**UNIVERSITI TUNKU ABDUL RAHMAN**

**LEE KONG CHIAN FACULTY OF ENGINEERING AND SCIENCE**

**UECS2344 SOFTWARE DESIGN**

**ASSIGNMENT (Part 1, Part 2 and Part 3)**



**January 2021 Trimester**

|  |  |  |  |
| --- | --- | --- | --- |
| **NO**. | **STUDENT NAME** | **STUDENT ID** | **PRACTICAL GROUP** |
| **1** | **NG KHEE LONG** | **1802451** | **P7** |
| **2** | **GERVIN FUNG DA XUAN** | **1801655** | **P7** |
| **3** | **GOH PEI JIN** | **1802410** | **P6** |
| **4** | **KOH RONG SOON** | **1802575** | **P7** |
| **5** | **TAY MING LIANG** | **1804612** | **P7** |

**Table of Contents**

[Assumptions 1](#_Toc68830970)

[Functional Requirements 1](#_Toc68830971)

[Analysis class diagram 2](#_Toc68830972)

[Use Case Diagram 3](#_Toc68830973)

[Use Case Description 4](#_Toc68830974)

[Use Case Name: Make a Booking 4](#_Toc68830975)

[Use Case Name: View All Booking Details 5](#_Toc68830976)

[Use Case Name: Update Booking Details 6](#_Toc68830977)

[Use Case Name: Cancel Booking 7](#_Toc68830978)

[Use Case Name: Search Booking Details 8](#_Toc68830979)

[Use Case Name: Log in 11](#_Toc68830980)

[Use Case Name: Create New User Profile 12](#_Toc68830981)

[Use Case Name: Edit Booking Information 13](#_Toc68830982)

[Sequence Diagram 15](#_Toc68830983)

[Cancel Booking 15](#_Toc68830984)

[Create New User Profile 16](#_Toc68830985)

[Edit Booking Information 18](#_Toc68830986)

[Log In 24](#_Toc68830987)

[Make a Booking 25](#_Toc68830988)

[Search Booking Details 26](#_Toc68830989)

[Update Booking Details 30](#_Toc68830990)

[View All Booking Details 31](#_Toc68830991)

[Class Diagram 43](#_Toc68830992)

[Version 1 43](#_Toc68830993)

[Version 2 44](#_Toc68830994)

[Package Diagram 45](#_Toc68830995)

# Assumptions

1. There will be only one admin handling the amusement park management system.
2. User ID is auto-generated when a user creates a new user.
3. For a booking, users can book all of the items on the same item date only.
4. All items will always be effective and its date can be changed even after the item is removed from the system.

# Functional Requirements

1. The application shall allow users to login into their account.
2. The application shall allow new users to register a new account.
3. The application shall allow client users to make booking.
4. The application shall allow client users to view their own booking records.
5. The application shall allow client users to cancel their own bookings.
6. The application shall allow client users to update their own bookings’ details by date.
7. The application shall allow client users to search their own bookings.
8. The application shall allow admin users to view booking records of all users.
9. The application shall allow admin users to cancel booking of all users.
10. The application shall allow admin users to search booking records of all users.
11. The application shall allow admin users to search all user’s profiles.
12. The application shall allow admin users to create new admin users’ accounts.
13. The application shall allow admin users to add new items for booking.
14. The application shall allow admin users to add, delete and change details of items available for booking.

# Analysis class diagram



# Use Case Diagram

Diagram

Description automatically generated



# Use Case Description



|  |  |  |
| --- | --- | --- |
| Use Case Name: Make a Booking | ID: 1 | Importance Level : **High** |
| Primary Actor: **Client** | Use Case Type: detail, essential | |
| Stakeholders and Interests:  Client - Client wants to make a booking for the amusement park online. | | |
| Brief Description:  This use case describes how **Client** uses the application to make a booking for the amusement park | | |
| Trigger:  Client wants to make a booking for the amusement park. | | |
| Relationships:           Association: Client           Include:N/A           Extend: N/A           Generalization: N/A | | |
| Normal Flow of Events:   1. Client login to the amusement park management system. 2. Client ***selects*** make a booking option. 3. The system displays all the items available for booking. 4. Client ***enters*** the date they want to book.    1. If the date is invalid   The S3a: retry date entry step is performed.   1. Client ***chooses*** the desired item that he/she wants to book. 2. Client ***enters*** the number of items he/she wants to book. 3. The application system adds the item into the list of items booked. 4. The system displays the current booking details including item booked and quantity of each item booked.    1. If client wants to add more items,   Step 5. is performed.   1. If client confirms to proceed with the booking,   Use case terminates.   1. If client wants to cancel booking,   The E7b: booking cancel is performed. | | |
| SubFlows:  **S3a: retry date entry**   1. The system shows that the date is invalid. 2. Step 4. is repeated by the client. | | |
| Alternate/Exceptional Flows:  **E7b: booking cancel**   1. The booking is canceled. 2. Use case terminates. | | |

|  |  |  |
| --- | --- | --- |
| Use Case Name: View All Booking Details | ID: 2 | Importance Level : **High** |
| Primary Actor: **User** | Use Case Type: detail, essential | |
| Stakeholders and Interests:  User - User wants to view the booking details of the amusement park | | |
| Brief Description:  This use case describes how **User** uses the application to view booking details for the amusement park | | |
| Trigger:  User wants to update the booking details of the amusement park. | | |
| Relationships:           Association: User           Include: N/A           Extend: N/A           Generalization: N/A | | |
| Normal Flow of Events:   1. User logins to the amusement park management system.    1. ***If*** the User is **Client**   The S1a: **Client** view booking details is performed.   1. ***If*** the User is **Admin**   The S1b: **Admin** view booking details is performed.   1. User ***enters*** the book id they want to choose. 2. ***If***  the ID is invalid   The E1a: Display Error Message is performed   1. ***If*** the ID is valid Details of the selected booking record including book date, ticket date, total price and items booked is displayed 2. ***If*** update booking is selected   Use case 3 **Update Booking Details** is performed   1. ***If*** cancel booking is selected   Use case 4 **Cancel Booking** is performed   1. ***If*** back is selected   Use case terminates. | | |
| SubFlows:  **S1a: Client view booking details**   1. The system displays all the booking records of the Client.   **S1b: Admin view booking details**   1. The system displays all the booking records in the system. | | |
| Alternate/Exceptional Flows:  **E1a: Display Error Message**   1. The system displays an error message. 2. **Step 2** is performed again. | | |

|  |  |  |
| --- | --- | --- |
| Use Case Name: Update Booking Details | ID: 3 | Importance Level : **High** |
| Primary Actor: **User** | Use Case Type: detail, essential | |
| Stakeholders and Interests:  User - User wants to update the booking details of the amusement park online | | |
| Brief Description:  This use case describes how **User** uses the application to update booking details for the amusement park | | |
| Trigger:  User wants to update the booking details of the amusement park. | | |
| Relationships:           Association: User           Include: N/A           Extend: N/A           Generalization: N/A | | |
| Normal Flow of Events:   1. User ***enters*** the date they want to change to.    1. If the date is not invalid   The S1a: retry date entry step is performed.   1. The system updates the booking list and writes to the file. | | |
| SubFlows: N/A  **S1a: retry date entry**   1. The system shows that the date is invalid. 2. Step 1. is repeated by the client. | | |
| Alternate/Exceptional Flows: N/A | | |

|  |  |  |
| --- | --- | --- |
| Use Case Name: Cancel Booking | ID: 4 | Importance Level : **High** |
| Primary Actor: **User** | Use Case Type: detail, essential | |
| Stakeholders and Interests:  User - User wants to cancel booking for the amusement park online | | |
| Brief Description:  This use case describes how **User** uses the application to cancel booking for the amusement park. | | |
| Trigger:  User wants to cancel a reserved booking in the amusement park system. | | |
| Relationships:           Association: User           Include: N/A           Extend: N/A           Generalization: N/A | | |
| Normal Flow of Events:   1. System asks for confirmation of the cancellation of the selected item.  1. ***If*** the user ***confirms*** the cancellation The system will update the booking list and write to the file. 2. ***If*** the user ***denies*** the cancellation   Use case terminates | | |
| SubFlows:  N/A | | |
| Alternate/Exceptional Flows: N/A | | |

|  |  |  |
| --- | --- | --- |
| Use Case Name: Search Booking Details | ID: 5 | Importance Level : **High** |
| Primary Actor: **User** | Use Case Type: detail, essential | |
| Stakeholders and Interests:  User - User want to search the booking details of the amusement park | | |
| Brief Description:  This use case describes how the user uses the application to search the booking details of the Amusement Park Management System. | | |
| Trigger:  The user wants to search for the booking details of the Amusement Park Management System. | | |
| Relationships:           Association: User           Include: N/A           Extend:  N/A           Generalization:  N/A | | |
| Normal Flow of Events:   1. User select search booking details option 2. ***If*** the User is **Client**,   The S1a: **Client** search booking details are performed.   1. ***If*** the User is **Admin,**   The S1b: **Admin** search booking details is performed. | | |
| SubFlows:  **S1a: Client search booking details**   1. Client wants to enter the search input    1. ***If***  Client wants to search by book ID   **SubFlow S1c: User enters book ID** is performed.   1. ***If***  client wants to search by date   **SubFlow S1d: User enters date** is performed.   1. ***If***  user enters 3   Use case terminates.  **S1b: Admin search booking details**   1. Admin wants to enter the search input    1. ***If*** Admin wants to search by book ID   **SubFlow S1c: User enters book ID** is performed   1. ***If***  admin wants to search by date   **SubFlow S1d: User enters date** is performed   1. ***If***  admin wants to search by user ID   **SubFlow S1e: User enters user ID** is performed   1. ***If***  user enters 0   Use case terminates.  **S1c: User enters book ID**   1. User enters book ID as search input. 2. ***If*** the book ID is invalid    1. E1: Display Error Message will be performed    2. **SubFlow S1c Step 1** is performed 3. ***If*** the book ID is valid    1. System display the booking record of that book ID    2. User is prompted to select further action    3. ***If***  update is selected   Use case 3 **Update Booking Details** is performed.   1. ***If***  cancel is selected   Use case 4 **Cancel Booking** is performed.   1. ***If***  back is selected   **Step 1** is performed.  **S1d: User enters date**   1. User enters date as search input 2. ***If*** the date is invalid 3. E1: Display Error Message will be performed. 4. **SubFlow S1d Step 1** is performed. 5. ***If*** the date is valid    1. System displays the bookings of that date.    2. User is prompted to select further action.    3. ***If*** user wants to enters Book ID,   **Subflow s1c** is performed.   1. ***If***  user enters 0   **Step 1** is performed.  **S1e: Admin enters user ID**   1. User enters user ID as search input 2. ***If*** the user ID is invalid 3. E1: Display Error Message will be performed 4. **SubFlow S1e Step 1** is performed 5. ***If*** the user ID is valid 6. System display the bookings of that user ID 7. User is prompted to select further action 8. ***If*** user wants to enters Book ID   **Subflow S1c** is performed   1. ***If***  user enters 0   **Step 1** is performed. | | |
| Alternate/Exceptional Flows:  **E1: Display Error Message**   1. Display error message | | |

|  |  |  |
| --- | --- | --- |
| Use Case Name: Log in | ID: 6 | Importance Level : **High** |
| Primary Actor: **User** | Use Case Type: detail, essential | |
| Stakeholders and Interests:  User - User wants to login to the account to access the Amusement Park Management System. | | |
| Brief Description:  This use case describes how **users** login to access the system. | | |
| Trigger:  Users want to access the Amusement Park Management System functions. | | |
| Relationships:           Association: User           Include: N/A           Extend: Create New User Profile           Generalization: N/A | | |
| Normal Flow of Events:   1. The system displays the login menu    1. ***If*** the User does not have an account   Use case 6.1 **Create New User Profile** is performed.   1. ***If*** the User have an existing account   The S1b: Login into the system is performed. | | |
| SubFlows:  S1b: Login into the system   1. User login into his account by using username and password. 2. The system checks the authentication of the account.    1. If the username or password is invalid   The E1bii: Fail to login is performed   1. ***If*** successful login   system displays the main menu | | |
| Alternate/Exceptional Flows:  **E1bii: Fail to login**   1. The system shows an error message to inform the user.  1. S1b-1 is performed again by the user. | | |



