

**UFCF8S-30-2**

**Advanced Software Development 22sep 1**

**Part-1: Analysis & Design of a software system**

**James Lear - Practical - CP1 03 Thursday 1430-1830**

**Date of Submission: 13/11/2022**

**Group 4**

**Members:**

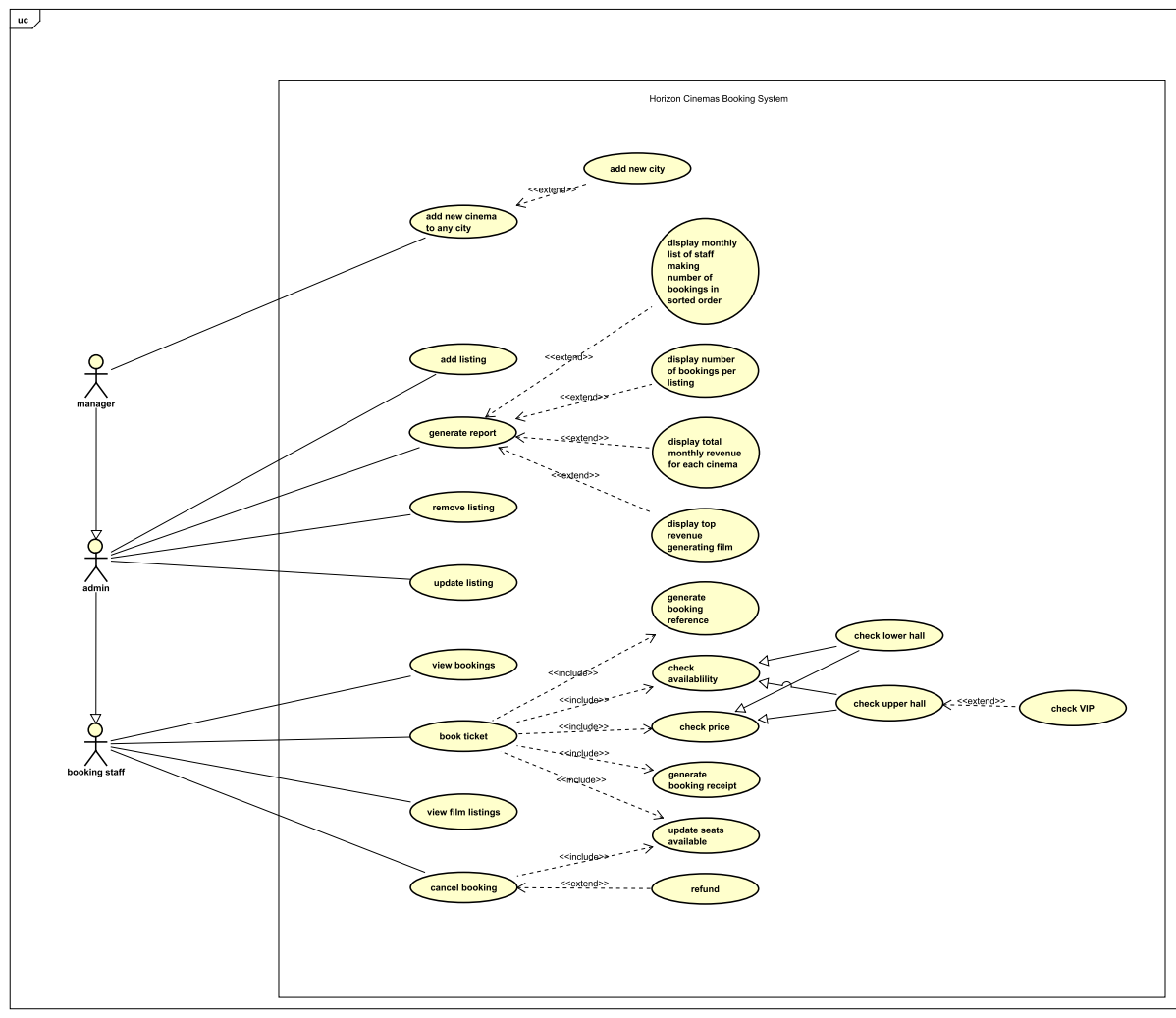
**Rohaam Aslam: 21017718**

**Benjamin Hussey: 21022768**

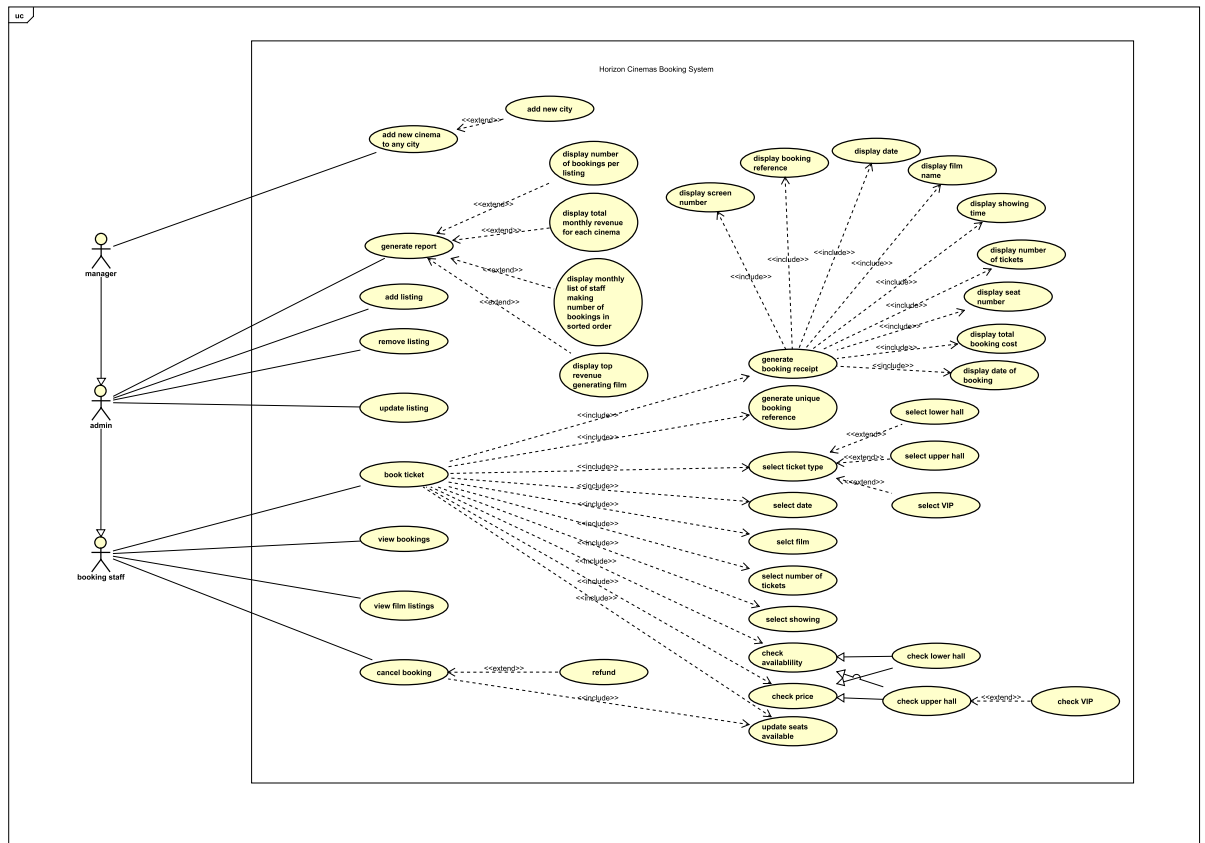
**Yazeed Abu-Hummos: 21014295**

# Use Case Diagrams & Descriptions:

## - Standard Use Case Diagram



## - Detailed Use Case Diagram



## - Use Case Description

- **Name:** cancel booking
  - **Actors:** Booking staff / Admin / Manager
  - **Pre-condition:**
    - The show is in one or more days
    - There are available seats
  - **Base Path:**
    1. User checks it is not the day of the show
    2. User checks if the list of bookings for the show
    3. User issues a cancellation order
    4. System cancels booking
    5. System issues a 50% refund
  - **Post-condition:** System updates available seats
  - **Alternative Paths:**
    - **The show is today or was missed:** The user informs the customer that a cancellation is not allowed
  - **Special Requirements:** N/A

