Feature name : Scheduling
Developer name : Tien Nguyen
Date developer submitted : 04/15/2023
Reviewer name : Garrett Tsumaki
Date review completed : 04/16//2023

Summary:

Overall this requires additional review of the BRD for checks of meeting all requirements possible, although otherwise it is okay. Security checks are absent and must be added, as well as perhaps a look-over for typos. Encapsulation is also lacking and should be maintained: do not create functions specific to this feature in other services when the functions that execute tasks already exist within them. Place logic within the services and managers for this feature and call those functions from within.

Unmet Requirements:

- *Requirements that go unmentioned are assumed to be already met by the given design.

BRD - Business requirements

- Requirement text: “It is illegal to enter an end date which precedes a start date or a start date which precedes the current date, doing so will prompt the user to enter a legal time frame”
    - Unmentioned, assumed to be resolved during implementation. This should be checked after implementation is complete.
- Requirement text: “Dates will be determined using PST”
    - Unmentioned, assumed to be resolved during implementation. This should be checked after implementation is complete.
- **Requirement text: “A user will only view the calendar on a month-to-month basis, starting with the last week of the previous month and extending to the first week of the next month.”**
    - Wireframes show only the current month, starting from the first of the month. Missing “the last week of the previous month” and “the first week of the next month”
- Requirement text: “Changing the currently viewed month is limited to the current month and 6 months in advance.”
    - Unmentioned, assumed to be resolved during implementation. This should be checked after implementation is complete.

- **Requirement text: “Each week of a month will start with Sunday and always list every day of the week even if there is no time frame available for booking on that day.”**
    - Due to the starting on the first of the month rather than including the previous week, it is not a guarantee that the first Sunday will be shown.
- Requirement text: “Cancellation of a booking must be requested 48 hours before the start date and time of the booking.”
    - Unmentioned, assumed to be resolved during implementation. This should be checked after implementation is complete.
- Requirement text: “In the event that a cancellation is requested within this time frame, the user will be asked to contact the host directly."
    - Unmentioned, assumed to be resolved during implementation. This should be checked after implementation is complete.

BRD - Change Monthly Calendar View’s Displayed Month

- Check ‘BRD - Business requirements’ section to see concerns regarding formatting of the calendar view.

BRD - Filter Calendar View

- **As there is no Sequence diagram, it is assumed that this exists solely in the wireframes. In the wireframes there is no clear location for this and does not seem to be present.**

BRD - Unfilter Calendar View

- **As there is no Sequence diagram, it is assumed that this exists solely in the wireframes. In the wireframes there is no clear location for this and does not seem to be present.**

BRD - Alter Selected Time Frame

- BRD mentions as Success Criteria “Previous time frames which are no longer selected are no longer highlighted”
    - **Not present in wireframes, although it seems like it will be present.** Assumed to be resolved during implementation. This should be checked after implementation is complete.

Booking of Listing Time Frame

- Many different parts go unmentioned. Assumed to be resolved during implementation. This should be checked after implementation is complete.

Document Clarity recommendations:

High level design - Find Listing Availability

- Numbering steps would be good, or color coding by stages and numbering those would work as well. As it is, it's hard to track the flow of data and process, and one just has to assume top to bottom.

High level design - Book a listing in a chosen time frame - Part I

- Maybe mention checks, for the sake of clarity, as none are mentioned, and could cause unexpected errors.
    - Make sure time is in future

High level design - Book a listing in a chosen time frame - Part III

- Same as Find Listing Availability advice.

High level design - Cancel a booking - Part II

- Same as Find Listing Availability advice.

Relational Tables

- Unclear if OwnerId in Listings Data store is inherited (It should be, otherwise we will have ghost listings for accounts that do not exist).

