

LOW-LEVEL DESIGNS DOCUMENT

Hubba - SCHEDULING SYSTEM

Version 1.1

Prepared By: Development Hell

Class: CECS 491-04

Date: March 6, 2023

Github Repository:

<https://github.com/DevelopmentHellaHell/SeniorProject>

Team Leader

Kevin Dinh

Members

Garrett Tsumaki

Bryan Tran

Jett Sonoda

Tien Nguyen

Darius Koroni

POC: tien.nguyen10@student.csulb.edu

Revision History

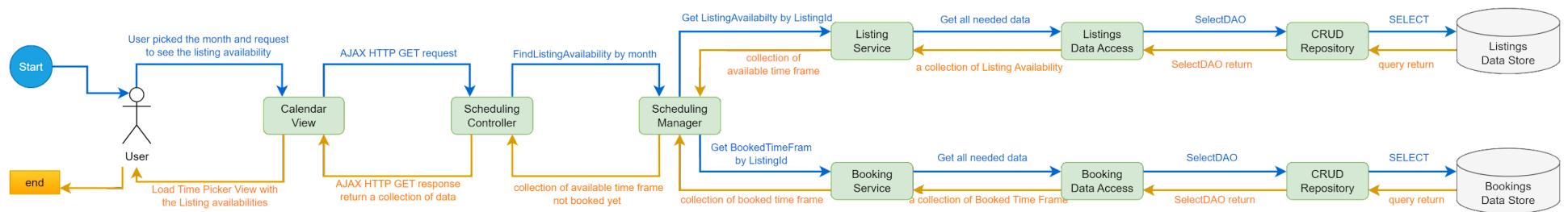
| Version | Overview | Date |
|---------|-----------------------------|-----------|
| v.1.0 | Initial LLD | 3/6/2023 |
| v.1.1 | Revise based on LLD Listing | 3/24/2023 |
| v.1.2 | Final draft | 3/31/2023 |

Table of Contents

| | |
|--|-----------|
| Table of Contents | 2 |
| High-Level Design | 3 |
| 1. Find Listing Availability | 3 |
| 2. Book a listing in a chosen time frame | 3 |
| 3. Cancel a booking | 3 |
| Relational Tables | 4 |
| Low-Level Design | 4 |
| 1. Find Listing Availability | 4 |
| Success Case | 4 |
| Failure Cases | 6 |
| Failed case 1 | 6 |
| Failed case 2 | 7 |
| 2. Booking a Listing in a time frame | 8 |
| Success Case | 8 |
| Failure Cases | 10 |
| Failed Case 1 | 10 |
| Failed Case 2 | 11 |
| Failed Case 3 | 12 |
| Failed Case 4 | 13 |
| 3. Cancel a Booking | 14 |
| Success Case | 14 |
| Failure Cases | 16 |
| Failed Case 1 | 16 |
| Failed Case 2 | 17 |
| Failed Case 3 | 18 |
| Failed Case 4 | 19 |
| References | 20 |

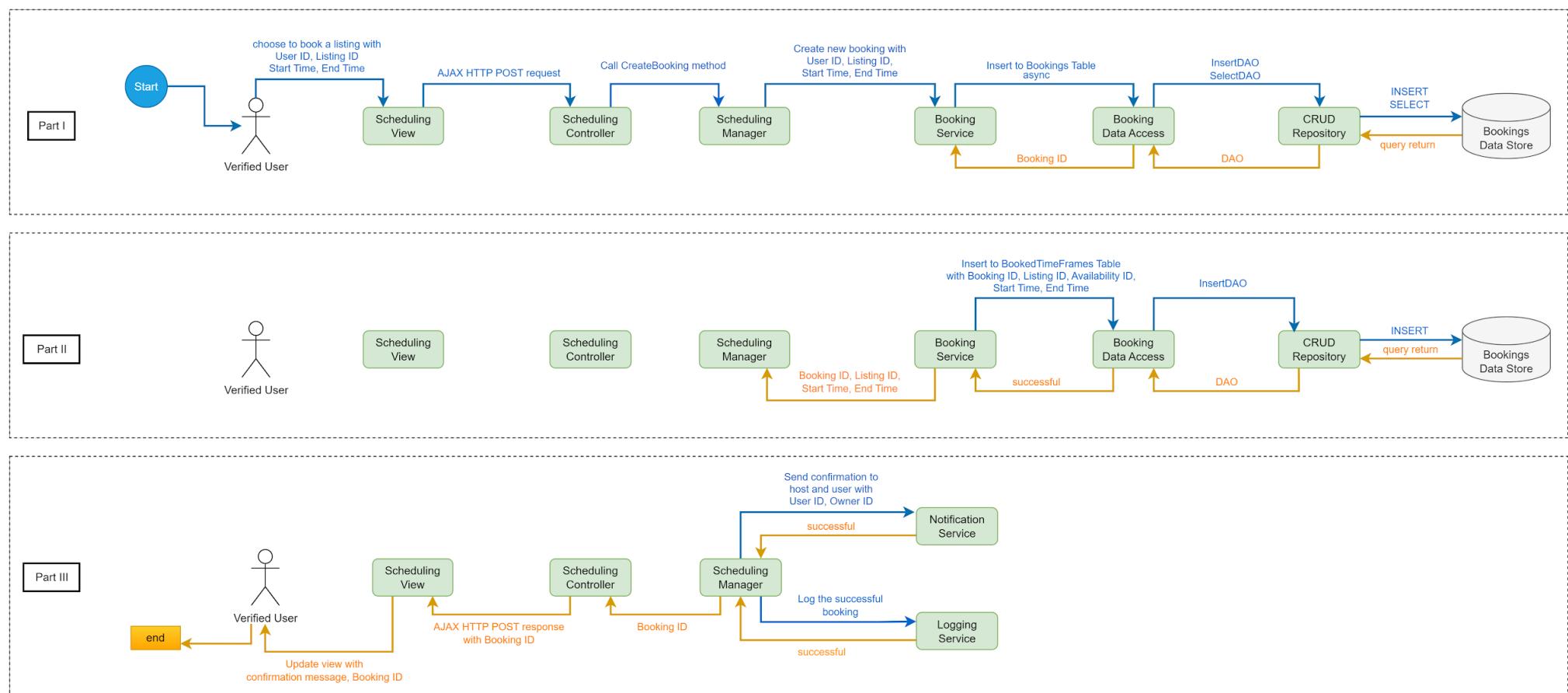
High-Level Design

1. Find Listing Availability

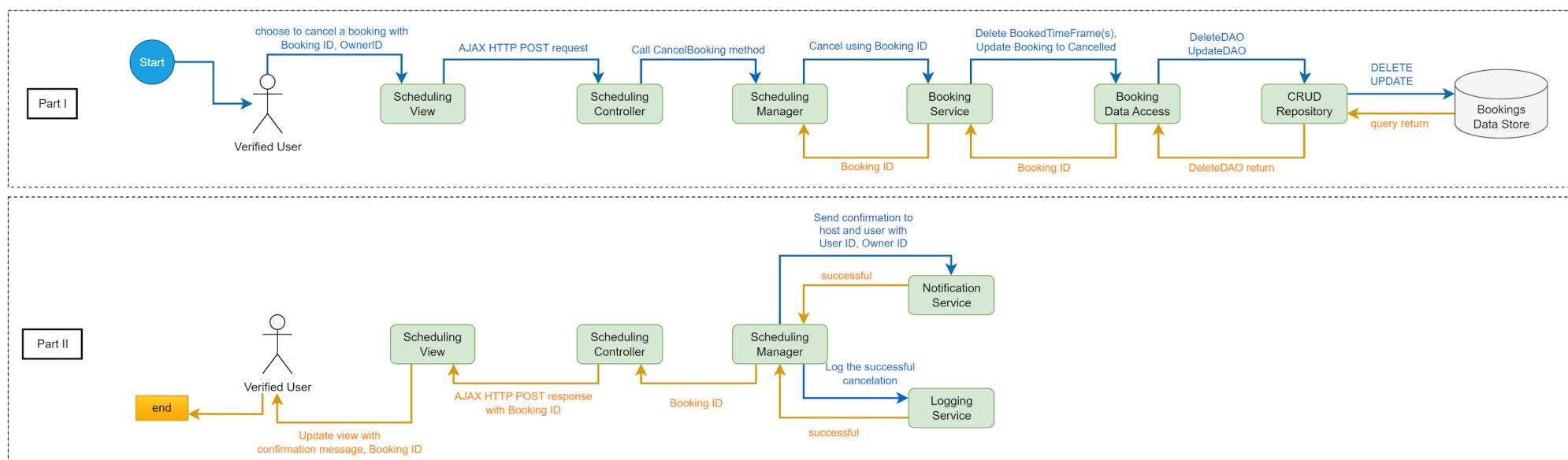


2. Book a listing in a chosen time frame

- Refer LLD Notification¹ for Notification Service
- Refer LLD Logging² for Logging Service