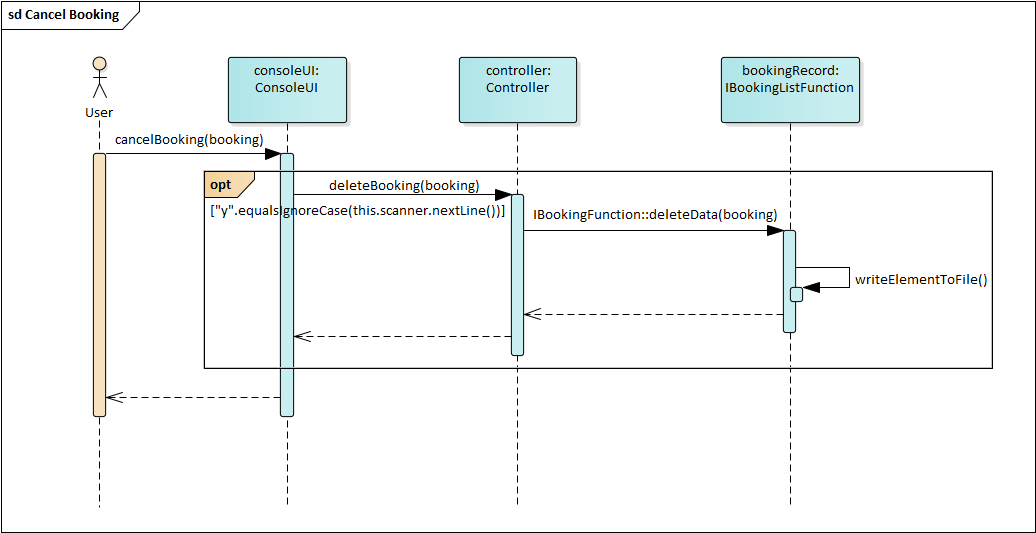
|  |  |  |
| --- | --- | --- |
| Use Case Name: Create New User Profile | ID: 6.1 | Importance Level : **High** |
| Primary Actor: **User** | Use Case Type: detail, essential | |
| Stakeholders and Interests:  User - User wants to sign up an account to access the Amusement Park Management System. | | |
| Brief Description:  This use case describes how **users** sign up to access the system. | | |
| Trigger:  Users want to create accounts for  the Amusement Park Management System functions. | | |
| Relationships:           Association: N/A           Include: N/A           Extend: Create N/A           Generalization: N/A | | |
| Normal Flow of Events:   1. User keys in personal details which include name, password and phone number.  1. The system checks if the telephone number is already belonged to a user account  1. If the telephone already belongs to an user account   The E3a: Fail to create account is performed   1. The system generates the account 2. The system shows the User ID of the user 3. User successfully login and system displays the main menu. | | |
| SubFlows: N/A | | |
| Alternate/Exceptional Flows:  **E3a: Fail to create account**   1. The system displays that the telephone entered already belongs to another account. 2. **Step 2** is performed again by the user. | | |

|  |  |  |
| --- | --- | --- |
| Use Case Name: Edit Booking Information | ID: 7 | Importance Level : **High** |
| Primary Actor: **Admin** | Use Case Type: detail, essential | |
| Stakeholders and Interests:  Admin wants to edit booking information, add new booking items and delete booking items so that items available for booking are updated. | | |
| Brief Description:  The use case describes how admin edit the information of the items available for booking. | | |
| Trigger:  Admin wants to update the system with the latest items that are available for booking. | | |
| Relationships:           Association: Admin           Include: N/A           Extend: N/A           Generalization: N/A | | |
| Normal Flow of Events:   1. Admin selects edit item information option. 2. System displays the list of items available for booking and provides options to choose to select an item or add a new item. 3. ***If*** admin select an item and choose to edit the information of the item   S2a edit existing booking step is performed.   1. ***If*** admin choose add new booking item   S2b adds a new booking item steps is performed.   1. ***If*** admin select an item and choose to delete the item   S2c delete a booking item is performed.   1. The system displays the edited list of items. | | |
| SubFlows:  **S2a: edit existing booking**   1. Admin selects the aspects including name,  price and description of the item to edit 2. Admin confirms the edit made on the item 3. The system updates the item details.   **S2b: add a new booking item**   1. Admin enters the name of the item and confirms it. 2. Admin enters the description of the item and confirms it. 3. Admin enters the price of the item and confirms it. 4. The system adds the new booking item into the list of items.   **S2c: delete a booking item**   1. System asks for confirmation of the deletion of the selected item. 2. ***If*** the user ***confirms*** the deletion The system will update the item list. 3. ***If*** the user ***denies*** the deletion   Use case terminates | | |
| Alternate/Exceptional Flows: N/A | | |

