numbered list of actions performed by the actor, alternating with responses provided by the system.

## **Alternative Flows**

Document other, legitimate usage scenarios that can take place within this use case separately in this section. State the alternative course, and describe any differences in the sequence of steps that take place. Number each alternative flow using “AF-S” followed by the step number where the alternative flow diverts from the normal flow. For example, if an alternative flow differs from the normal flow starting from step 5, then the alternative flow should be titled as “AF-S5”.

## **Exceptions**

Describe any anticipated error conditions that could occur during execution of the use case, and define how the system is to respond to those conditions. Also, describe how the system is to respond if the use case execution fails for some unanticipated reason. Number each exception using “EX” to indicate “Exception”, followed by the exception number for that use case. For example, if a use case has 2 exceptions, the first one should be titled as “EX1” and the second one titled as “EX2”.

## **Includes**

List any other use cases that are included (“called”) by this use case. Common functionality that appears in multiple use cases can be split out into a separate use case that is included by the ones that need that common functionality.

## **Special Requirements**

Identify any additional requirements, such as nonfunctional requirements, for the use case that may need to be addressed during design or implementation. These may include performance requirements or other quality attributes.

## **Assumptions**

List any assumptions that were made in the analysis that led to accepting this use case into the product description and writing the use case description.

## **Notes and Issues**

List any additional comments about this use case or any remaining open issues or TBDs (To Be Determineds) that must be resolved. Identify who will resolve each issue, the due date, and what the resolution ultimately is.

**Use Case Template**

| Use Case ID: | 1.0 | | |
| --- | --- | --- | --- |
| Use Case Name: | Manage Bookings | | |
| Created By: | Wai Man | Last Updated By: | Wai Man |
| Date Created: | 29 Jan 2023 | Date Last Updated: | 8 Feb 2023 |

| Actor: | User, Database |
| --- | --- |
| Description: | This use case handles actions including create bookings, reject bookings, accept bookings and cancel bookings. |
| Preconditions: | 1. User must have an account and be able to login into the system successfully 2. If a renter wants to cancel a booking, he/she must have at least one confirmed booking record in the system |
| Postconditions: | 1. A success message (notification) must be shown if the booking has been created/accepted/rejected/canceled successfully. 2. The renting post must be hidden from the available list to the public once the booking is confirmed 3. Listing status must be updated    1. Rented    2. Pending (wait for approve)    3. Available 4. All the listings will be stored in database |
| Priority: | High |
| Frequency of Use: | High |
| Flow of Events: | 1. User login into the system using his login credentials 2. User selects a desire location (or current location) to find the nearby car parks 3. User select a car park to view the cars available for rent 4. User select a car he wants to view the details 5. User click the “Book” button to request for booking 6. The status of the post will be changed to “Pending” 7. The car owner will receive a notification for the booking request 8. The car owner accept one request from all the requests by clicking the “Accept” button 9. The listing post will be hidden from the public, and a notification will be sent to both car owner and renter 10. The renter can cancel the booking within 12 hours by clicking the “Cancel” button 11. A notification will be sent to the car owner that the booking has been canceled |
| Alternative Flows: | AF-S5:   1. The car owner reject booking requests by clicking the “Reject” button 2. A notification will be sent to the renter that the booking request has been rejected   AF-S8: If car owner accept more than one request   1. a message will be prompted to block the car owner to accept other requests   AF-S9 : If the booking has been accepted by the car owner for more than 12 hours   1. The cancel button will be disabled or hidden 2. Renter cannot get refund |
| Exceptions: | EX1: If an error happens when creating / rejecting / accepting / canceling a booking   1. an error message and error code must be shown to notify users   EX2: If the listing (rent car post) has been deleted or the status changed to “Rented”   1. An alert will be prompted to notify the renter that the car is no longer available for renting |
| Includes: | Create booking, reject booking, accept booking and cancel booking |
| Special Requirements: | Both renter and car owner must be aged 18 and above with valid driving license |
| Assumptions: | 1. All necessary information (age, driving license, etc.) has been verified when user creating an account 2. User uses the booking system for renting. If not, it is hard to manage the bookings created by users |
| Notes and Issues: | - |

| Use Case ID: | 2.0 | | |
| --- | --- | --- | --- |
| Use Case Name: | Chat, Database | | |
| Created By: | Wai Man | Last Updated By: | Wai Man |
| Date Created: | 29 Jan 2023 | Date Last Updated: | 8 Feb 2023 |