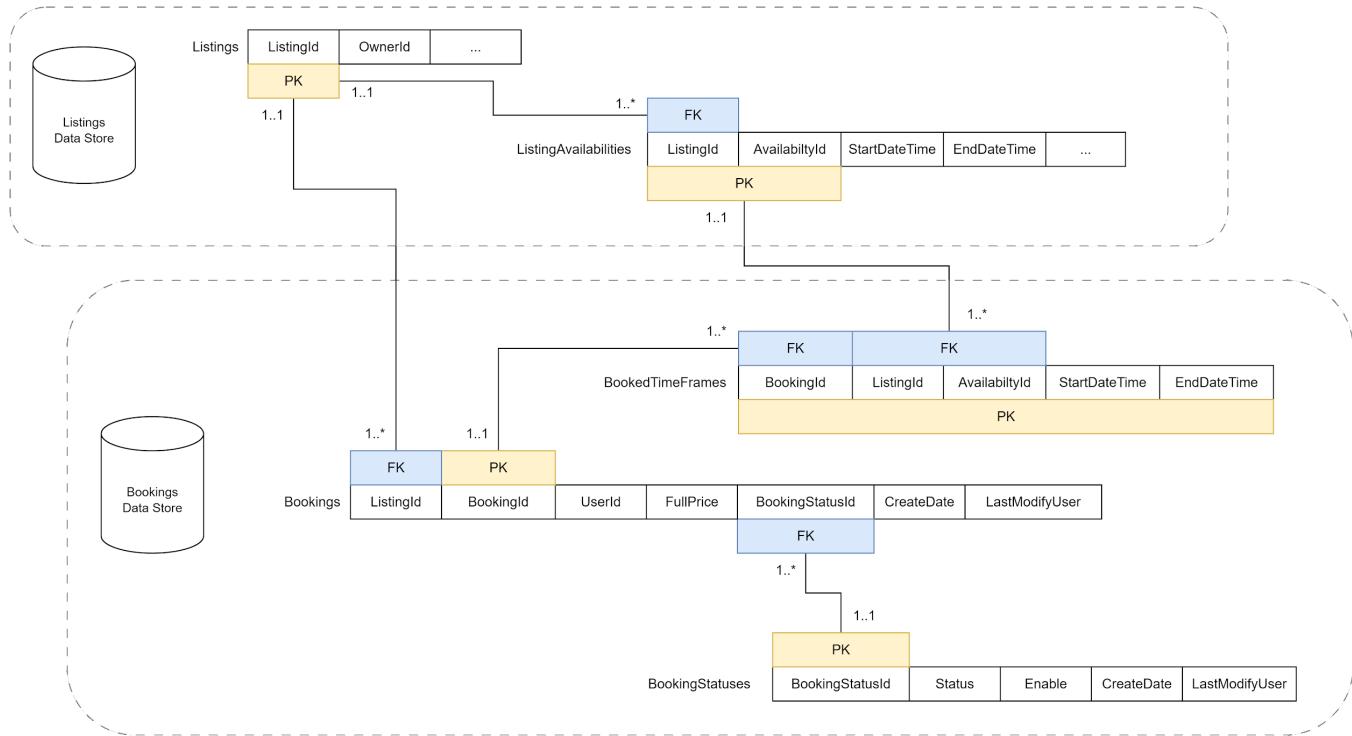
3. Cancel a booking



¹ LLD Notification URL: <https://github.com/DevelopmentHellaHell/SeniorProject/blob/main/docs/Reviews/Notification%20System/Notification%20Design.pdf>

² LLD Logging URL:
https://github.com/DevelopmentHellaHell/SeniorProject/blob/main/docs/LL%20Design/Backend/Common%20Components/Logging/LLDesign_Logging_v1.pdf

Relational Tables



Low-Level Design

1. Find Listing Availability

User story: As a user, I can request the listing availability, choose a time frame for that listing. System will display options of available time frames if any occurs.

Business rules:

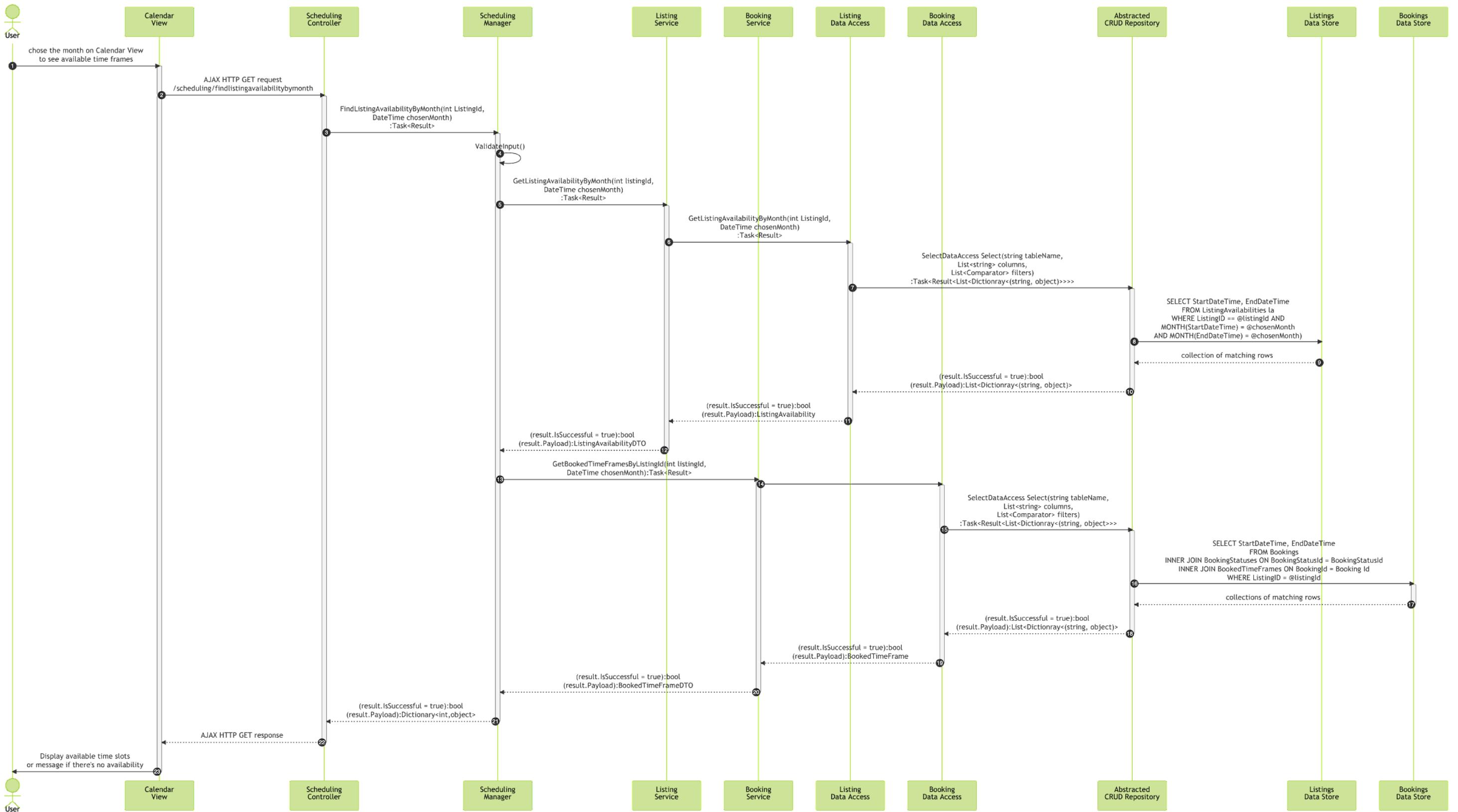
- Any user can request the listing availability.
- Available time slots must be displayed after the request within 5 seconds.
- System failure from this feature will not cause the system to go offline.