# Sequence Diagram

## Cancel Booking

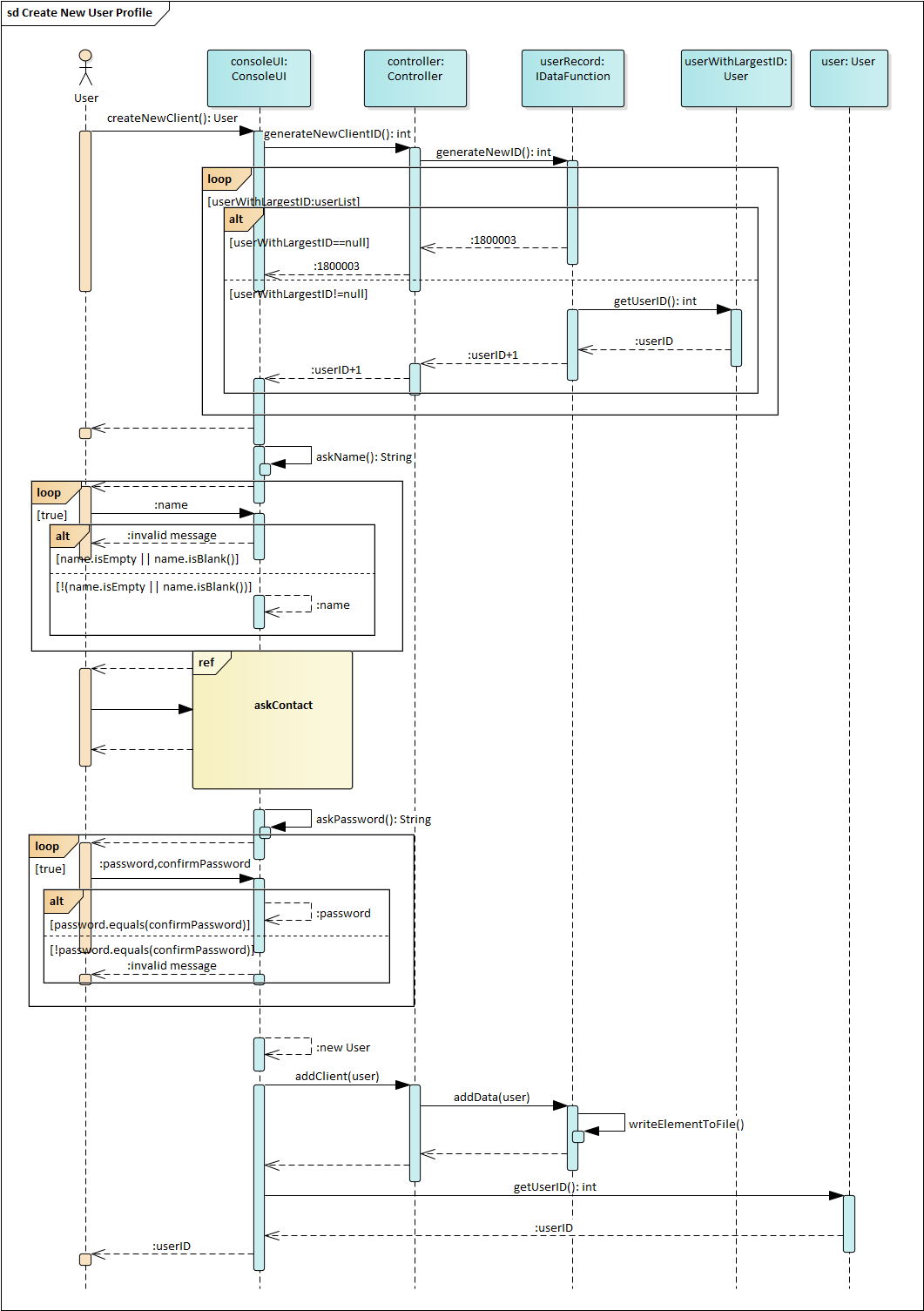




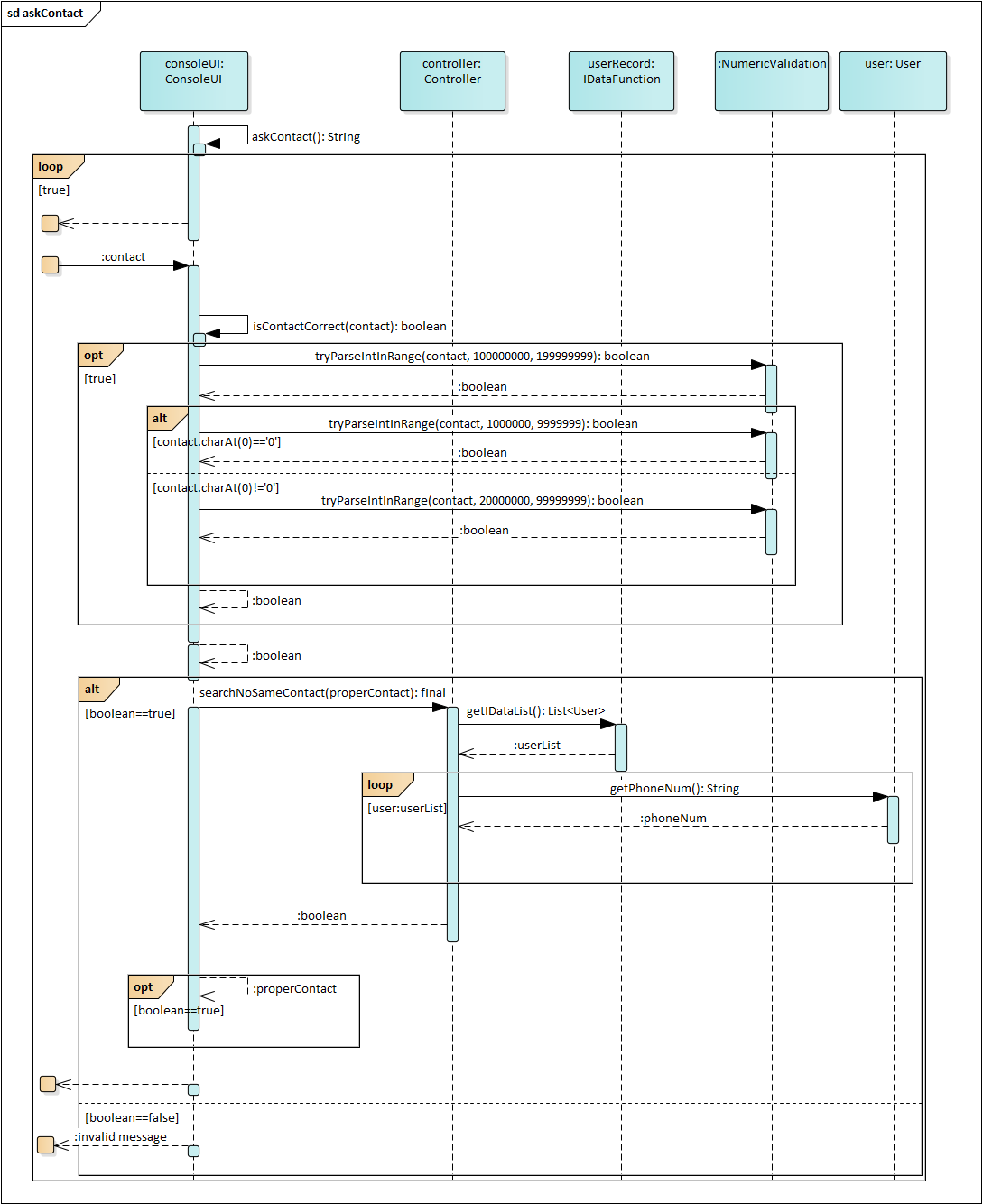


## Create New User Profile







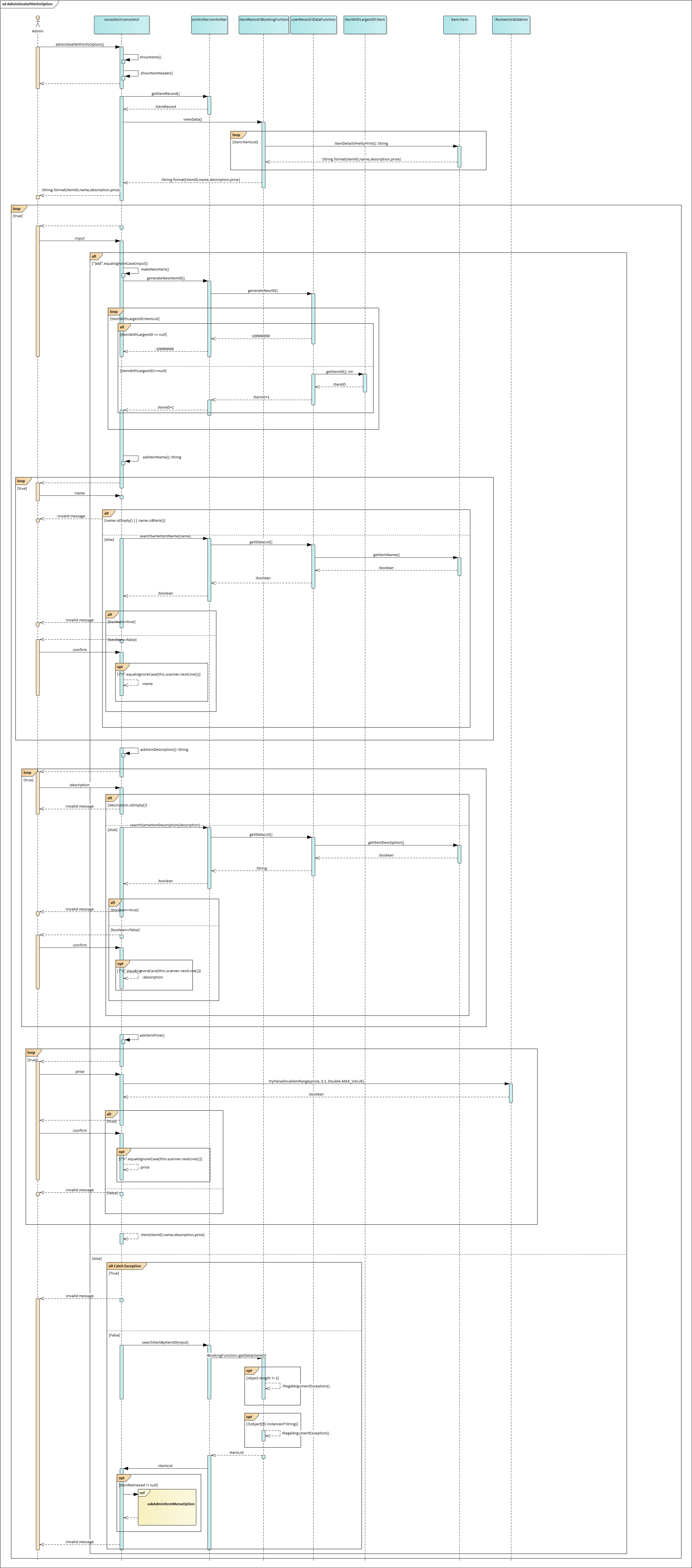




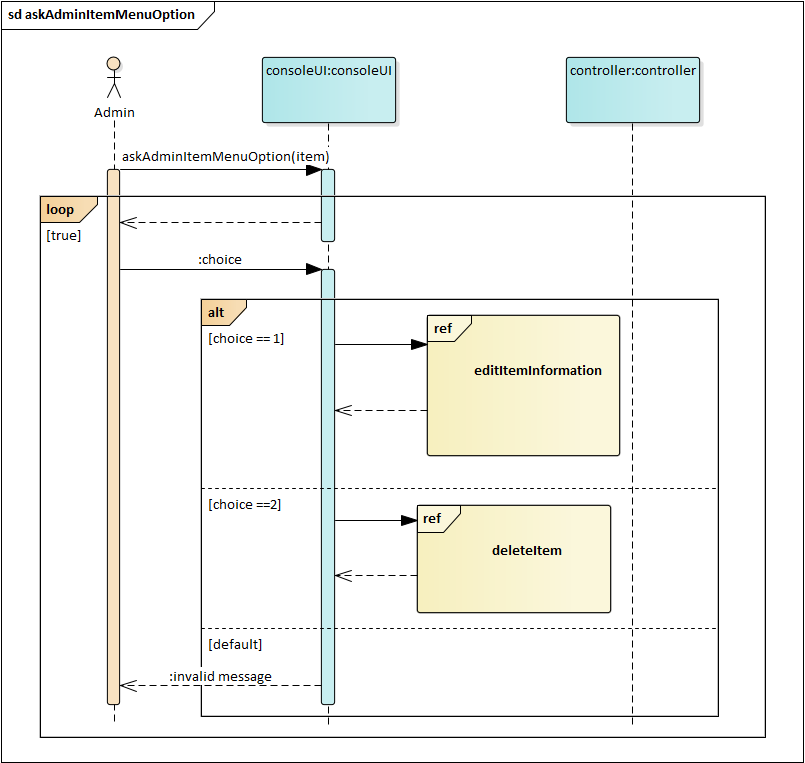
## Edit Booking Information



(can refer to ea, under use case diagram, in use case Edit Booking Information, sequence diagram name AdminDealwithInfoOption)