| Actor: | User, Database |
| --- | --- |
| Description: | Chatting system allows users (renter and car owner) to communicate with each other. This helps to enhance the overall user experience as the renter can use it to clarify any queries with the car owner directly |
| Preconditions: | 1. User must have an account and be able to login into the system successfully 2. User found the car he wants and start chatting by clicking the “chat” button in the car detail page |
| Postconditions: | new message notification will be sent to the receiver |
| Priority: | Low |
| Frequency of Use: | Medium |
| Flow of Events: | 1. User select a car park to view the cars available for rent 2. User select a car he wants to view the details 3. User click the “Chat” button in car details page to start a chat 4. Chat history will be stored in the database |
| Alternative Flows: | - |
| Exceptions: | EX1: If an error happens in chatting (the message cannot be sent out, etc.)   1. an error message and error code must be shown to notify users |
| Includes: | - |
| Special Requirements: | A chatting system extension/plug-in is needed for this feature since it is difficult for us to build a chatting system from scratch |
| Assumptions: | - |
| Notes and Issues: | - |

| Use Case ID: | 3.0 | | |
| --- | --- | --- | --- |
| Use Case Name: | Rating and Review | | |
| Created By: | Woon Keng | Last Updated By: | Wai Man |
| Date Created: | 29 Jan 2023 | Date Last Updated: | 9 Feb 2023 |

| Actor: | User, Database |
| --- | --- |
| Description: | This use case allows renters and owners to provide a review for their rental experience which will help future renters make informed decisions about which vehicle to rent and future owners to decide who to rent to. |
| Preconditions: | The trip that the renter and owner are involved in has been completed. |
| Postconditions: | The renter’s and owner’s ratings and reviews are displayed in the application, helping future renters and owners make informed rental decisions. |
| Priority: | Medium |
| Frequency of Use: | High |
| Flow of Events: | 1. The application prompts both the renter and the owner to leave a review via a notification in the application. 2. The renter and owner click on the notification and are brought to a review form. 3. The renter and owner rate the rental experience on a scale of 1-5 stars and can provide an optional written review to provide details like the condition of the vehicle, communication experience and any other related issues. 4. The renter and owner click on the “Submit Review” button at the bottom of the review form. 5. The review and rating are submitted and displayed in the application, where others can view them. |
| Alternative Flows: | AF-S3: If the user attempts to write a review after the allowed review window of 10 days after the completion of the rental.   1. The application displays an error message indicating that the review and rating must be submitted within the 10 day review window after the rental has ended. 2. The user will not be able to enter the review form page and will remain on the main notification page. |
| Exceptions: | - |
| Includes: | - |
| Special Requirements: | * The renter and the owner must submit a review within 10 days of the completion of the rental if they wish to submit a review. They will not be able to leave a review after 10 days. |
| Assumptions: | * The user decides to leave a review. It is not mandatory to leave a review after a rental. |
| Notes and Issues: | - |

| Use Case ID: | 4.0 | | |
| --- | --- | --- | --- |
| Use Case Name: | Manage Car Listings | | |
| Created By: | Woon Keng | Last Updated By: | Wai Man |
| Date Created: | 29 Jan 2023 | Date Last Updated: | 9 Feb 2023 |

| Actor: | User, Database |
| --- | --- |
| Description: | This use case will cover all the aspects of listings in the application, including the addition, deletion and editing of listings by users. |
| Preconditions: | * The user has registered an account and logged in to the application. * The user must have at least one listing if he wants to edit or delete |
| Postconditions: | The listing will be created/updated/deleted |
| Priority: | High |
| Frequency of Use: | Medium |
| Flow of Events: | 1. The user navigates to his Profile page in the app. 2. All of the user’s active listings will be shown under the “Listings” tab 3. Click the “+" button at the bottom center 4. The user needs to fill up the car information form 5. Click the “Post” button to submit listing 6. The listing will then be shown in the car renting list |
| Alternative Flows: | AF-S4: If the user has not provided all the required information   1. The application displays the message “Please enter all the required information!” 2. The user returns to step 2) in normal flow   AF-S3: If the user wants to update the existing listing   1. Select the listing that he wants to edit 2. Click the “Edit” button 3. The user makes the edits he wants in the detail fields. 4. The user clicks on the “Save” button when he is finished. 5. The application registers the changes, closes the edited listing and returns to the listing management page.   AF-S2: If the user wants to delete an existing listing   1. The user selects the listing that he wants to delete 2. The application displays the full listing with its details along with a “Delete listing” button at the bottom of the page 3. The user clicks on the “Delete listing” button. 4. The application asks the user to confirm his action with a warning that the action is not reversible. 5. The user clicks “Confirm”. |
| Exceptions: | - |
| Includes: | Add Listings, Delete Listings, Edit Listings |
| Special Requirements: | - |
| Assumptions: | * The user has all the necessary information and photos to complete the listing |
| Notes and Issues: | - |