Preconditions:

- User has selected a listing
- User must be on calendar view

Success Case

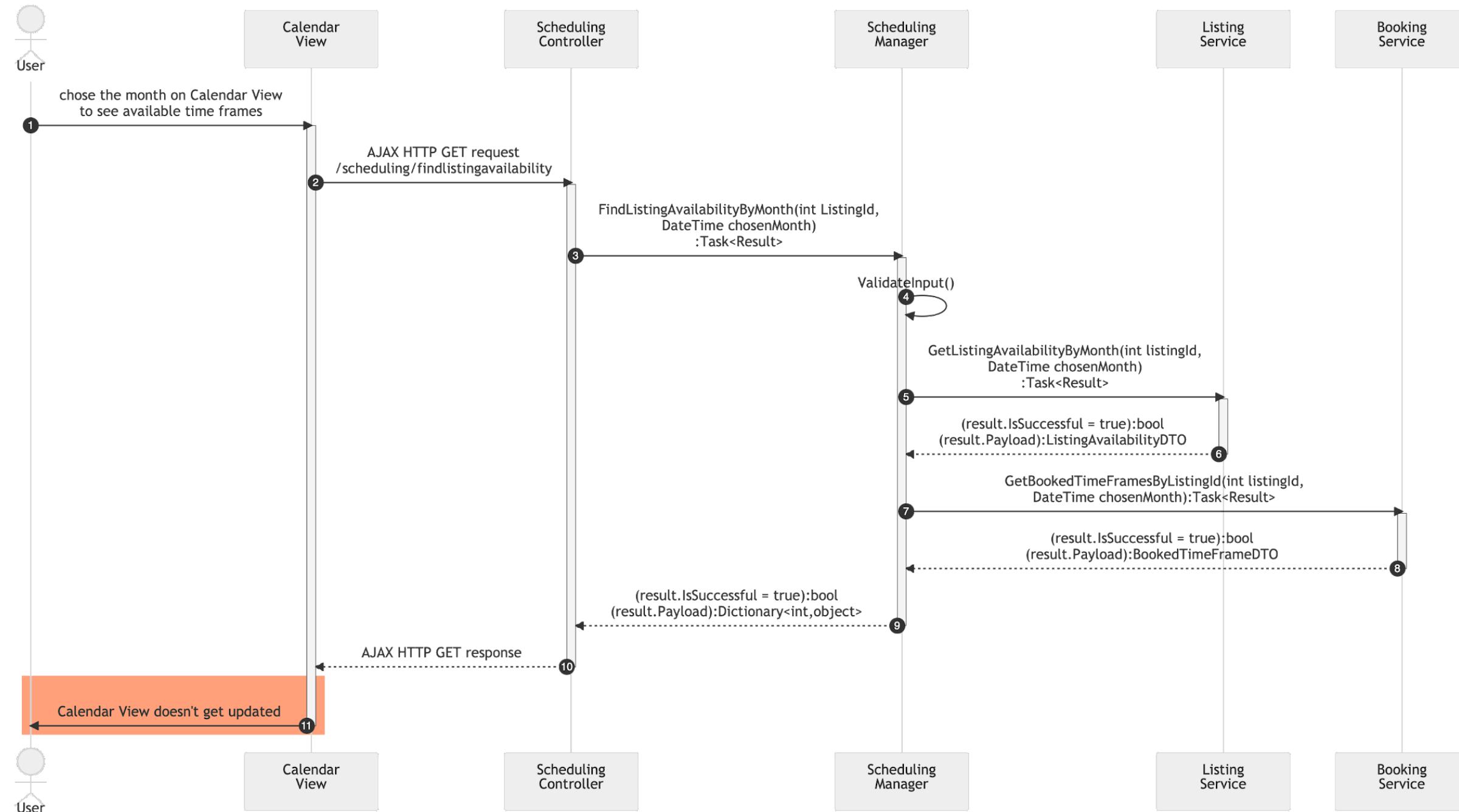
- Available time slots for the chosen time frame will be displayed.
- Available time slots are able to be updated according to the Calendar View filters.
- If there's no available time slots, Calendar View displays "No available time found. Please pick another date."



Failure Cases

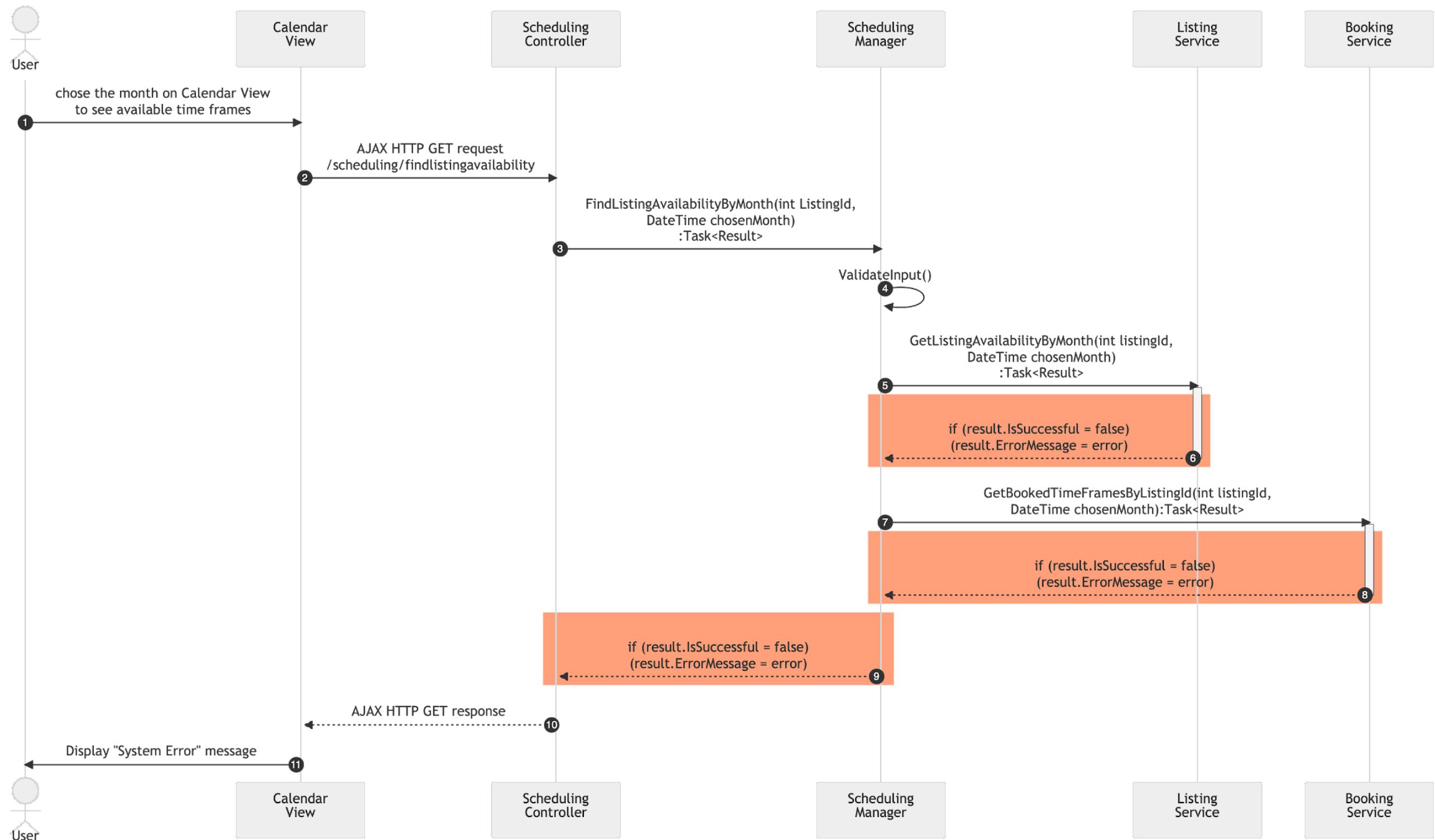
Failed case 1

- Calendar View doesn't display updates within 5 seconds.
- Available time slots are not updated when Calendar View filters change.



