**Use Case 1**

**Use Case:** User wants to cancel a booking

**Primary actor:** User

**Goal in context:** To cancel a room booking

**Preconditions:**  User should log in to use service. A booking must have already been made. Either user provide booking details and click on the cancel button or go to show booking page and click on the booking which the user wants to delete and click on the cancel button

**Trigger:** User provides booking information and clicks the cancel button

The support user logs in and sees the Support page. They will click on delete request number to populate the information for the request. They can Approve or Deny the request.

**Scenario:**

1. User: logs into the app
2. User: selects Schedule button from the Profile page
3. User: provide booking details and clicks Cancel Booking button or click on the booking to be deleted and click Cancel Booking button
4. User: presented with a message that says booking cancel is on pending, and the user will be taken back to a Profile page, booking to be canceled will have updated status on the list
5. Support: logs into app
6. Support: selects a cancel request number to populate the old schedule information, and reason for the cancelation
7. Support: can either Approve or Deny the request

**Exceptions:**

1. User will not be able to delete booking which doesn’t exist or is already on pending

**Priority:** Essential, must be implemented

**When available:** Second iteration

**Frequency of use:** Whenever a user wants to cancel a reservation.

**Channel to actor:** Via GUI Interface

**Secondary actors:** Support

**Open Issues:**

1. When will be the latest time a user can request booking cancelation? (a week, a day before?)

Use cases for other User interactions will be created in a similar format. Please read and understand each use case. If elements of the use case are not clear, you may request a review for additional information or changes.