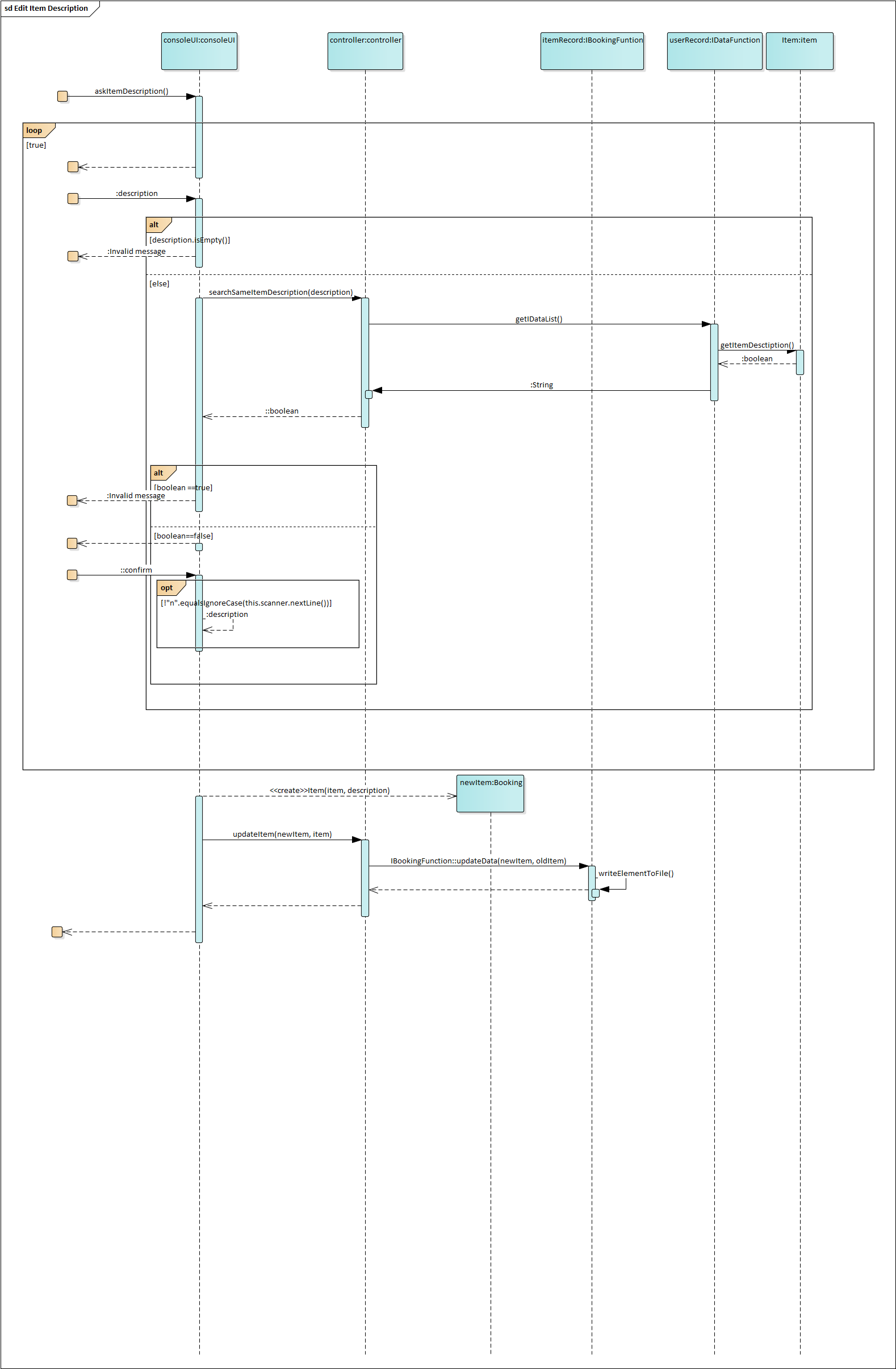


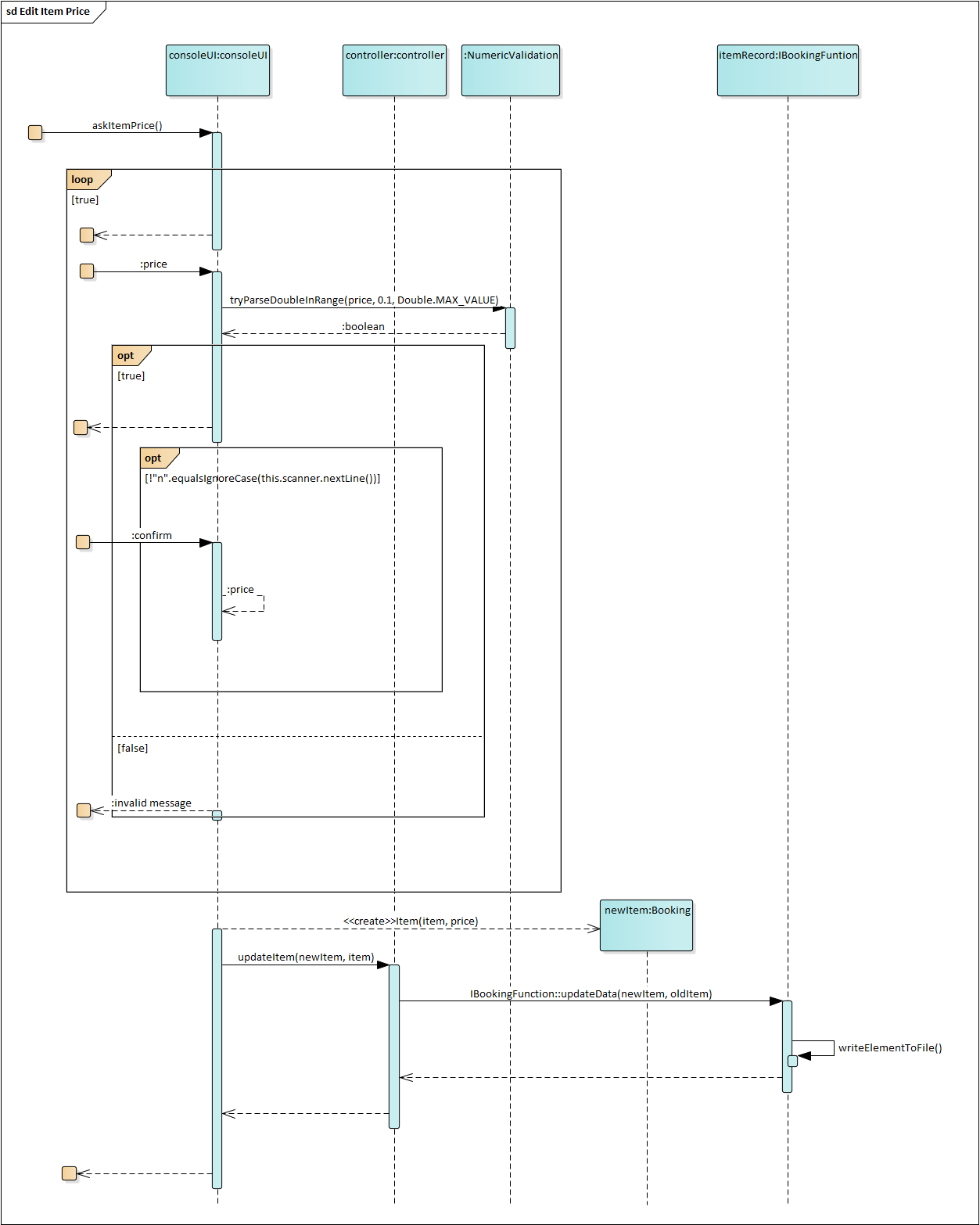


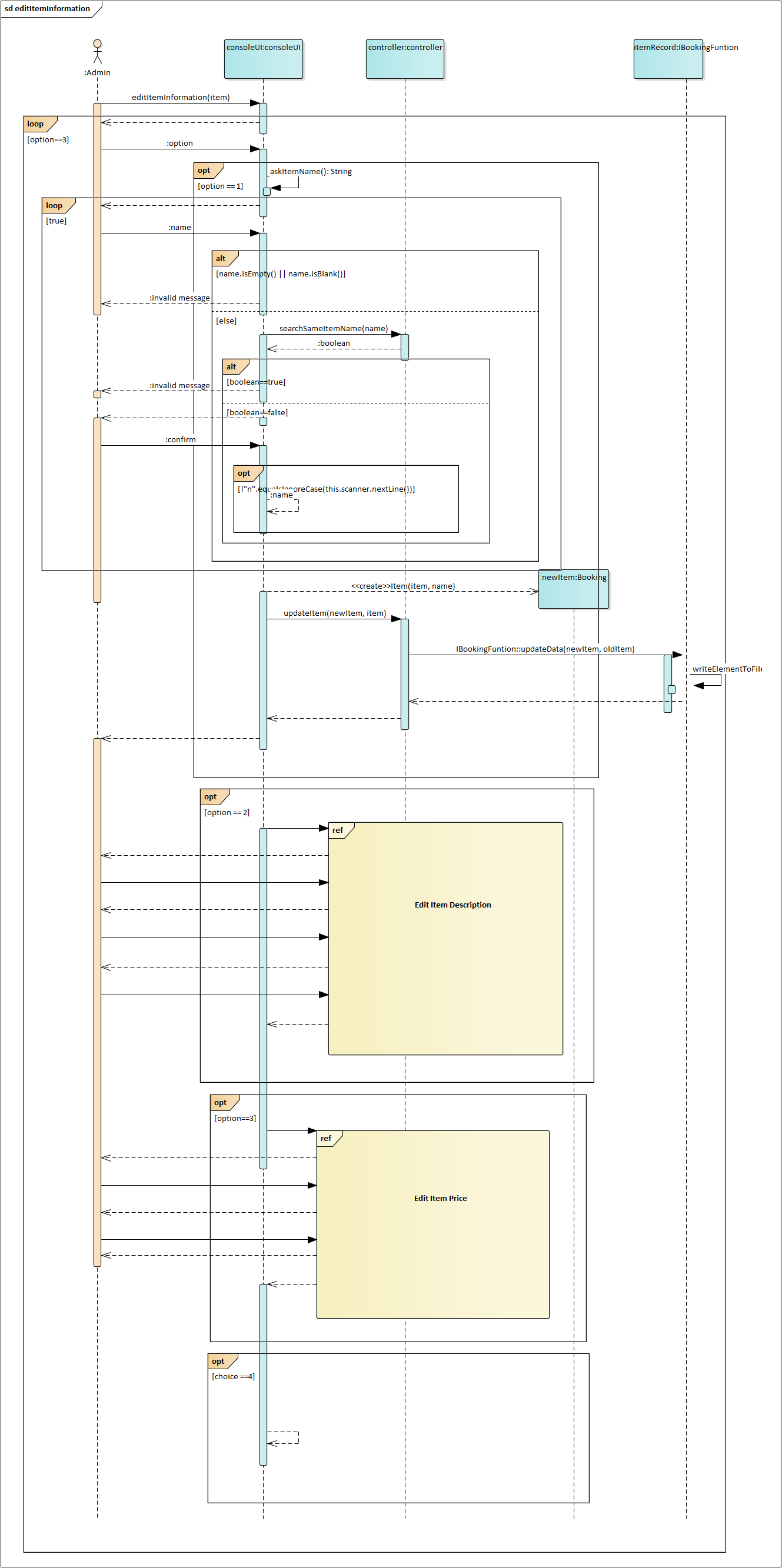






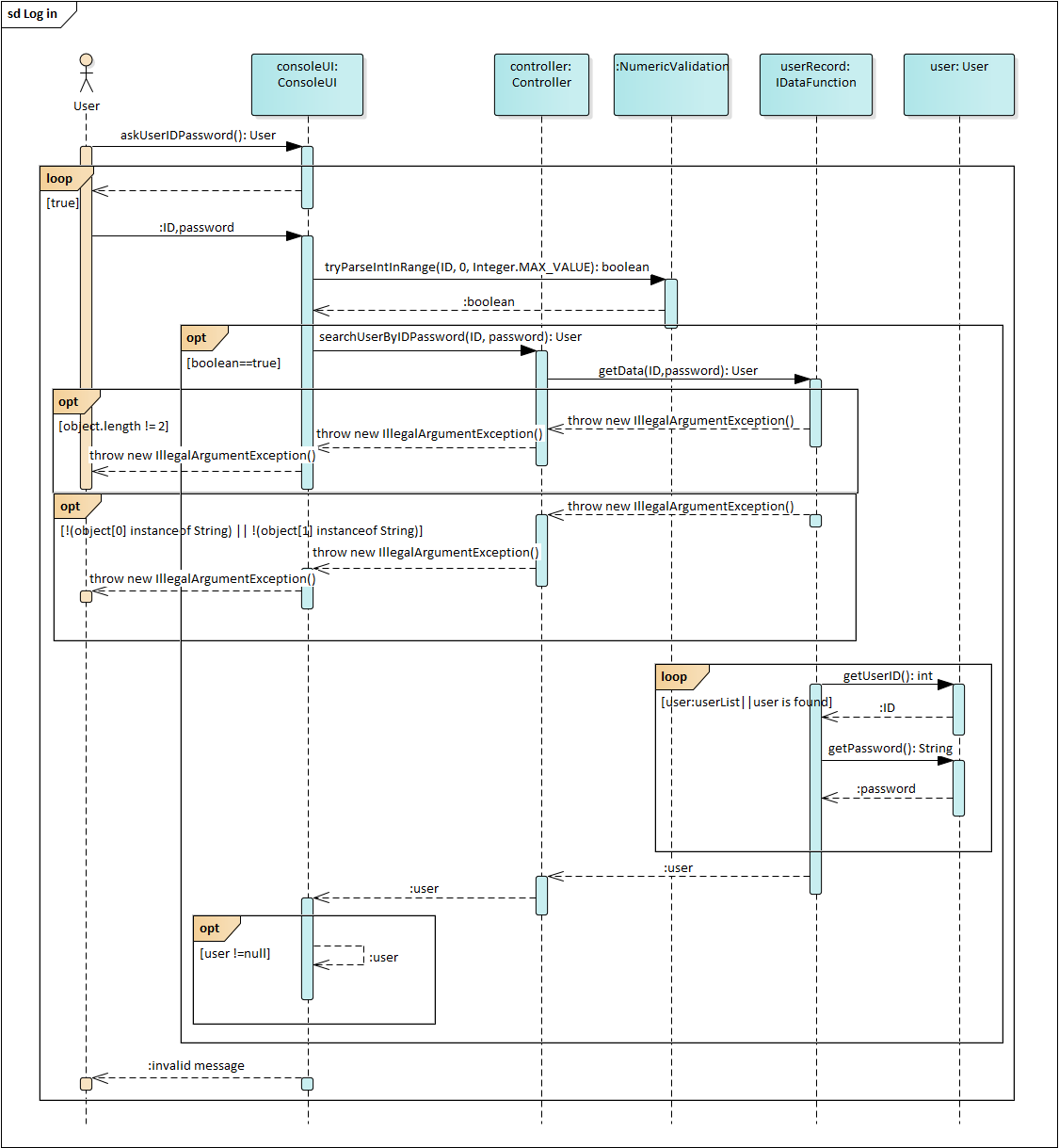






## Log In







## Make a Booking



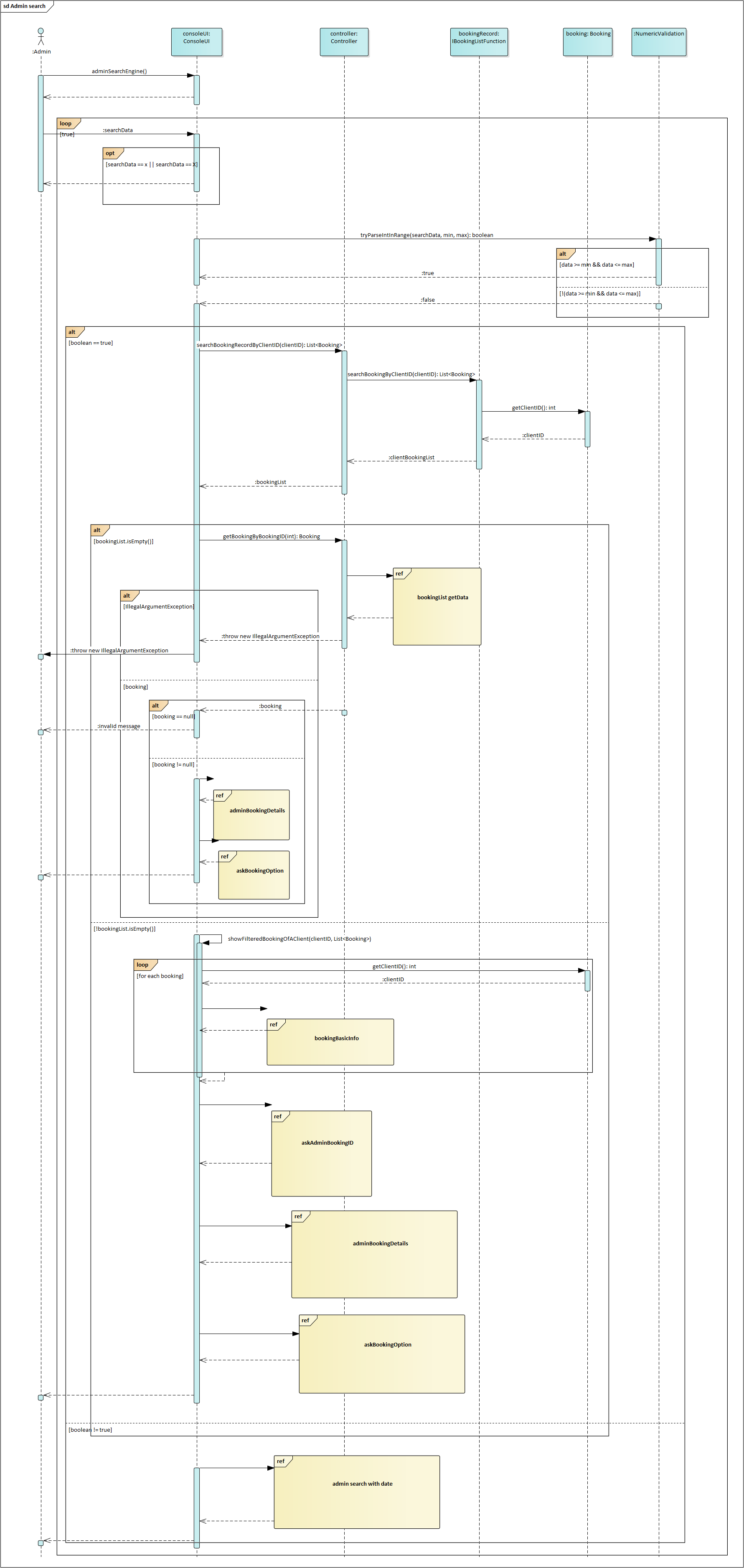
(can refer to ea, under use case diagram, in use case Make a Booking)