Failed case 2

- Any failure from Service Layer will cause the frontend to display a “System Error” message.



2. Booking a Listing in a time frame

User story: As a user, I can submit a booking with a selected time frame, confirm the selected time frame. System will display a confirmation message after booking successfully.

Business rules:

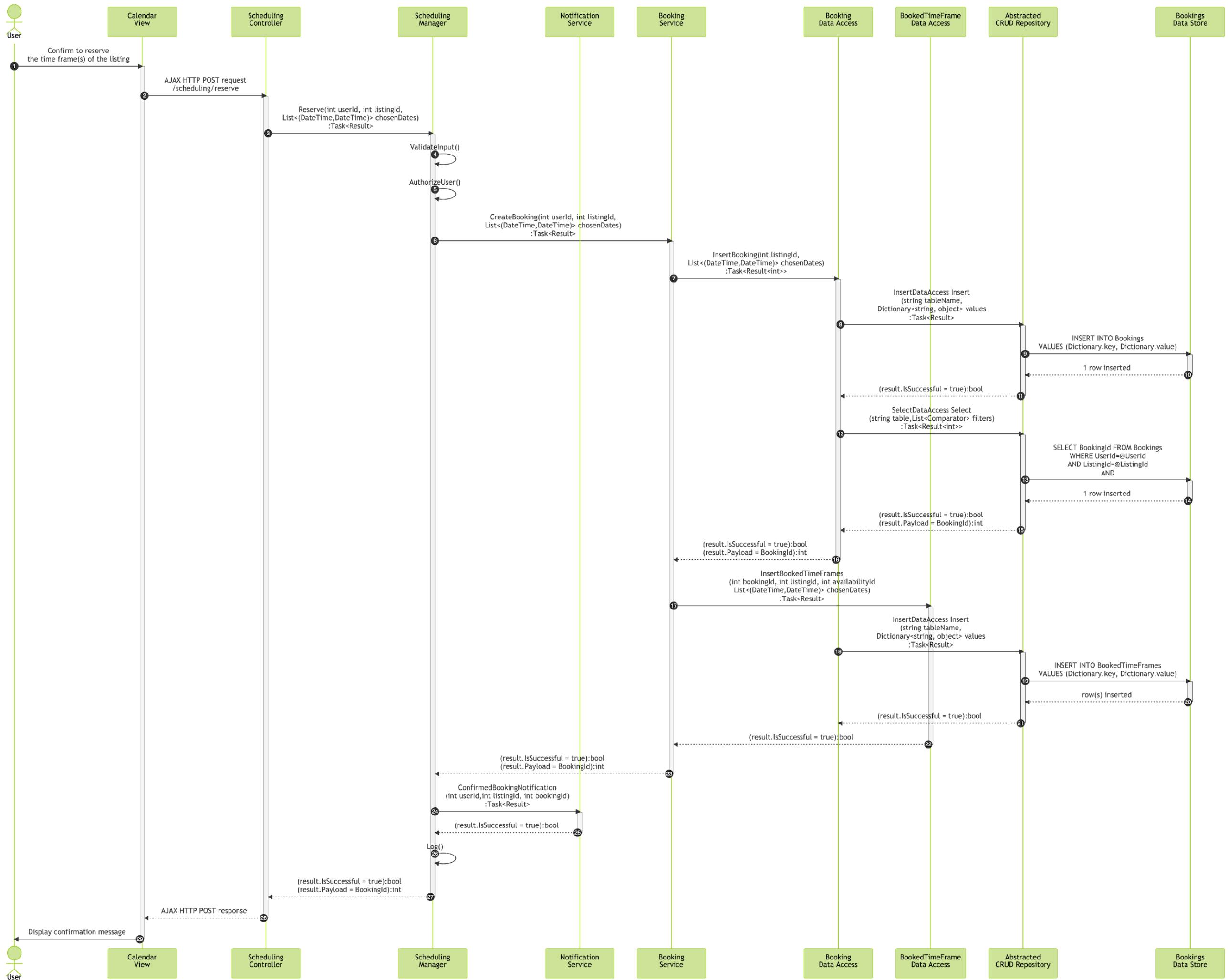
- Any authenticated user can perform a booking operation
- Confirmation of a time frame must be confirmed within 5 seconds
- User must confirm action from options of yes or no
- Selection of no returns the user to their previous view
- System failure from this feature will not cause the system to go offline.
- Error rate of users' booking must not exceed 5%

Preconditions:

- User has selected a listing
- User must have selected time frame
- User must not be owner of listing
- User must be logged in

Success Case

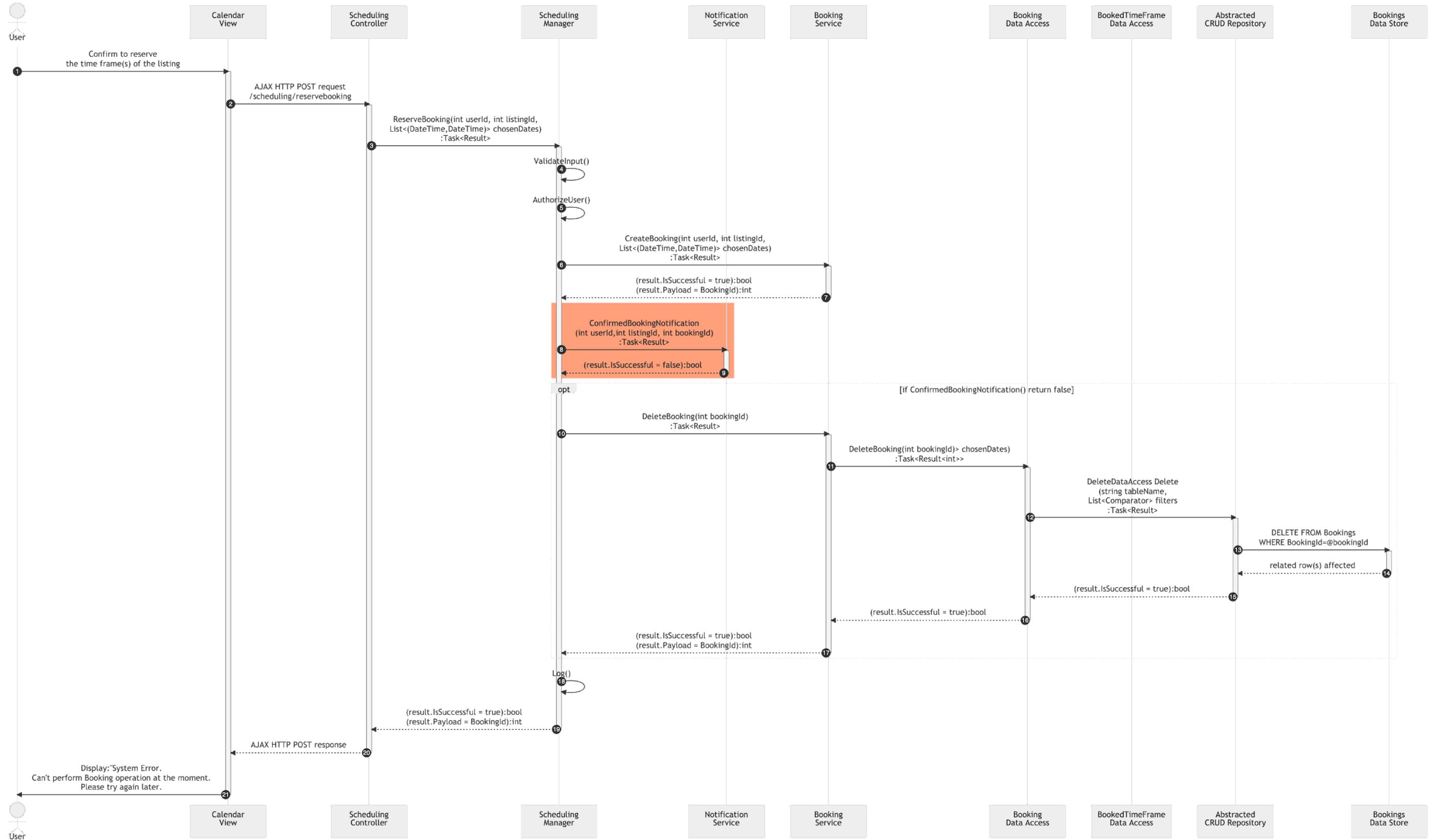
- There is a confirmation view of booking the selected time frame
- Notify user and host of the selection with listing information including date, location, price, and time frame
- The listing's availability during the selected time frame is indicated as unavailable
- Booking information is stored in database