- **Name:** view film listings
  - **Actors:** Booking staff / Admin / Manager
  - **Pre-condition:** N/A
  - **Base Path:**
    1. User chooses filters for the listing wanted to view
    2. System queries available listings based on the filters
    3. System returns listings
  - **Post-condition:** user can view filtered listings
  - **Alternative Paths:**
    - **Booking staff tries to view listings from other cinemas:** System declines
    - **Admin or manager wants to view listings:**
      1. User chooses City
      2. User chooses Cinema
      3. User chooses filters for the listing wanted to view
      4. System queries available listings based on the filters
      5. System returns listings
  - **Special Requirements:** if the user wants to view listings at a different cinema, then booking staff cannot do that, should be either an admin or manager
  
- **Name:** book ticket
  - **Actors:** Booking staff / Admin / Manager
  - **Pre-condition:**
    - The show has not started yet
    - There are available seats
  - **Base Path:**
    1. User checks available shows for the requested film listing
    2. User chooses screen based on customer's desired time
    3. User checks if the seat-hall that the customer wants to reserve in is not full
    4. User fills in booking form
    5. User checks if the customer wants to proceed after showing the price
    6. User fills in check out information
    7. System generates booking reference, ticket and receipt
  - **Post-condition:** System updates available seats
  - **Alternative Paths:**
    - **No upcoming shows for the desired film:** Check with other cinemas or if the customer wants to book a different film (might need a different user type)
    - **No seats available in the desired hall:** check with the customer if they want to reserve in a different hall if available, otherwise check availability for other shows or in nearby cinemas (might need a different user type)
    - **Customer no longer wants to reserve:** stop at step 3
  - **Special Requirements:** if the user wants to book at a different cinema, then booking staff cannot do that, should be either an admin or manager

- **Name:** update listing
  - **Actors:** Admin / Manager
  - **Pre-condition:**
    - Listing exists
  - **Base Path:**
    1. User chooses City
    2. User chooses Cinema
    3. User chooses Listing
    4. User fills in new information
    5. User Submits form
    6. System updates Listing in database
  - **Post-condition:** Customers who have a booking for this listing get notified
  - **Alternative Paths:**
    - **Listing does not exist:** User can look for a different listing
    - **User fills in invalid information (i.e. date and the past):** System does not update listing and requests user for correct data or to discard changes
    - **Listing is old:** System will not allow the user to change anything
  - **Special Requirements:** Listing is in the future
  
- **Name:** remove listing
  - **Actors:** Admin / Manager
  - **Pre-condition:**
    - Listing exists
  - **Base Path:**
    1. User chooses City
    2. User chooses Cinema
    3. User chooses Listing
    4. System checks there are no bookings on the listing
    5. User confirms removal
    6. System updates database
  - **Post-condition:** Listing and all its shows get removed and will not be available for booking again
  - **Alternative Paths:**
    - **Listing does not exist:** User can look for a different listing
    - **Listing has bookings:** This depends on the implementation. If it allows removal customers need to get a refund and to be notified, otherwise the system declines
  - **Special Requirements:** No bookings for the shows in the listing otherwise it would require refunds for booked customers

- **Name:** generate report
  - **Actors:** Admin / Manager
  - **Pre-condition:** N/A
  - **Base Path:**
    1. User chooses type of report
    2. System queries database for the information required in the report
    3. System formats data to a report
    4. System displays requested report
  - **Post-condition:** User will be able to view requested report
  - **Alternative Paths:**
    - **Not enough data available to generate report:** System informs user that the report cannot be generated
  - **Special Requirements:** N/A
  
- **Name:** add listing
  - **Actors:** Admin / Manager
  - **Pre-condition:** N/A
  - **Base Path:**
    1. User chooses City
    2. User chooses Cinema
    3. User chooses Film
    4. User chooses Screens and Times
    5. User Submits form
    6. System adds Listing to database
  - **Post-condition:** New listing and its shows will be available for booking from other Systems
  - **Alternative Paths:**
    - **Listing time clashes with another listing:** System asks user to modify the given listing information
    - **User tries to add new film that does not exist in the system yet:** System declines and the user has to add the film first
  - **Special Requirements:** New listing does not clash with other listings at the screen