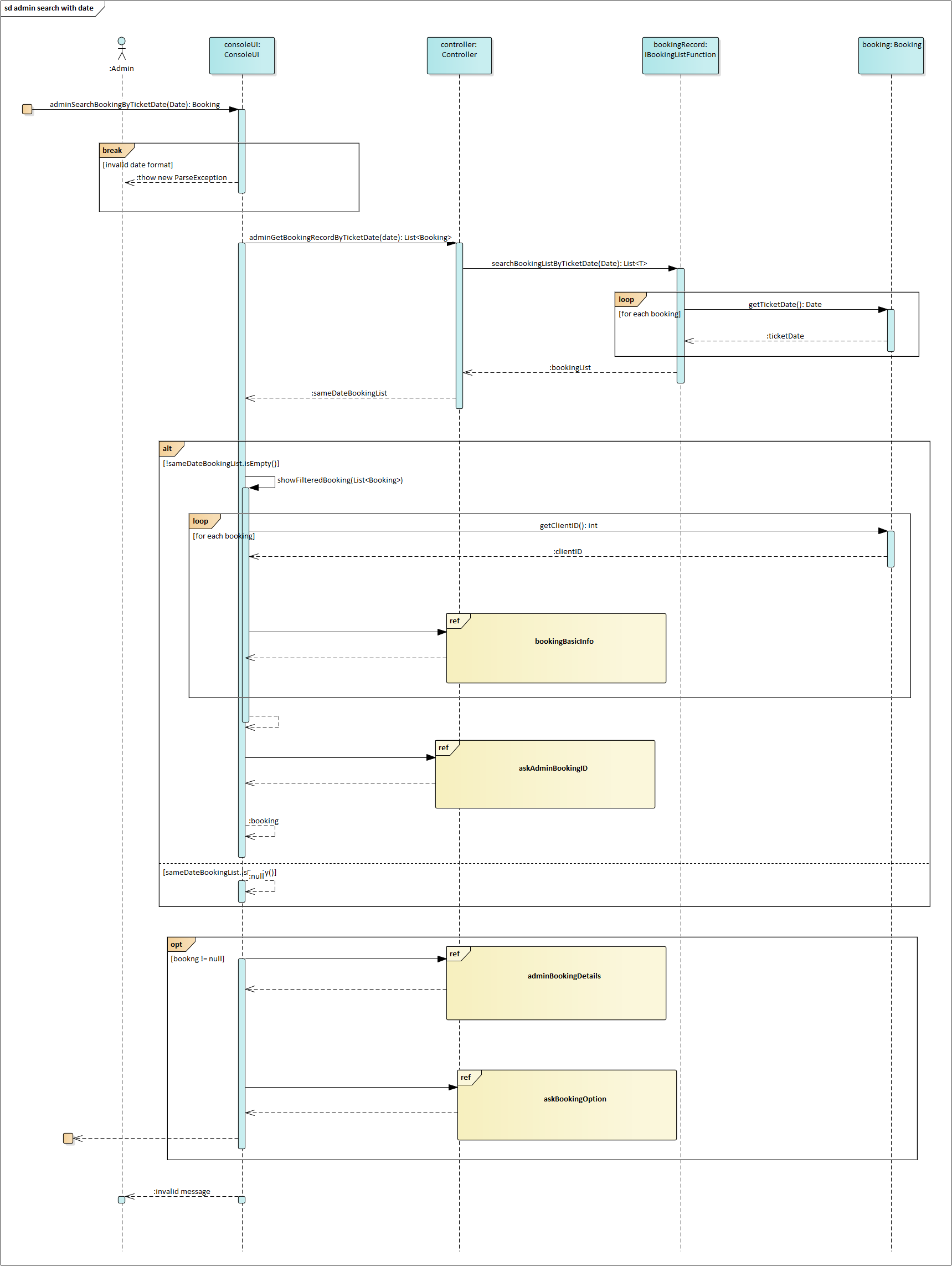
## Search Booking Details



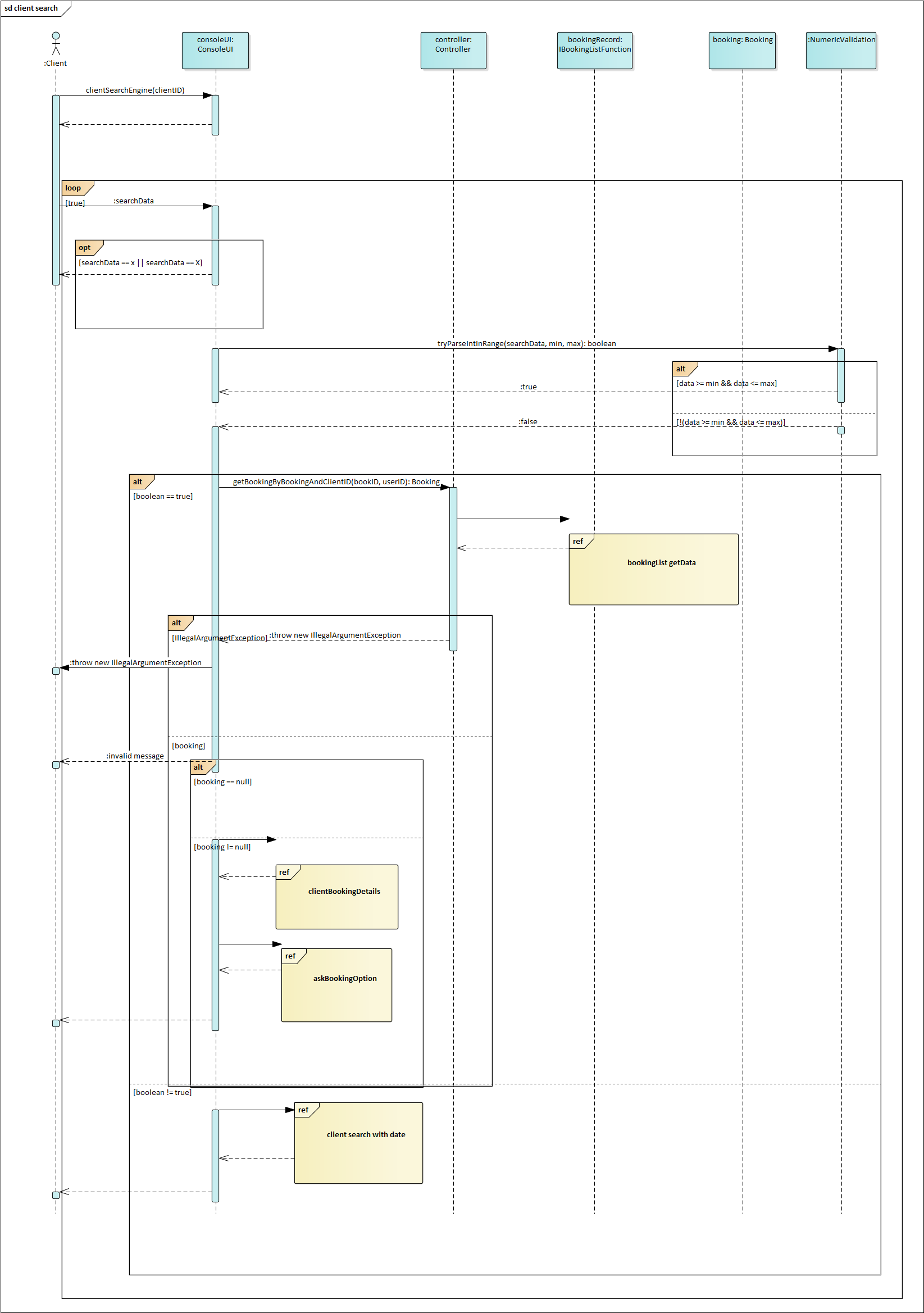
(can refer to ea, under use case diagram, in use case Search Booking Details, in Admin search fragment)



