Feature Name: Space Booking Center

Developer: Carine Gordillo

Developer Submitted: 03/14/2024

Peer Reviewer: Kay Kayale

Review Completed: 03/16/2024

Major Positives:

1. Methods are detailed
    - Each method being called is specified with parameters and parameter types
2. Timing Operation
    - Good use of timers throughout the design to check for business timing rules
3. Clear success feedback to user:
    - The system provides clear feedback to users upon successful completion of an action. This clarity enhances the user experience by confirming that their actions have been successful
4. Good checking points
    - Operations are integrated to ensure operations will not cause errors and include valid user inputs

Major Negatives:

1. Improper response calls
    - Response objects are being created before functions are being called. Nothing is triggering your response calls currently.
2. Incorrect View functionality
    - The view is only there for the user to interact with
    - The view itself is not getting the element by Id, make sure you add a layer for your javascript that is performing these actions
3. Include logging details, if any
    - There lacks logging details or reference to logging diagrams
4. User Roles:
    - Unspecified which user roles have access to this feature
5. Inconsistent diagram:
    - Some validation that is being checked in failure diagrams is not shown being checked in success diagrams. Successful scenario will involve these validations.
6. Inconsistent operation numbering
    - Either remove all numbering or make them in numerical order

Unmet Requirements:

1. Company/Facility Display
    - The current design lacks mechanisms for data retrieval and presentation of a list of spaces available for reservation to the user
    - The design should include processes for querying the database to gather information on available spaces, including any relevant attribute and methods for dynamically displaying this list to users
2. User Role
    - The design does not specify the necessary user roles for accessing certain functionalities, particularly in relation to authentication and authorization levels required to make a reservation
3. Timeout scenario
    - Absence of a sequence diagram for handling failed scenarios where operations exceed a 3 second response time
4. Availability display information
    - The design does not adequately specify the information to be displayed regarding space availability, particularly missing start and end dates and times as stated in our BRD
5. Reservation conflicts
    - The system's current design fails to include a verification step to check if a requested reservation spot is already occupied
    - System fails to check if user inputs end time before start time
6. Incomplete confirmation message
    - As stated in our BRD, confirmation message of reservation information is displayed in the following order:
        - Name of Company or Facility selected
        - Chosen Space ID
        - Start and End Date and Time
        - Authorized User's username that submitted reservation
        - “Please confirm the reservation was created by checking you personal overview center”

Design Recommendations:

1. Incorporate use of models and objects in design
    - Enhance the system's architecture by incorporating well-defined models and objects that represent the various entities involved in the reservation system, such as Companies, Spaces, Reservations, and Users.
    - Pros:
        - Improved data organization: Utilizing models and objects allows for a clearer representation of the system's entities and their relationships
        - Enhanced scalability: it becomes easier to add new features or modify existing functionalities without extensive redesign
        - Efficient data manipulation: allows for more robust data validation and manipulation methods directly within the models
    - Cons:
        - Potential for Complexity: As the number of models and their interactions increase, the system could become more complex, necessitating careful management and documentation to maintain clarity
2. Role Based Access Control (RBAC)
    - Enhance the RBAC system to include more granular roles and permissions, particularly for different levels of management and employees. Include how employees are the only types of user that have access to the corresponding company reservations
    - Pros:
        - Enhanced Security: Tighter control over access rights ensures that sensitive operations are performed only by authorized personnel
        - User Experience: the system can offer a more relevant and streamlined experience to each user type
    - Cons:
        - Increased complexity: adding a more nuanced RBAC system could lead to more complexity to the application

3. Prebuilt Commands
    - Adopt the use of prebuilt SQL commands for frequently accessed queries, such as retrieving the list of all companies, facilities, and their associated spaces.
    - Pros:
        - System reliability: remove need for services to build queries, the system's reliability is enhanced, reducing potential points of failure
        - System resources: prebuilt commands reduce the necessity for additional asynchronous system calls, thereby conserving system resources and improving performance
    - Cons:
        - Customizability: may necessitate additional queries to accommodate specific user access levels, such as employees, due to their unique access privileges compared to regular users
4. Enforce business rule in front end
    - Integrate business rule validations directly within the front-end application, utilizing input validation techniques to ensure data integrity before submission to the server
    - Pros:
        - Reduce server side load: the server is less burdened with processing invalid requests, leading to more efficient use of server resource
        - Enhanced user experience: Immediate feedback on input validity can make the interface more responsive and user-friendly
    - Cons:
        - Potential exposure of business rules: Incorporating business rules into client-side validations may reveal certain aspects of the application's logic to users, could be negative or positive aspect
        - More load on end user’s system: leverages the user's system resources, which could affect the performance of the application on less powerful devices

Test Recommendation:

General Recommendations

- Validation for Backend services: Ensure that all backend services, including SQL command builders, have fallback mechanisms or redundancies in place to handle failures or offline scenario
- Confirmation Message Completeness: Ensure that the reservation confirmation message includes all required details (as specified in the BRD) in the correct format and order.
- Historical integrity: Ensure that any modifications to reservations or spaces maintain the integrity of historical reservation data, allowing for accurate reporting and analysis
- Logging: regardless of how comprehensive testing is developers will be unable to adapt system in the presence of errors without comprehensive, detailed logs

Reservation

- Reservation Time Logic: Test the system's ability to reject reservations where the end time is set before the start time, regardless of frontend validations, to ensure backend integrity.
- Reservation of Occupied Space: Test the system's ability to prevent users from reserving a space that is already occupied or booked by another user for the same time slot. This includes verifying that the system checks for overlapping reservations and accurately reflects the unavailability of the space before allowing a new reservation to proceed

Availability display

- Real time availability Updates: Ensure the system accurately reflects real-time changes in space availability, including immediate updates after new reservations or cancellations
- Invalid date range: Test the system's response to invalid date range in availability queries, ensuring it properly handles or rejects such requests

Reservation Modification

- Modification during reservation: Verify that the system prohibits modifications to spaces that have active reservations, to avoid conflicts or customer dissatisfaction
- Cancellation notification: Test the system's mechanism for notifying users of any modifications or cancellations to their reservations