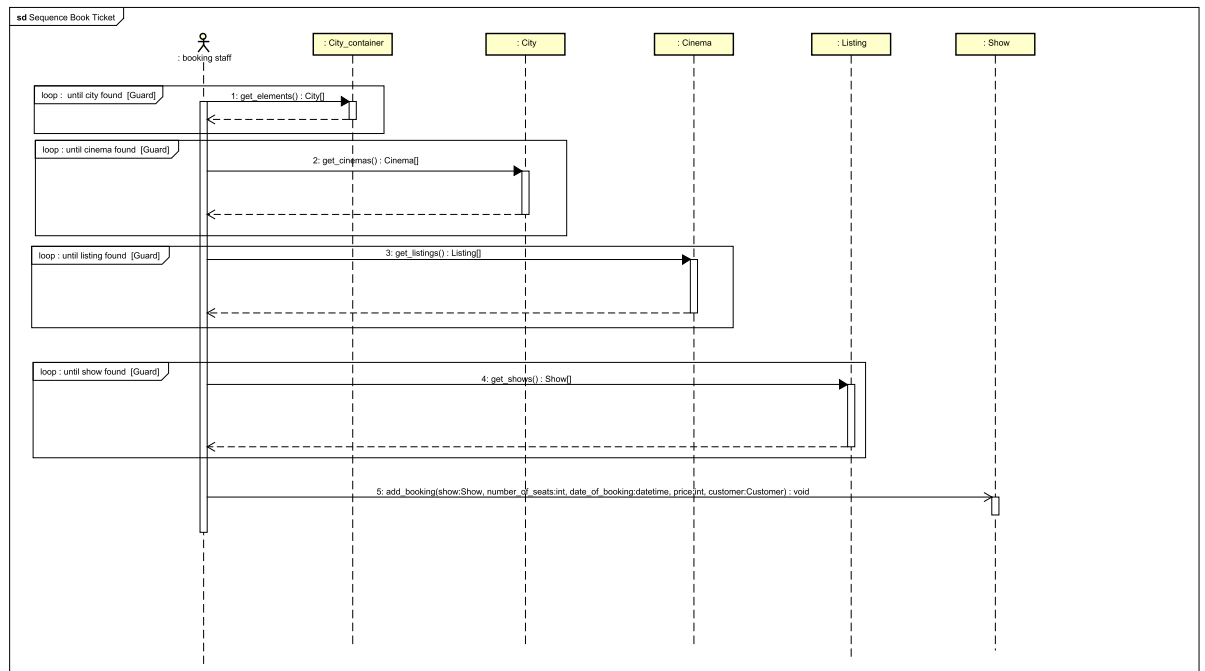
- **Name:** add new cinema to any city
  - **Actors:** Manager
  - **Pre-condition:**
    - City exists in the system
    - Cinema does not
  - **Base Path:**
    1. User chooses City
    2. User fills in new Cinema information
    3. User Submits form
    4. System adds Cinema to database
  - **Post-condition:** New Cinema will be available from other Systems
  - **Alternative Paths:**
    - **City does not exist:** User will have to add city first or choose a different city
    - **Cinema exists:** User will have to enter different information or discard
  - **Special Requirements:** If user wants to add a cinema to a new city, the city has to be created first.

[illegible]