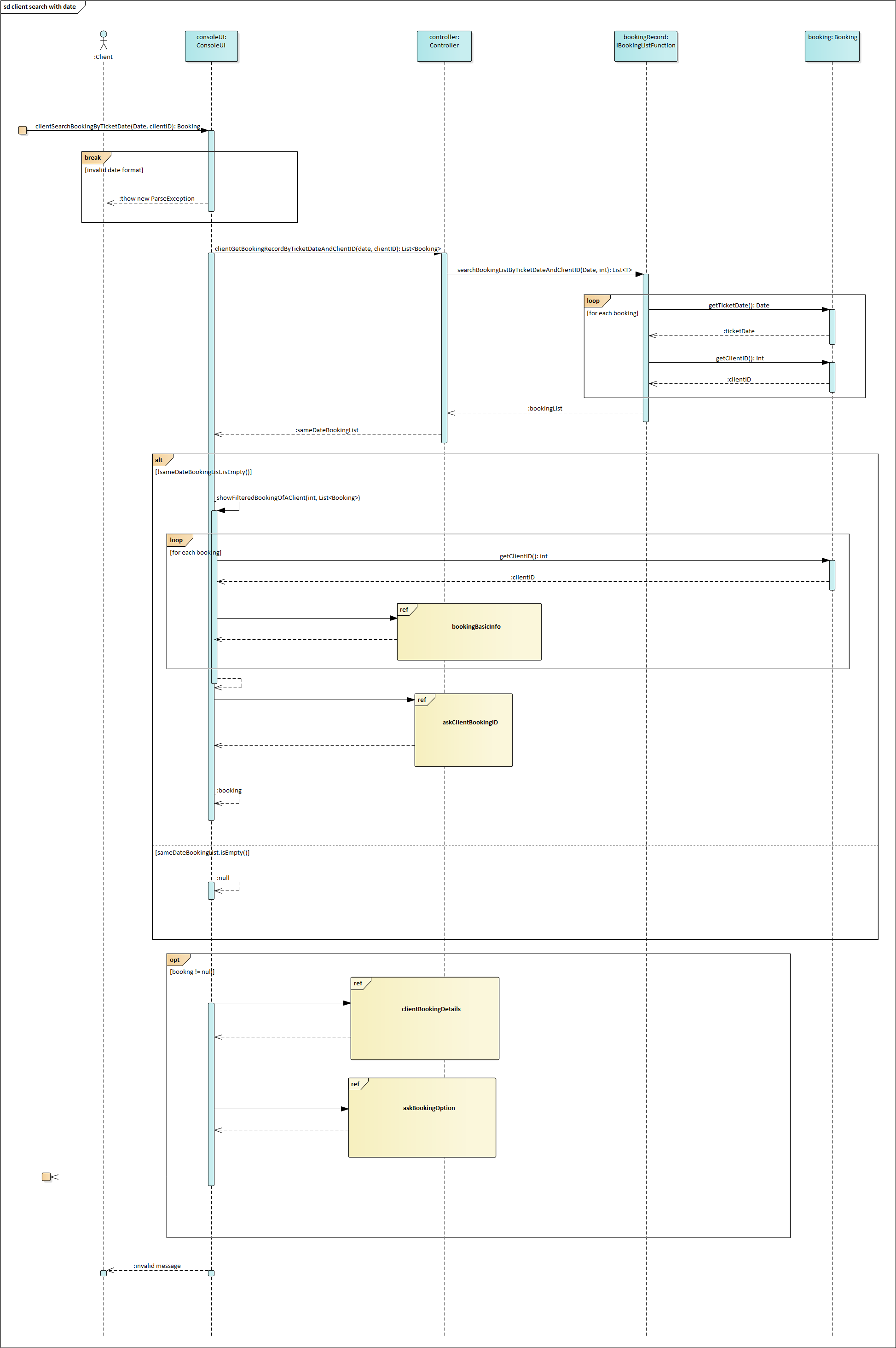








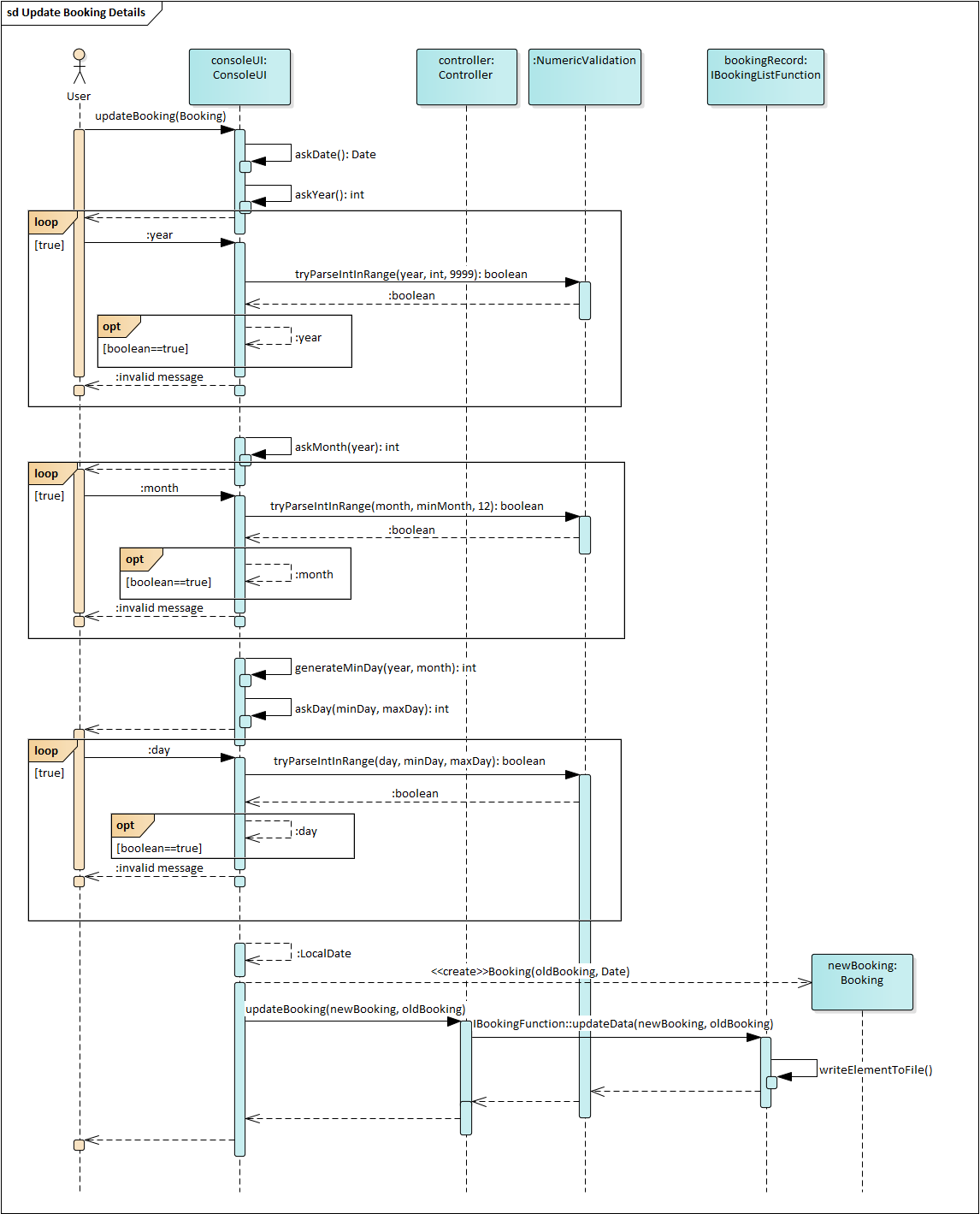






## Update Booking Details

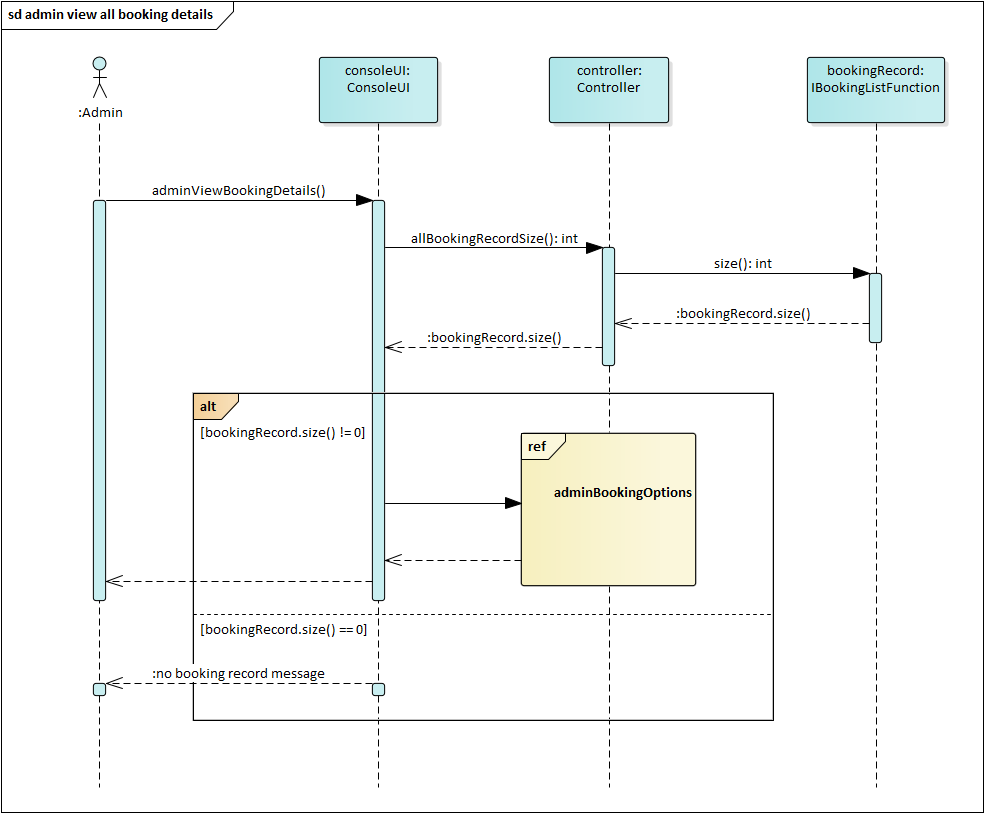




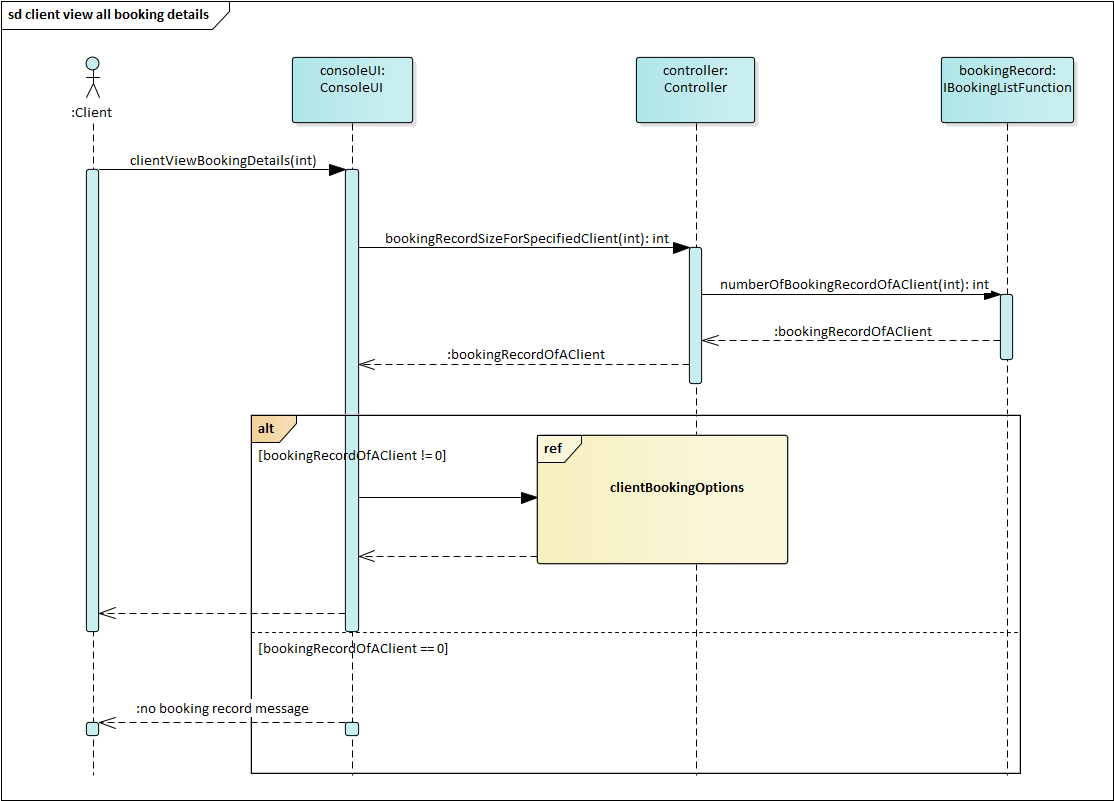


## View All Booking Details

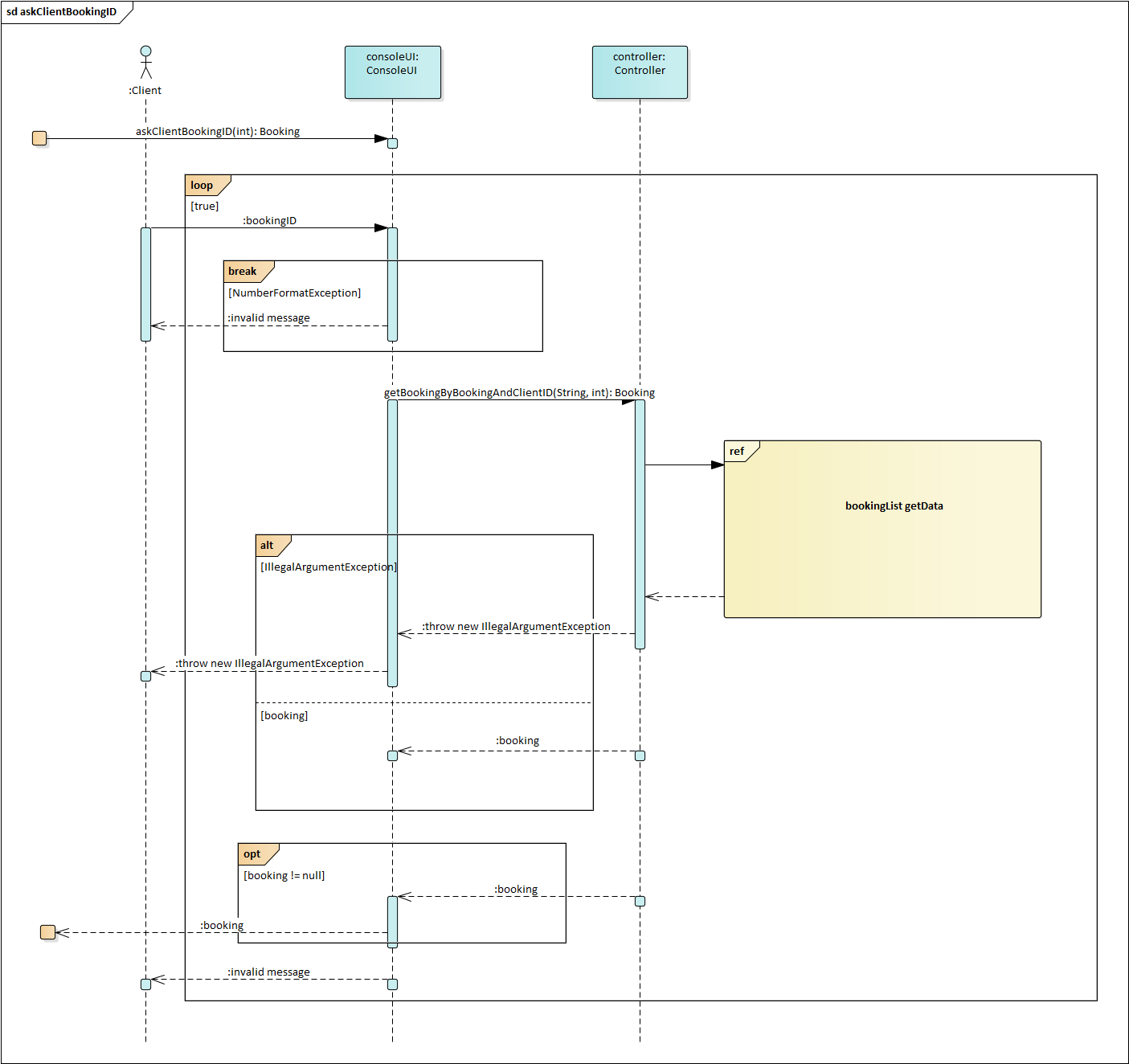