Failure Cases

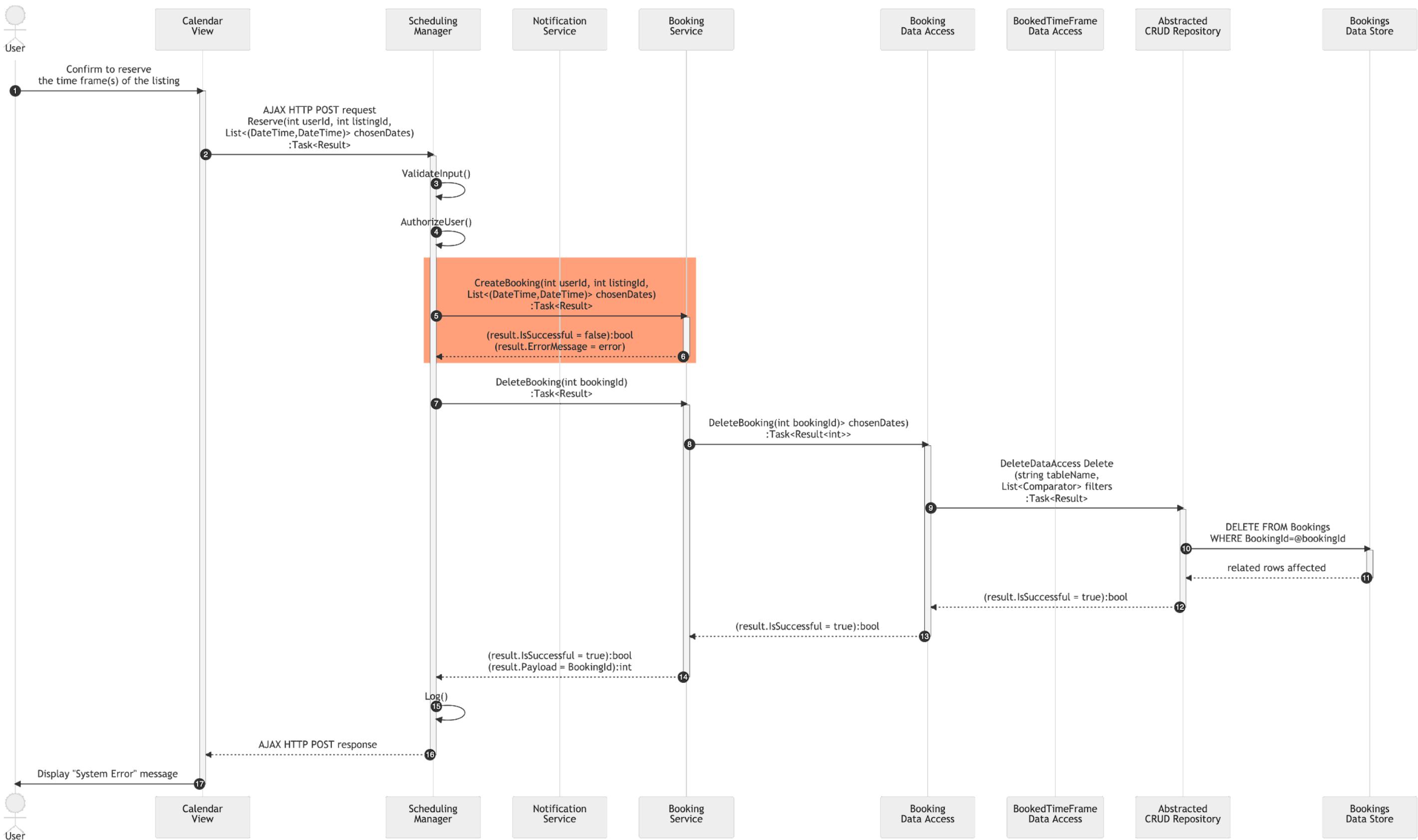
Failed Case 1

- Confirmation of booking is not sent to host



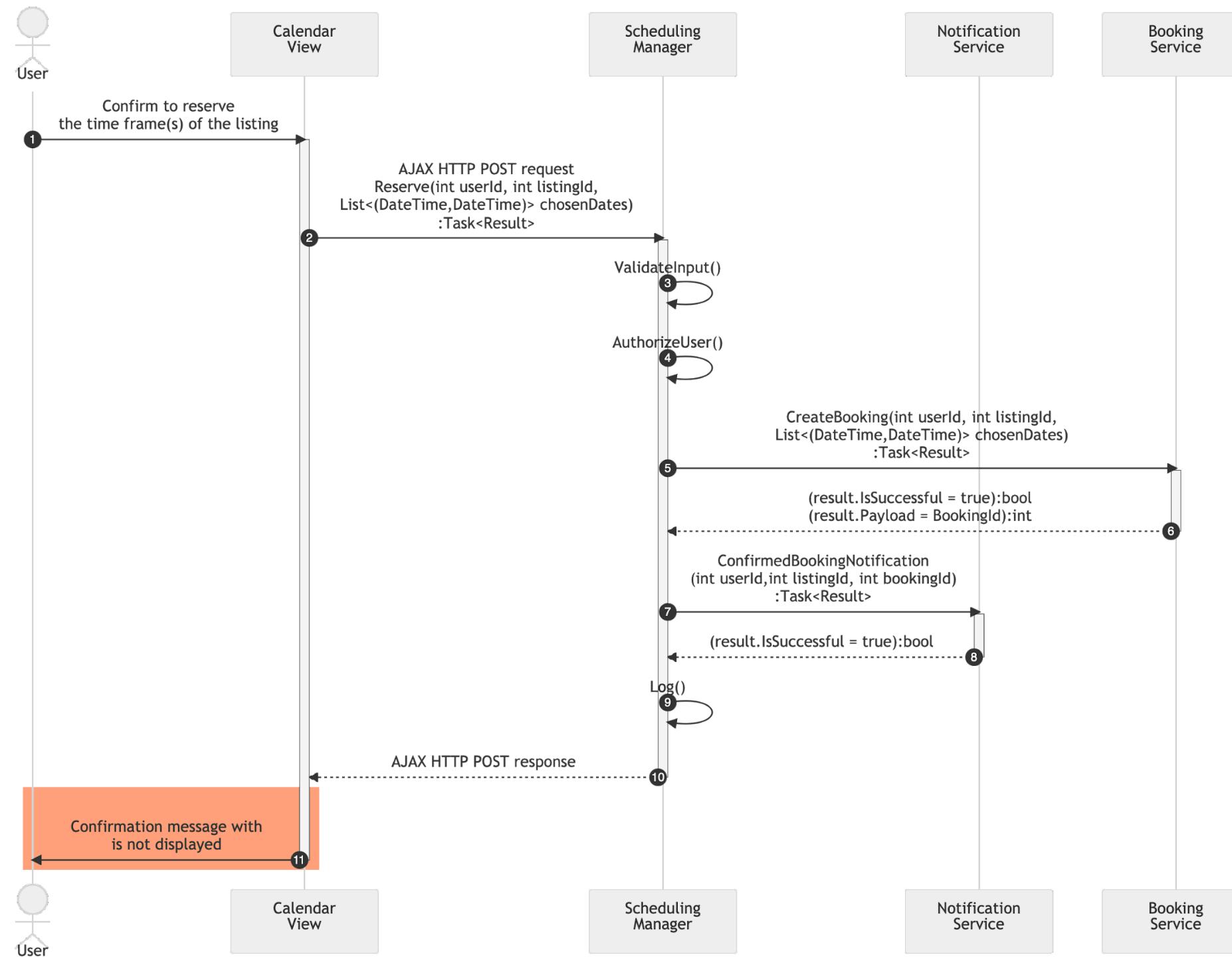
Failed Case 2

- Failure from Booking Service will cause the frontend to display “System Error” message