\

| Use Case ID: | 5.0 | | |
| --- | --- | --- | --- |
| Use Case Name: | Payment | | |
| Created By: | Woon Keng | Last Updated By: | Woon Keng |
| Date Created: | 29 Jan 2023 | Date Last Updated: | 29 Jan 2023 |

| Actor: | User, Database |
| --- | --- |
| Description: | This use case describes the process of payment that occurs whenever a user attempts to book a car. |
| Preconditions: | 1. The user has selected a rental vehicle and confirmed the booking details. 2. The user has a valid payment method on file. |
| Postconditions: | 1. The user’s payment is processed and the rental is confirmed. 2. The user’s payment information is securely stored in the application system. 3. The owner of the vehicle received notification of the payment. 4. Transaction history will be stored in the database |
| Priority: | High |
| Frequency of Use: | High |
| Flow of Events: | 1. The application displays the payment page, which includes the rental vehicle details, the rental period, and the total cost. 2. The user reviews the payment details and confirms they are correct. 3. The user selects their preferred payment method from the list of options (e.g., credit card, PayPal, etc.). 4. The user clicks on the “Pay” button. 5. The application processes the payment using the external Payment System and displays a confirmation message. 6. The application sends a confirmation email to the user’s email address and notifies the vehicle's owner of the successful payment. |
| Alternative Flows: | AF-S5: The payment is rejected by the external payment system   1. The application displays the relevant error message 2. The user returns to step 3) in normal flow |
| Exceptions: | EX1: If the external payment system is down   1. The application displays the message “Sorry, we are currently experiencing technical difficulties with our payment system. Please try again later or try a different payment method.” 2. The user returns to step 3) in normal flow |
| Includes: | - |
| Special Requirements: | - |
| Assumptions: | * The user has access to a valid payment method * The use case also assumes that the application has a secure payment gateway integration to process the payments * The use case also assumes that the application has a notification system in place to inform the owner of the rental vehicle about the payment |
| Notes and Issues: | * This use case assumes that the payment is done before the rental, but in some cases, the payment might be done after the rental is completed |

| Use Case ID: | 6.0 | | |
| --- | --- | --- | --- |
| Use Case Name: | Rent Car | | |
| Created By: | Kavithran | Last Updated By: | Kavithran |
| Date Created: | 29 Jan 2023 | Date Last Updated: | 11 Feb 2023 |

| Actor: | User |
| --- | --- |
| Description: | A user can rent another user’s car that is listed for rent. |
| Preconditions: | 1. User has a verified account in the system. |
| Postconditions: | 1. User successfully rents a car from the car park. (AND) 2. User successfully returns the rented car to the car park. |
| Priority: | - |
| Frequency of Use: | - |
| Flow of Events: | 1. User picks a location (current location or searches the location he wants) 2. System displays the nearest car parks to the location. 3. User selects preferred car park. 4. System displays available cars for rent at the selected car park. 5. User selects preferred car for rent. 6. If the car owner accepts the user’s rental request, the user can rent the car. 7. User returns the car back to the same car park after renting it. |
| Alternative Flows: | AF-S5: If there are no cars available for rent at the selected car park,   1. The system displays “Cars unavailable. Please try another car park”. 2. System returns to step 3.   AF-S7: If the car owner rejects the user’s rent request,   1. The system sends a notification to the user, “Your request has been rejected, please try another car”. 2. System returns to step 3. |
| Exceptions: | - |
| Includes: | Retrieve Picked Location, Retrieve Carpark Location |
| Special Requirements: | - |
| Assumptions: | User has valid driving license |
| Notes and Issues: | - |

| Use Case ID: | 6.1 | | |
| --- | --- | --- | --- |
| Use Case Name: | Retrieve Picked Location | | |
| Created By: | Kavithran | Last Updated By: | Kavithran |
| Date Created: | 29 Jan 2023 | Date Last Updated: | 11 Feb 2023 |

| Actor: | User, Geolocation API |
| --- | --- |
| Description: | Retrieve the user’s picked location. |
| Preconditions: | 1. User selects to rent a car. 2. Mobile location services must be on. |
| Postconditions: | 1. System retrieves the user-picked location |
| Priority: | - |
| Frequency of Use: | - |
| Flow of Events: | 1. User picks a location 2. The system retrieves location information through a Geolocation API 3. System displays nearby car parks of the location on the map |
| Alternative Flows: | - |
| Exceptions: | EX1: Mobile phone’s location not turned on.   1. System will prompt the user to turn on location services. |
| Includes: | - |
| Special Requirements: | - |
| Assumptions: | 1. Geolocation API will return the most accurate user’s location. |
| Notes and Issues: | 1. Need to further work on how cars for rent will be retrieved 2. User-picked location refers to either the current user’s location, or another location searched by the user |