Wireframes

- It would be good to map the Requirements being fulfilled to the large wireframe, as it contains most of the UI functionalities. As it currently is, things like "Filter Calendar View", "Unfilter Calendar View" are too hard to find to verify that they are fulfilled in accordance with the BRD

Design Recommendations

Relational Tables (Database design)

- Listings and Bookings are placed in separate datastores, despite the fact that they need to interact with each other explicitly. I recommend moving them to the same datastore, wherever it is found to be appropriate based on further examination by the designer.
    - Bookings exist ONLY with listings, and listings are useless without bookings. For this reason, they should be able to interact with each other more tightly
    - **Foreign keys CANNOT be created between databases in SQL Server, unless with trigger, under which they are technically no longer foreign keys. This would be good to specify if this is the intended path so that the reader can verify the possibility of the plan.**

Book a listing in a chosen time frame

- **This MUST go through our Authorization service, otherwise, we are allowing requests from anyone to delete anyone's booking.**
    - If AuthorizeUser() is meant to be an internal authorization, it should be moved to AuthorizationService instead to maintain encapsulation. If it calls from AuthorizationService, that call should be demonstrated or at least mentioned to reference the Authorization document.
- Low level design
    - Success Case:
        - Step 4: ValidateInput() should be from our ValidationService so that it can be used by all members. Beyond that, there is likely already one in there that does what is needed, and using the existing one or adding one will maintain encapsulation.
        - Step 24: Should not need to create a function just for booking notifications inside of NotificationService. If anything, that should be a function inside of the manager that calls existing functions within NotificationService for sending those notifications in order to maintain encapsulation.
    - Failure Cases:
        - General
            - Besides just not showing confirmation, an error should be shown to the user so that they may understand what happened instead of repeatedly clicking a button that may not be functional

Cancel a booking

- **This MUST go through our Authorization service, otherwise, we are allowing requests from anyone to delete anyone's booking.**

        - If AuthorizeUser() is meant to be an internal authorization, it should be moved to AuthorizationService instead to maintain encapsulation. If it calls from AuthorizationService, that call should be demonstrated or at least mentioned to reference the Authorization document.
- Low level design
    - Success Case:
        - Step 4: ValidateInput() should be from our ValidationService so that it can be used by all members. Beyond that, there is likely already one in there that does what is needed, and using the existing one or adding one will maintain encapsulation.
        - Step 20: Should not need to create a function just for booking notifications inside of NotificationService. If anything, that should be a function inside of the manager that calls existing functions within NotificationService for sending those notifications in order to maintain encapsulation.
    - Failure Cases:
        - General
            - Besides just not showing confirmation, an error should be shown to the user so that they may understand what happened instead of repeatedly clicking a button that may not be functional

Find Listing Availability

- Low level design
    - Success Case:
        - Step 4: ValidateInput() should be from our ValidationService so that it can be used by all members. Beyond that, there is likely already one in there that does what is needed, and using the existing one or adding one will maintain encapsulation.
        - Step 14: Missing Call to BookingDataAccess
    - Failure Cases:
        - Failed case 1
            - Should show response, not just cases in which it fails. If the frontend doesn’t know what failed or how, we cannot show the right message to the user.

Wireframes

Booking date selection

- The single date section should likely be removed
    - There are still both “start date” and “end date” boxes there for the single date selection
- The multi date section should allow for multiple times to be selected for a day

    - The button for this is available, although due to the issue with the start and end boxes, it is unclear if this button is usable
- The differentiation between single and multi day booking is required, but instead, the designation between the two could be labeled depending on if you have more than one date selected or not, with the text switching depending on the number of days selected. This would reduce the amount of work required, while also compressing diagrams and allowing for easier reading and understanding by both readers of the designs and by the users themselves, as it is not clear, for example, what would happen if a user decided to select the multi-day option and only select a single day.

General

- Hide or gray out interactable elements that the user cannot presently interact with.

Cancel Booking & Confirm Booking

- **In our current state, we cannot process transactions. The current wording is misleading and could result in user misinterpretation of our services.**
    - The price totaling looks nice, but we do not process taxes, and that would be up to those that are providing the service.
    - During cancellation, refund should not be mentioned, as that is not the service we provide (currently).
        - Could be altered to be “Request refund” instead, perhaps an additional feature that will notify the owner of the Listing of the request so they may conduct a conversation off-site, as we do not have the authority to assist with refunds if funds never passed through our site.