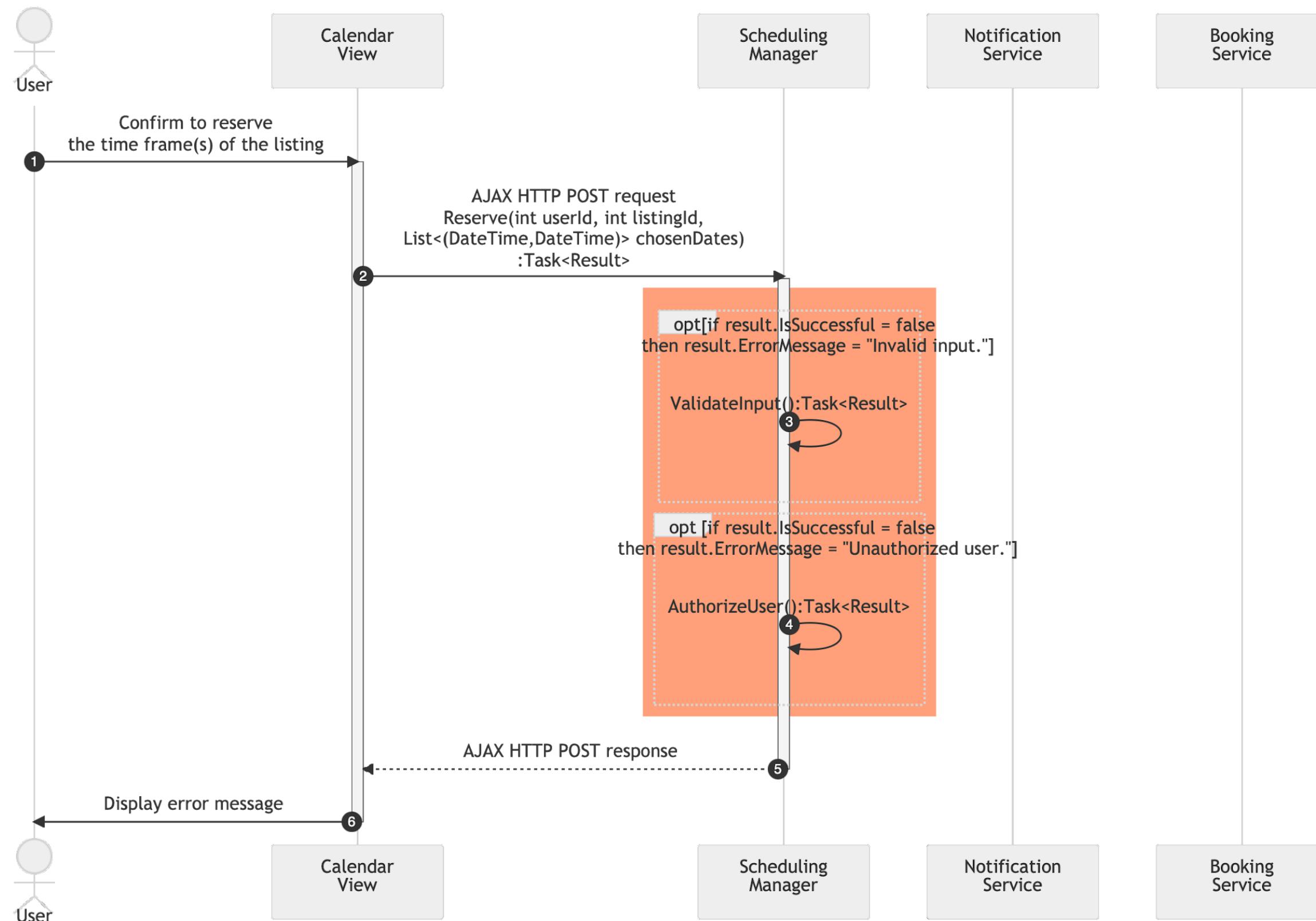
Failed Case 3

- Confirmation box of booking is not displayed.



Failed Case 4

- Input validation or User authorization fails



3. Cancel a Booking

User Story: As a user, I can submit a request to cancel a selected time frame of my booking, then confirm the cancellation. System displays a confirmation message upon a successful cancellation.

Business Rules:

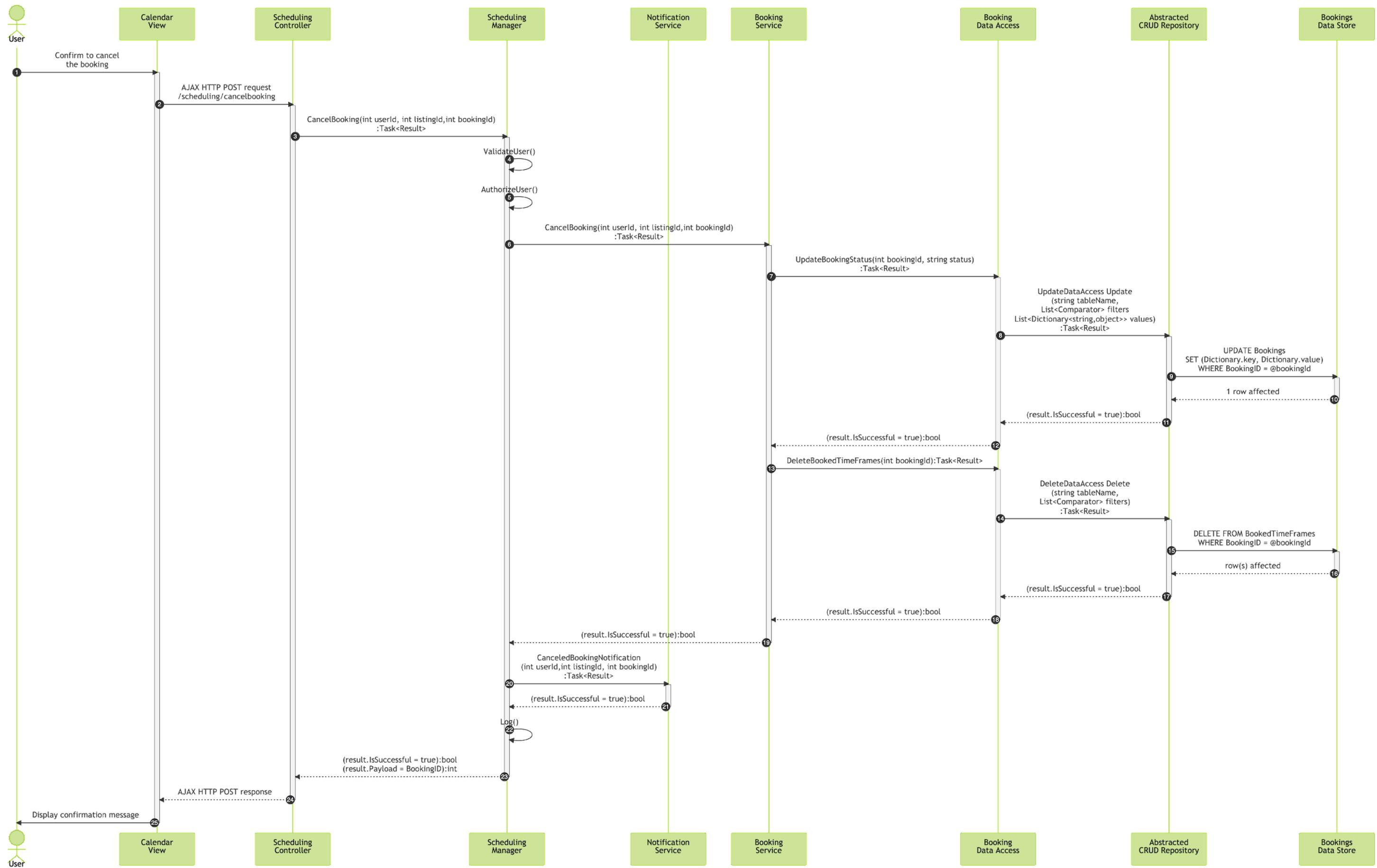
- Cancellation of a time frame must be confirmed within 5 seconds
- User must confirm action of cancellation from options of yes or no
- Selection of no returns the user to their previous view
- Error rate of users' booking cancellations must not exceed 10%

Preconditions:

- User already selected Booking from Booking History view
- User must have a booking
- User is not the owner of the Listing
- User must be logged in