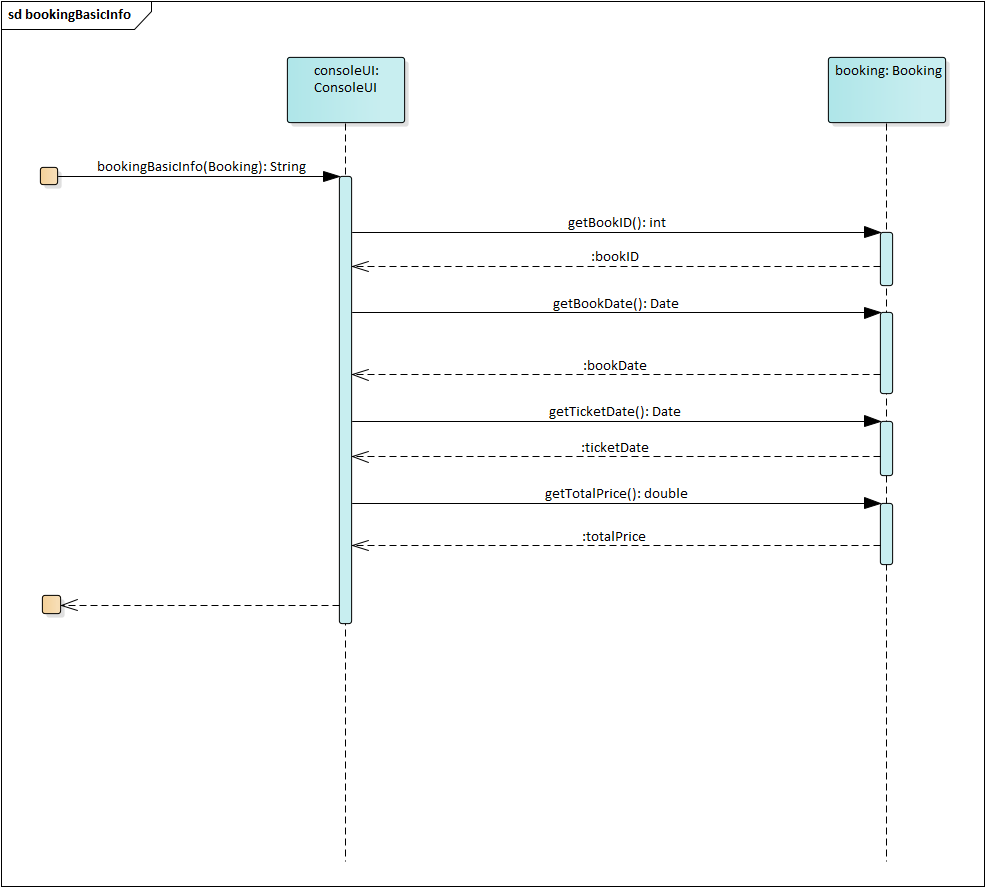








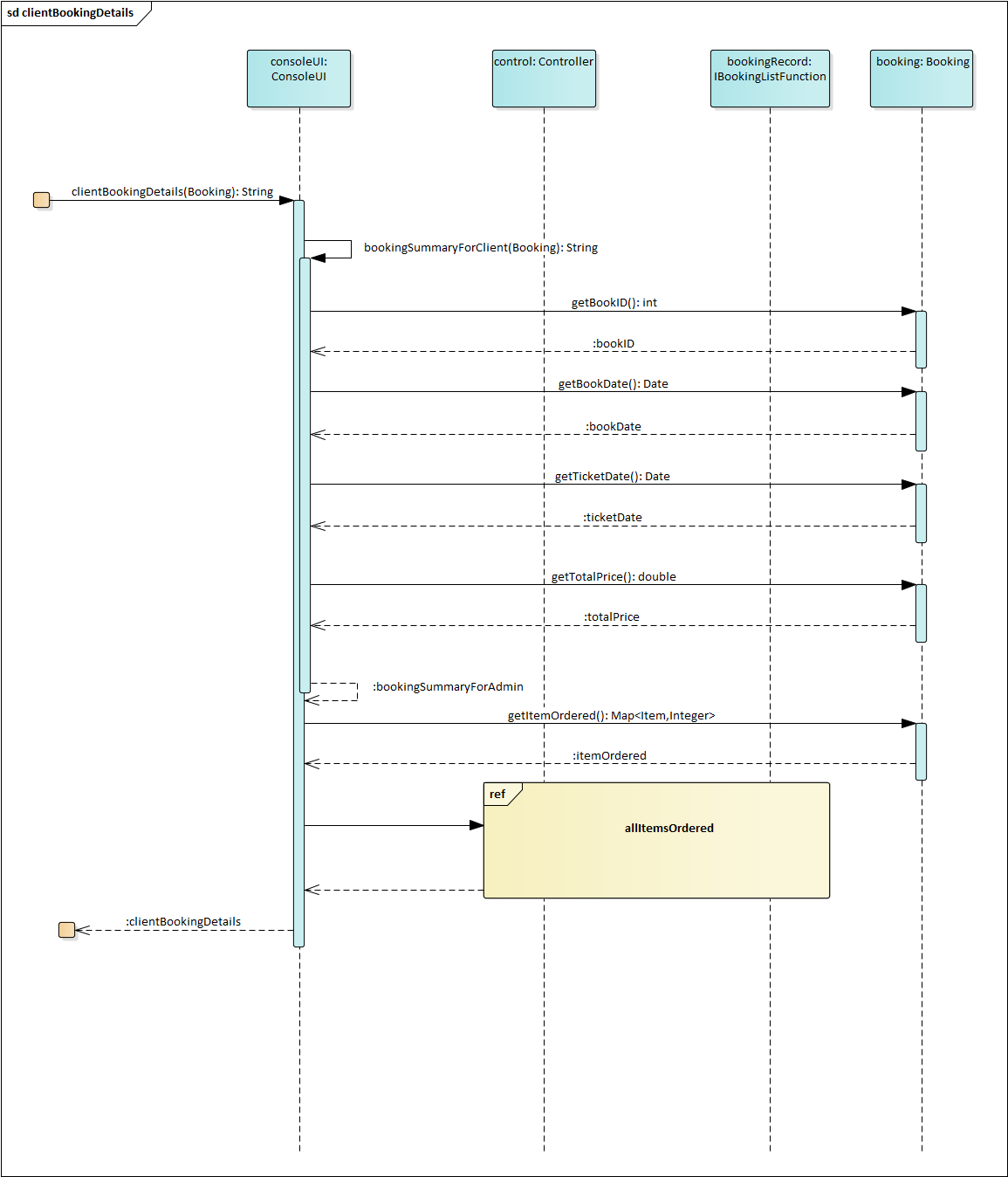




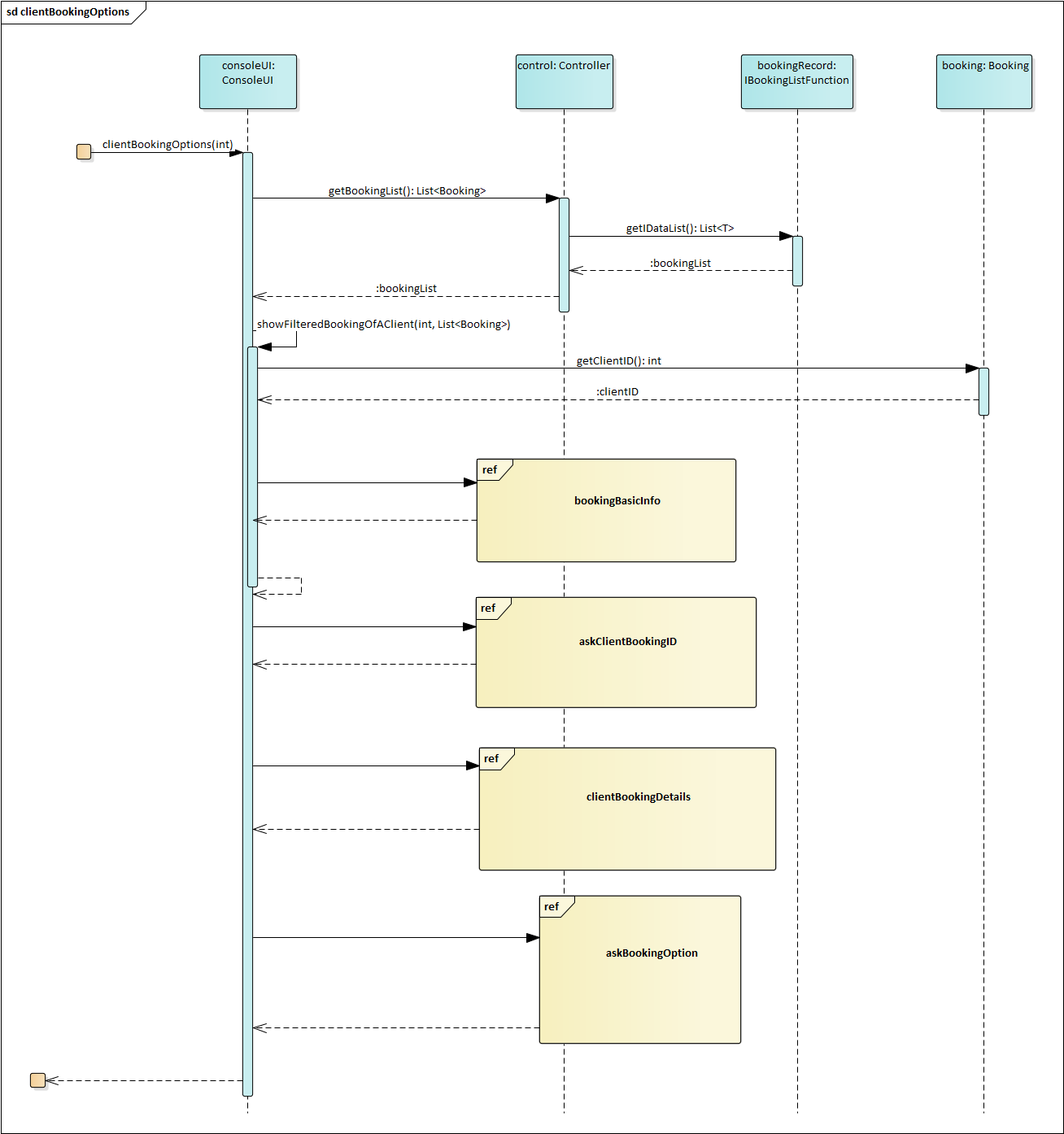




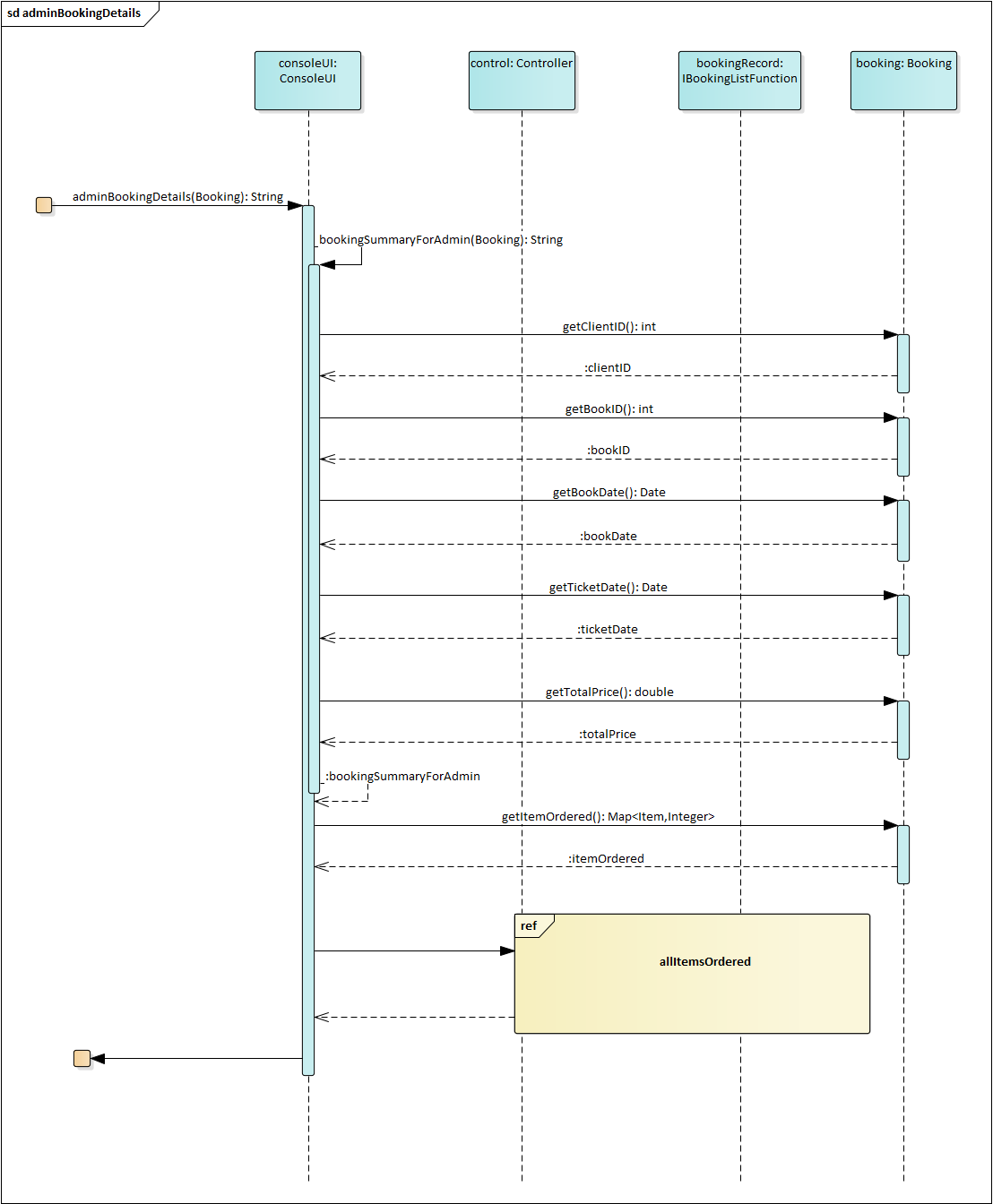




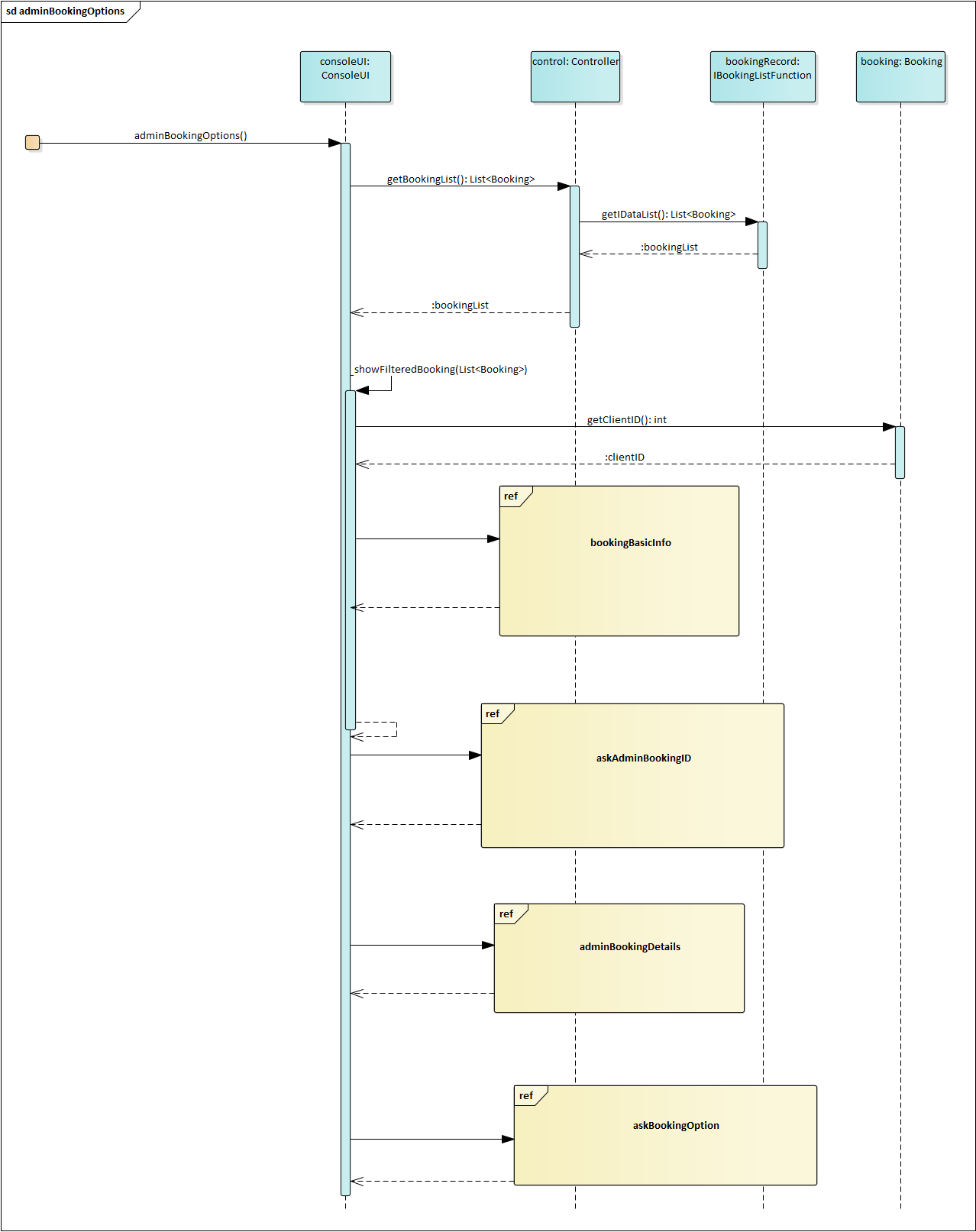




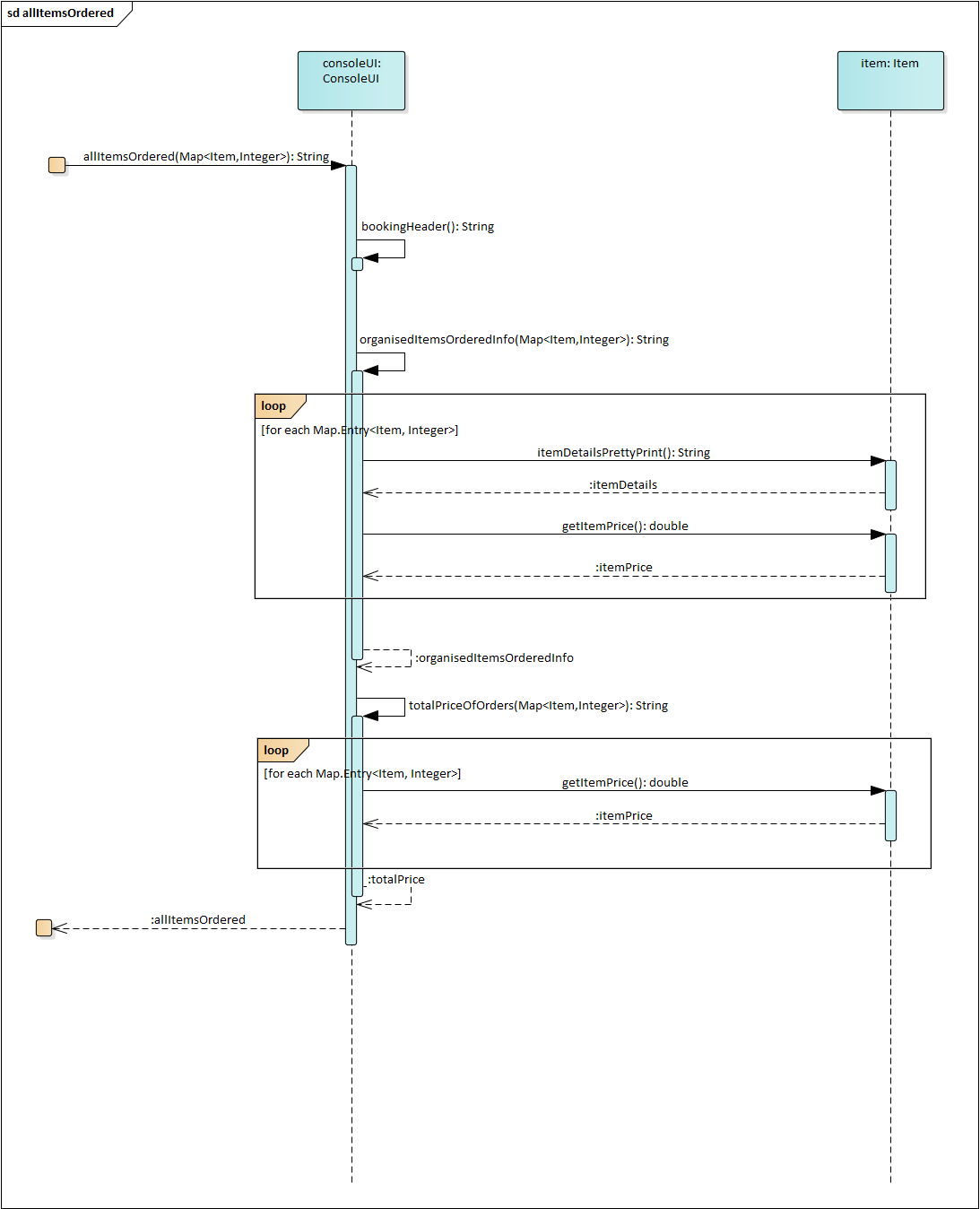