| Use Case ID: | 6.2 | | |
| --- | --- | --- | --- |
| Use Case Name: | Retrieve Carpark Location | | |
| Created By: | Kavithran | Last Updated By: | Kavithran |
| Date Created: | 29 Jan 2023 | Date Last Updated: | 11 Feb 2023 |

| Actor: | User, Carpark-Location API |
| --- | --- |
| Description: | Retrieves a list of car parks near selected location. |
| Preconditions: | 1. Mobile’s location services must be on. 2. Mobile connected to internet services. |
| Postconditions: | 1. Carpark-Location API returns a list of car parks to the system. |
| Priority: | - |
| Frequency of Use: | - |
| Flow of Events: | 1. User picks a location 2. The system gets car park locations through a Carpark-Location API. 3. System filters away car parks that are not near the user-picked location 4. System displays nearby car parks of the picked location on the map |
| Alternative Flows: | - |
| Exceptions: | EX1: Mobile phone’s location not turned on.   1. System will prompt the user to turn on location services. |
| Includes: | - |
| Special Requirements: | - |
| Assumptions: | 1. Carpark-Location API will return the most accurate car park locations near the user. |
| Notes and Issues: | 1. Need to work on which data to extract from the API and how to use it to find the nearest car parks to the user. 2. User-picked location refers to either the current user’s location, or another location searched by the user. |

| Use Case ID: | 6.3 | | |
| --- | --- | --- | --- |
| Use Case Name: | Display locations on map | | |
| Created By: | Aki | Last Updated By: | Wai Man |
| Date Created: | 29 Jan 2023 | Date Last Updated: | 9 Feb 2023 |

| Actor: | Google Maps’ API |
| --- | --- |
| Description: | Display the selected location and nearby car park on map |
| Preconditions: | 1. An internet connection must be established 2. Information about the user-selected location has been retrieved successfully 3. Car parks information has been retrieved successfully. |
| Postconditions: | User can select the car park he wants to find cars to rent |
| Priority: | High. |
| Frequency of Use: | High. |
| Flow of Events: | 1. The user-picked location will be located at the center of the map with a marker 2. Display the nearest car parks with markers 3. The user is able to pan around and zoom in or out of the map to view the location of the car parks. 4. The user will be able to view information about the selected car park after tapping on the marker. |
| Alternative Flows: |  |
| Exceptions: |  |
| Includes: |  |
| Special Requirements: |  |
| Assumptions: |  |
| Notes and Issues: |  |

| Use Case ID: | 7.0 | | |
| --- | --- | --- | --- |
| Use Case Name: | Create Account | | |
| Created By: | Aki | Last Updated By: | Aki |
| Date Created: | 29 Jan 2023 | Date Last Updated: | 29 Jan 2023 |

| Actor: | User |
| --- | --- |
| Description: | This use case allows users to create an account to be used for our application. |
| Preconditions: | The user selects to register for a new account. |
| Postconditions: | New User account is successfully registered in the system database.  The user can now log in and start using the services offered by our application. |
| Priority: | Medium |
| Frequency of Use: | Used only by first time users |
| Flow of Events: | 1. The User chooses to register for a new account. 2. On selection the user will be led to the appropriate outcome 3. For the ‘Register’ button, the user will be redirected to the Register page with 4 input fields (Username, Email, Password, and Confirm Password) 4. On selection, the user will be led to the appropriate outcome    1. The user enters a username, email and password    2. The system checks if the email fulfills the email criteria (Valid Email)    3. The system then checks if the password fulfills the minimal criteria of having at least 8 characters    4. The system checks if the confirmed password entered is the same as the password entered 5. A suitable error message is displayed under the field if any of the conditions above are not satisfied 6. If any fields are invalid, the ‘Register’ button cannot be pressed 7. The user clicks the “Register” button 8. The system adds the new user record to the MongoDB database 9. The website prompts the success message “Registration complete. You will now be redirected to the login page” 10. The website redirects users to the start page |
| Alternative Flows: | If the user selects an existing username:  The system will display an error message ‘The User already exists’ |
| Exceptions: |  |
| Includes: |  |
| Special Requirements: |  |
| Assumptions: | The customer is a first time user  The system is bug free |
| Notes and Issues: |  |