Success Case

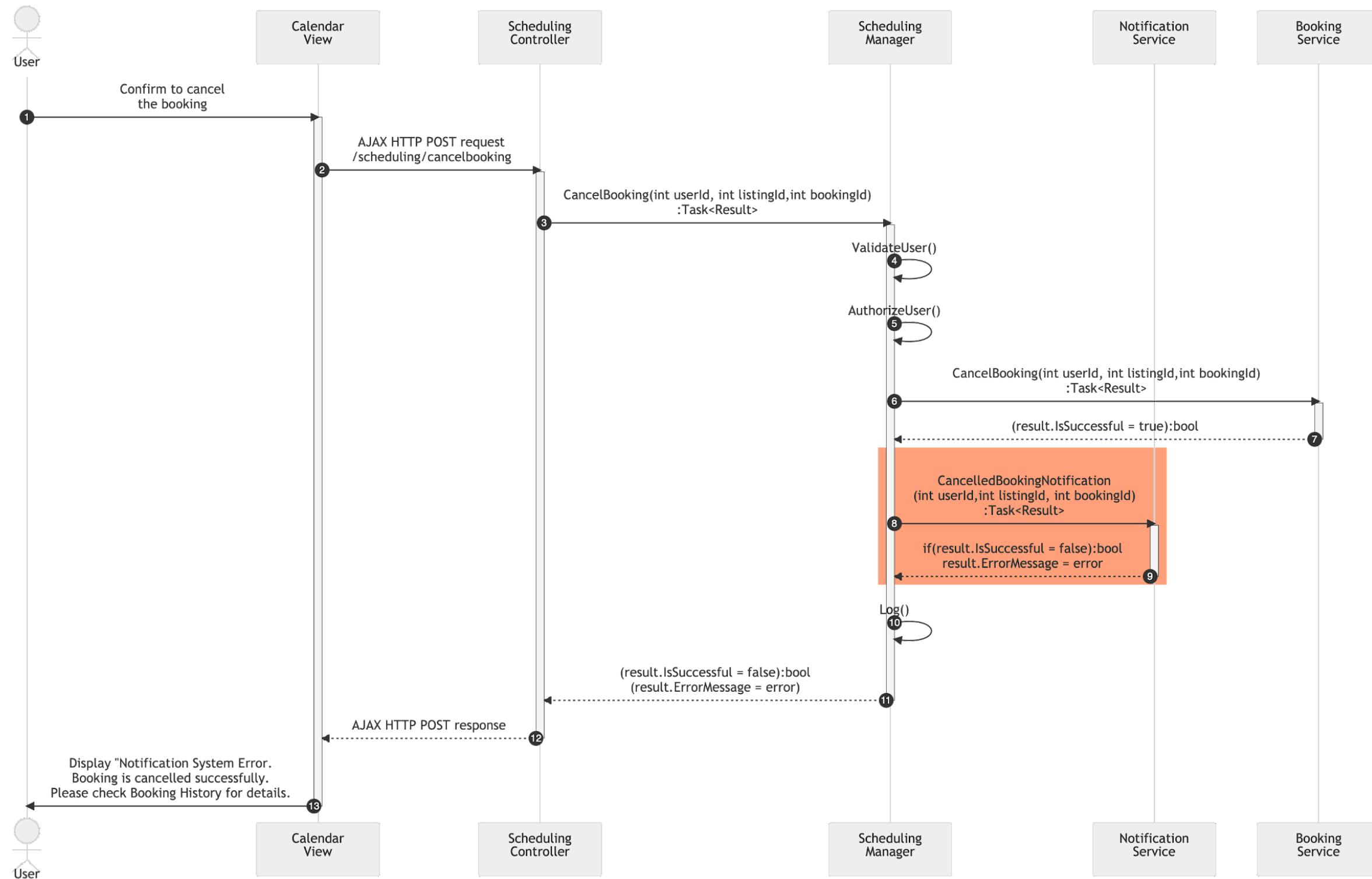
- There is a confirmation box of cancellation the selected time frame
- There is communication to the host of the cancellation
- The listing's availability during the selected time frame is indicated as available
- A confirmation email is sent to the user with cancellation information including date, location, and time frame



Failure Cases

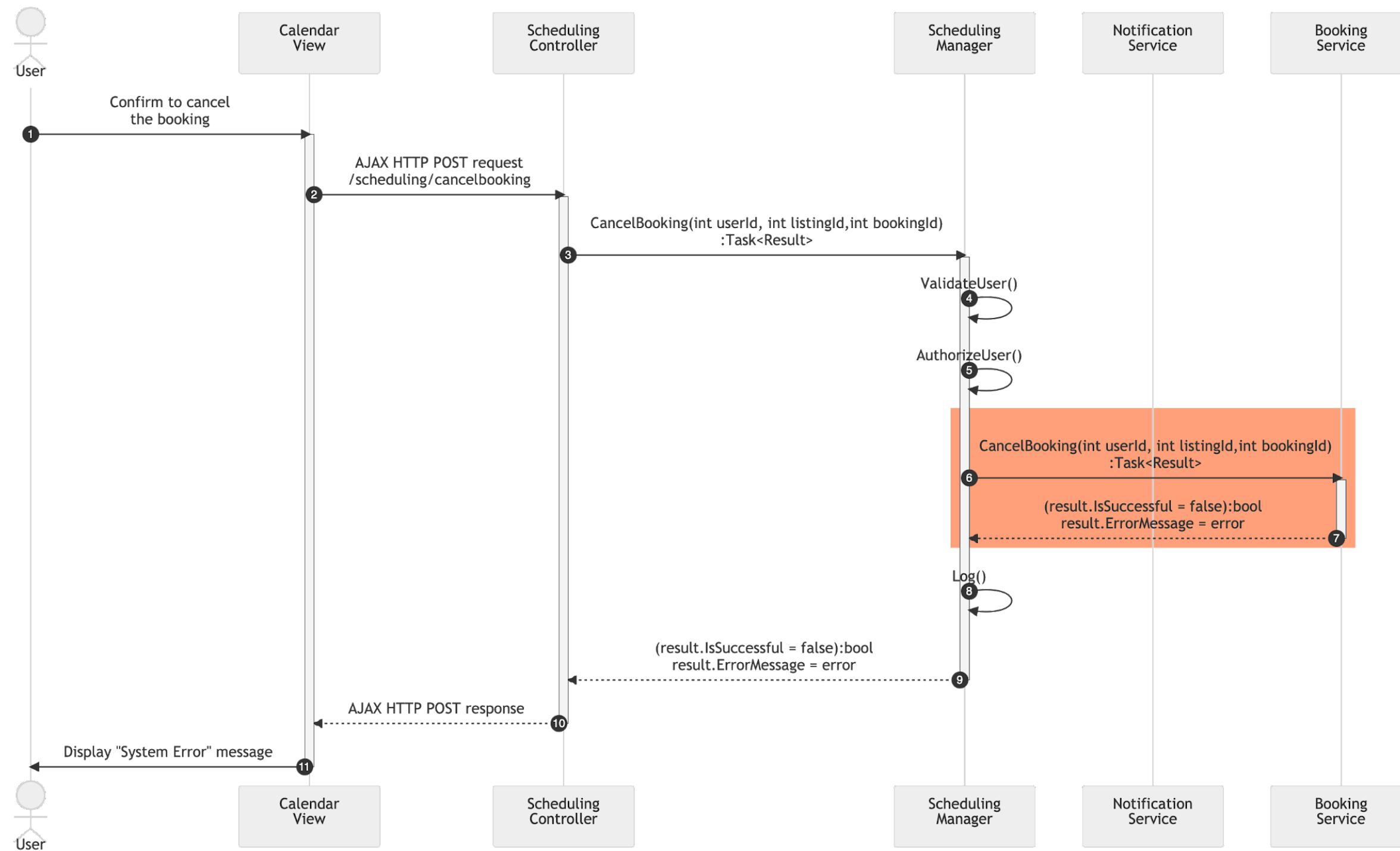
Failed Case 1

- Confirmation of booking cancellation is not sent to user
- Confirmation of booking cancellation is not sent to host