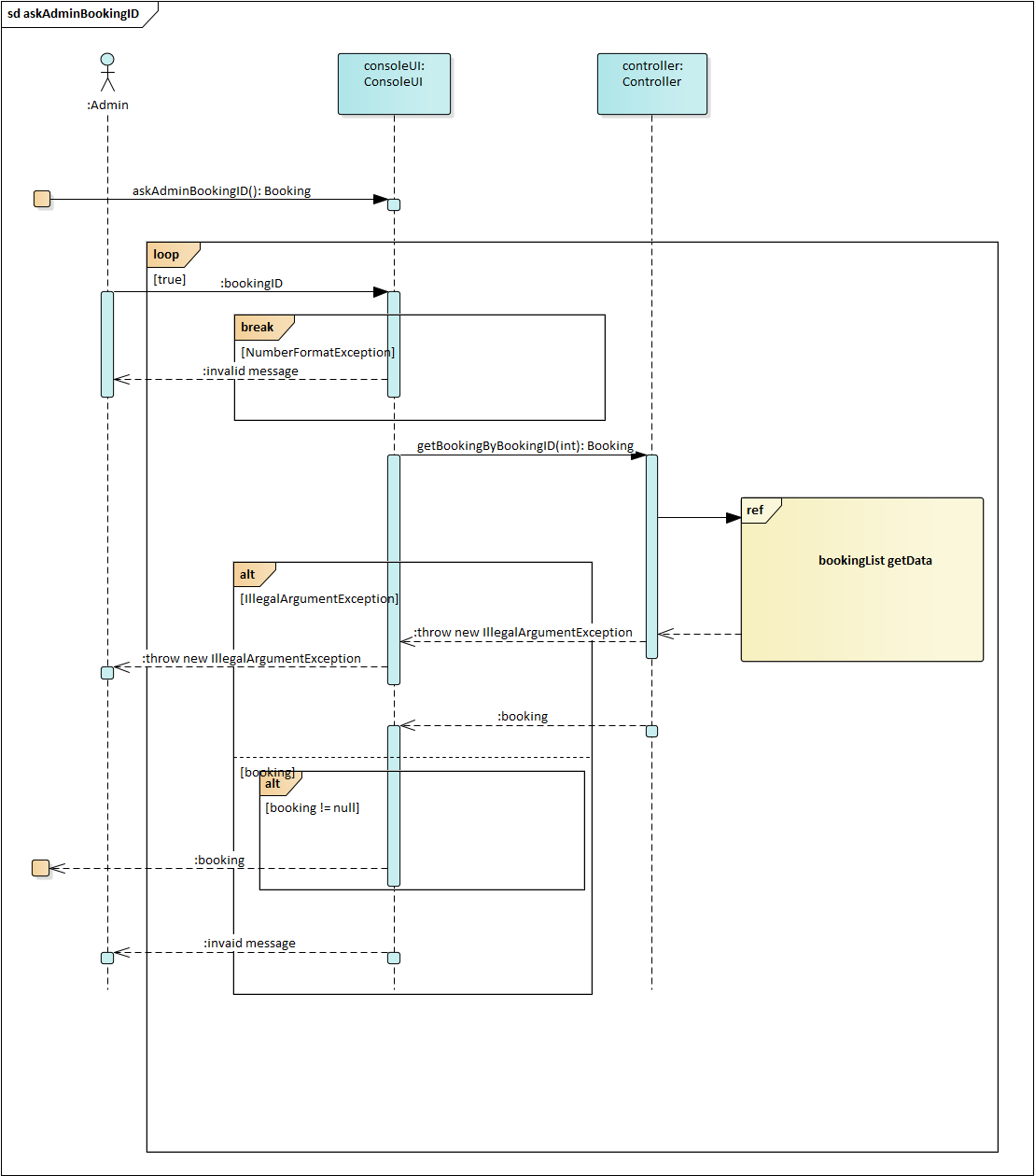




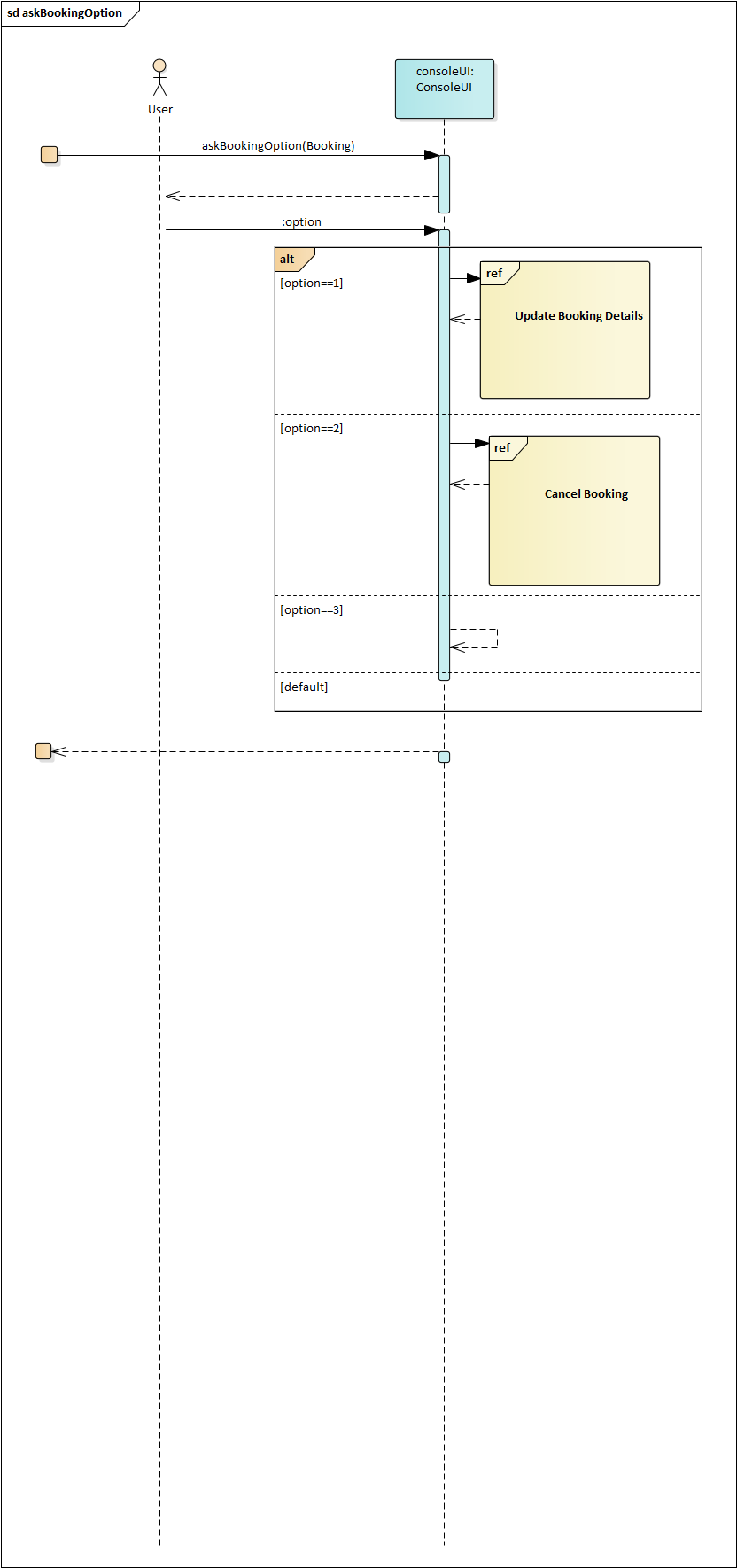








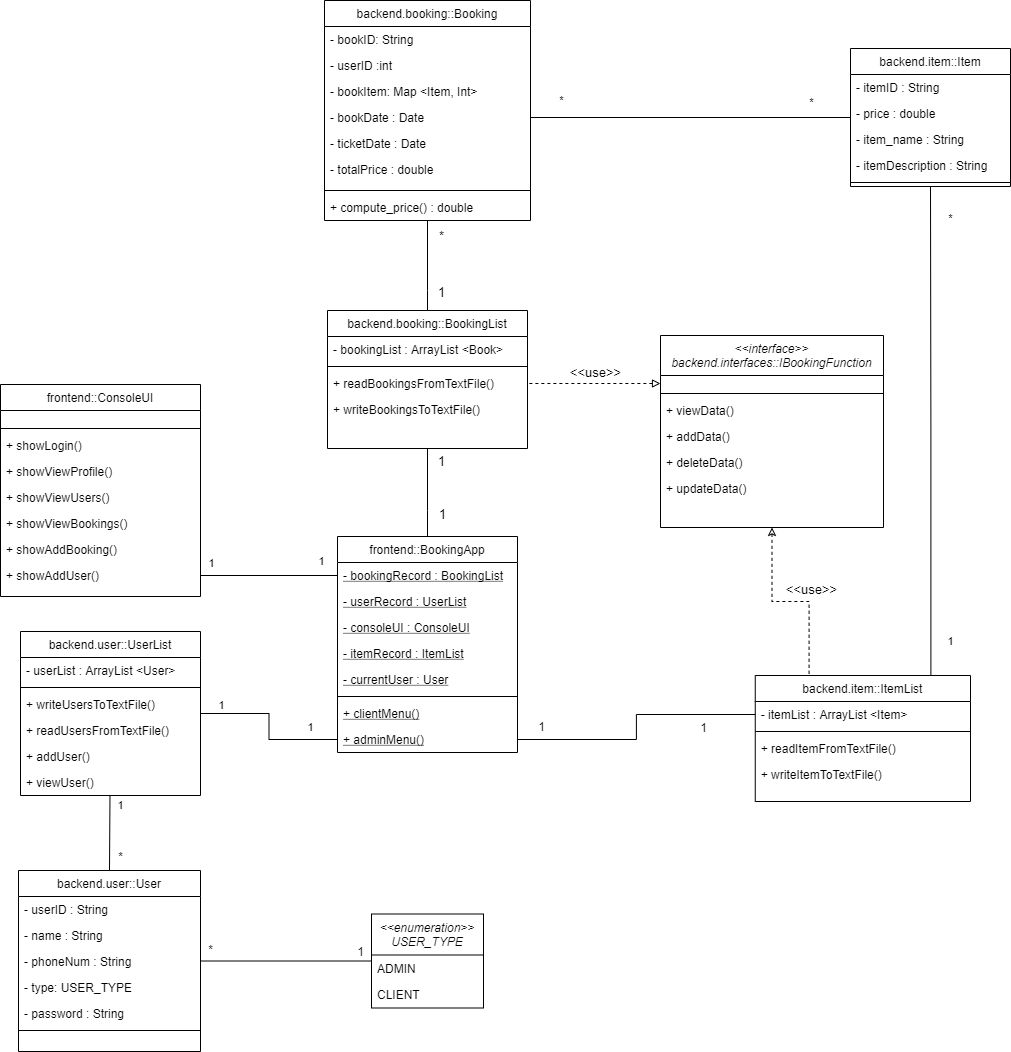






# Class Diagram

## Version 1



## Version 2

(can refer to ea, under Class Diagram)





# Package Diagram