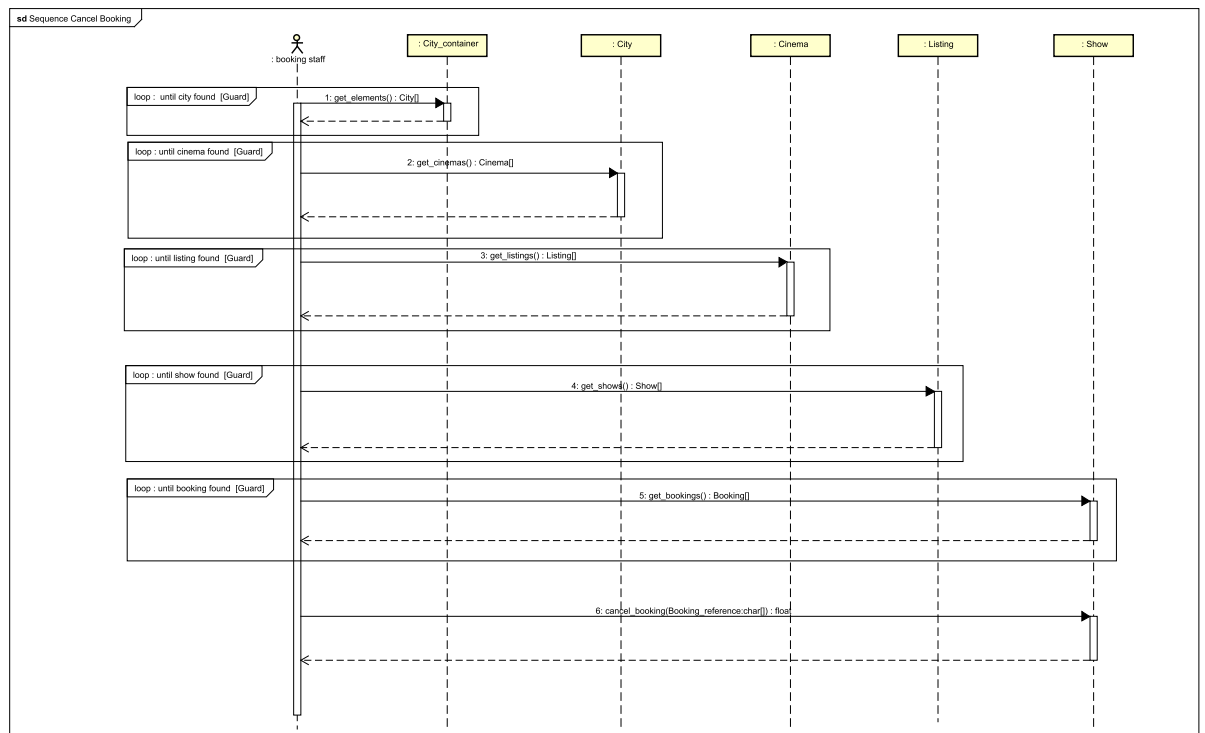


# Sequence Diagrams:

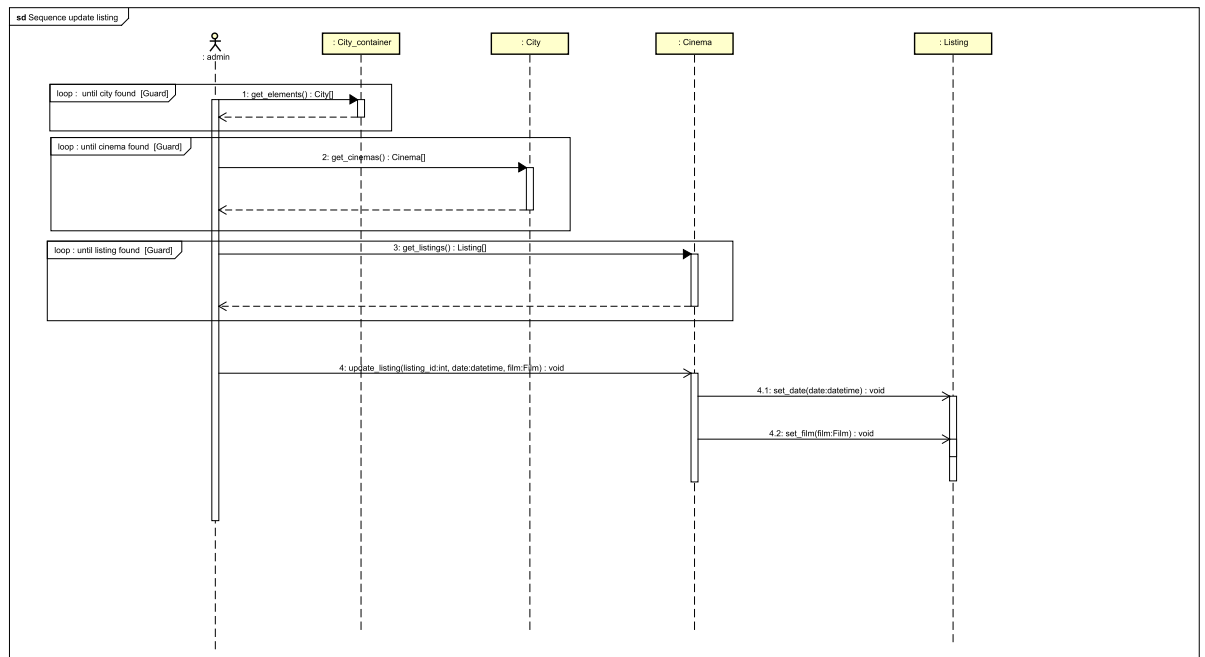
## - Book Ticket



## - Cancel Booking



## - Update Listing



## - Add Cinema