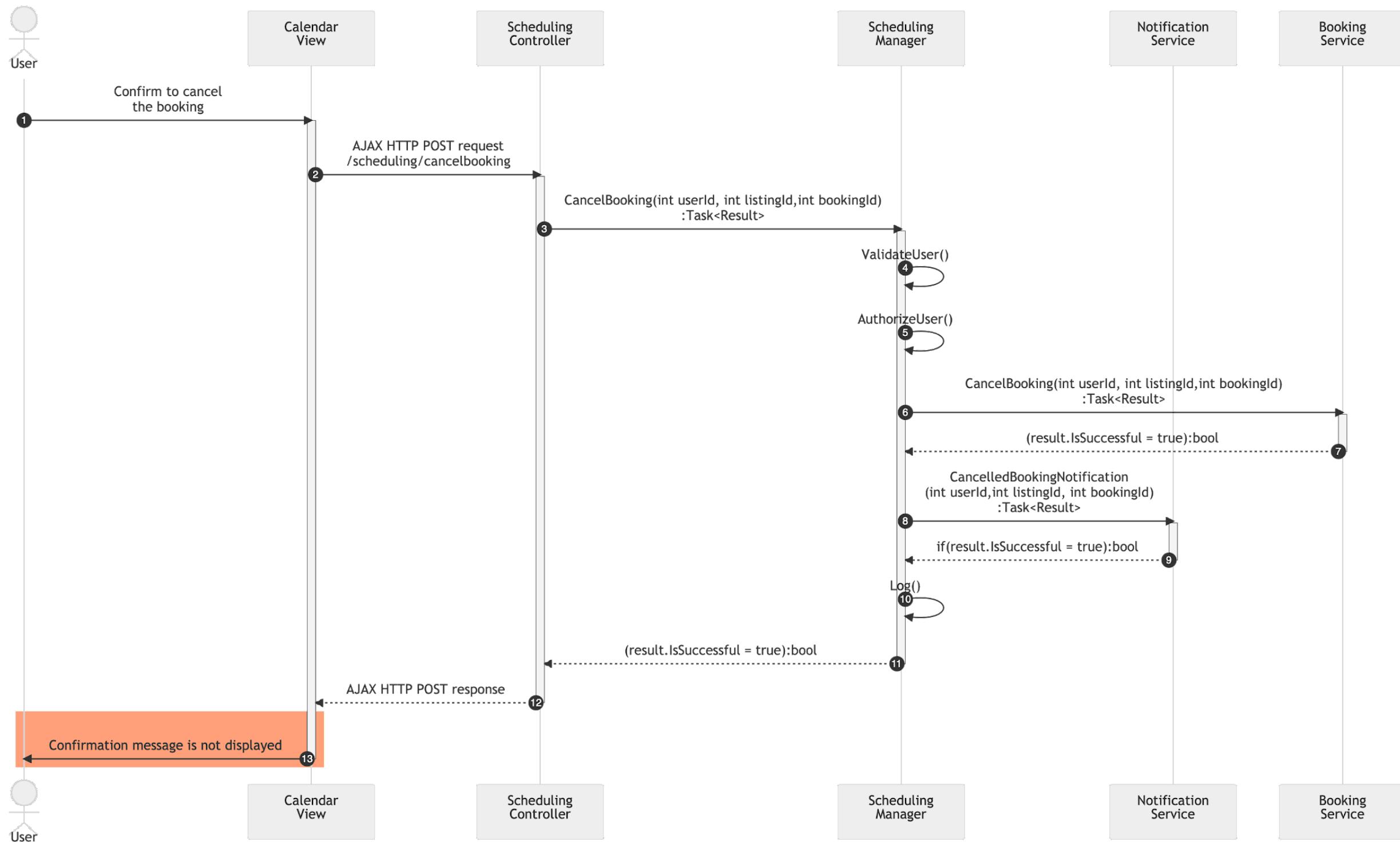
Failed Case 2

- Any other failure from the backend will cause the frontend to display “System Error” message



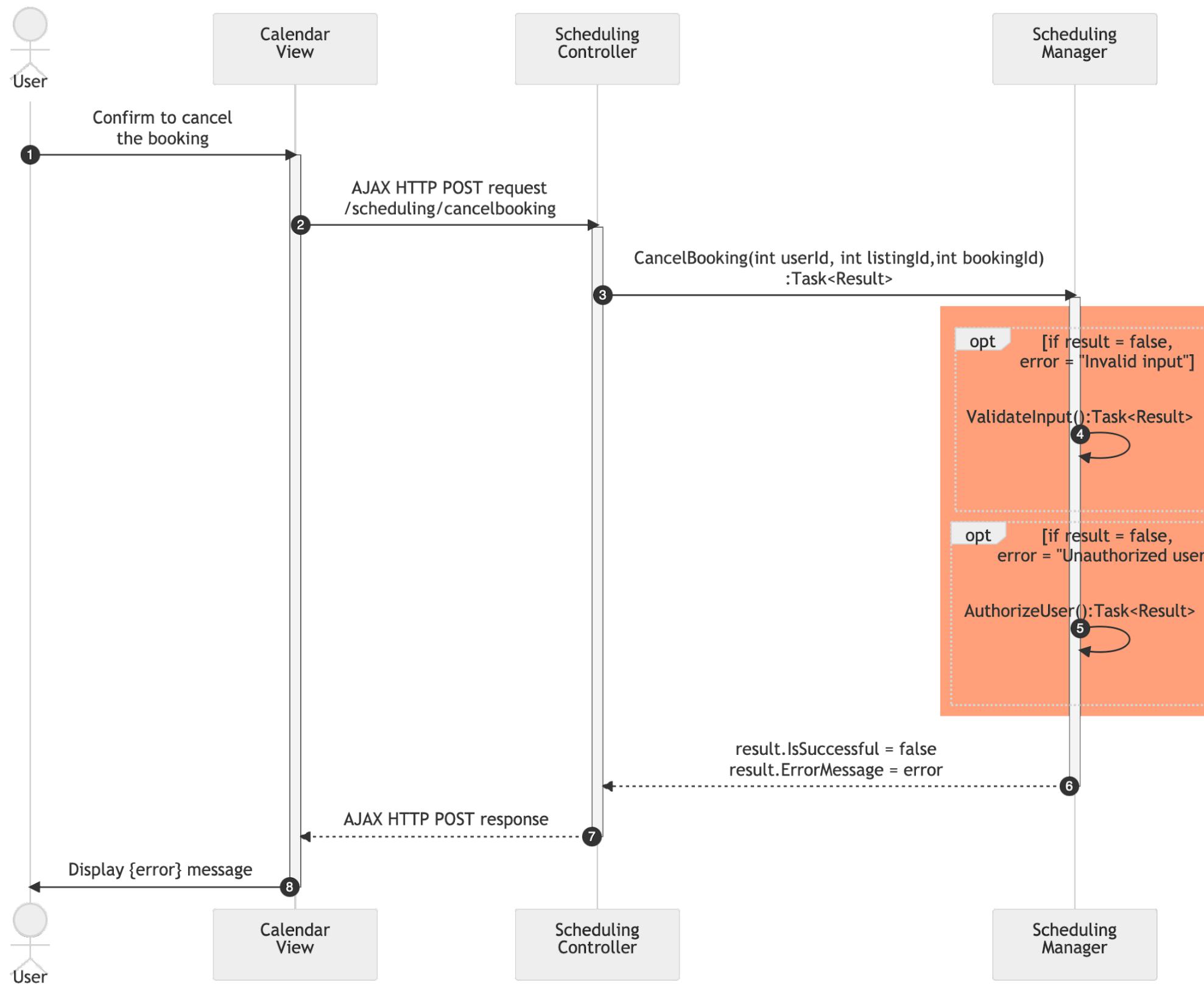
Failed Case 3

- Message on application view does not indicate cancellation



Failed Case 4

- Input validation or Authorization fails



References

This document elaborates Business Requirements Document³ for one of the product's features - Scheduling System.
The requirement is designed based on the system architecture provided in High-Level Design Document⁴.
The diagrams included in this document utilized diagrams.net⁵ and Mermaid.js⁶.

³ Business Requirements Document, URL: https://github.com/DevelopmentHellaHell/SeniorProject/blob/main/docs/BRD/DevelopmentHell_BRD_v.3.2.pdf

⁴ High-Level Design Document, URL:
<https://github.com/DevelopmentHellaHell/SeniorProject/blob/b43182c4076d471ef03520052675f1f88371dafd/docs/HL%20Design/DevelopmentHell%20HLD%20v1.2.pdf>

⁵ <https://www.diagrams.net/>

⁶ <https://mermaid-js.github.io/mermaid/#/>