

Group A Analysis Report

1. Missing Information and Clarifications Needed:

Analyzing the Last Resort Hotels requirements, we found many areas where we needed additional information such as the following:

1.1 Room Status and Availability

- The specific workflow for room status transitions is not clearly defined, such as how a room moves from "under renovation" to "available"
- How the housekeeping staff changes a room's status from cleaned to ready
- The detailed mechanism for determining if early/late extensions (up to 2 hours) can be granted

1.2 Pricing and Billing

- The exact formula for calculating discounts for multiple non-eating meeting room usages
- The specific rules for determining when meeting room charges should be reduced or waived based on guest count
- The exact surcharge amount for extended stays beyond the standard checkout time
- The prices of hosting an event, and how organizers would pay (over the phone, before the event, after the event)

1.3 Room Adjacency and Layout

- Technical specifications for the graphical representation of room and wing layouts
- Detailed information about movable walls (dimensions, configurations, tracking mechanisms)
- Procedures for reconfiguring large meeting rooms with movable walls

1.4 Customer Tracking

- Specific privacy rules regarding the confidentiality of guest whereabouts
- Policies for handling message relay and guest contact

1.5 Reservation and Event Management

- We were not sure about the algorithm for assigning rooms based on wing sequence numbers starting from the lowest
- It's unclear what the process for handling room changes during a guest's stay is

1.6 Beds in Room

- It's not clearly stated that if the beds in the rooms can be moved to other rooms when needed. Based on this, we need to track the beds.
- There is a bridging table to help manage the information on which bed(s) are in the room.

2. Assumptions Made:

To address some of these gaps in information, we made the following assumptions:

2.1 Billing and Payment Assumptions

- We're assuming that multiple guests can split a bill in any manner they agree upon.
- We've assumed that when a guest switches rooms during their stay, the system should still generate a single composite bill. This is handled through the relationship between customers and reservations, where multiple reservation details can be linked to a single customer.
- For meeting rooms, we've assumed that on default, one party is responsible for payment, while sleeping rooms can have split billing arrangements.

2.2 Room Management Assumptions

- We assumed that rooms are assigned using a simple priority based on wing sequence number and room number, similar to the one in the requirements.
- We assume there is a standard flat fee surcharge for extensions beyond the 2-hour grace period

2.3 Meeting Room Assumptions

- We assume standardized time slots for meeting rooms such as an hourly based system with fixed start and end times.
- We assume the system will automatically assign one free non-eating slot for each paid eating slot based on simple rules.

2.4 Technical Assumptions

- We assume that room layout visualization will be handled by a separate module that reads data from our database.
- We assume the card access system will update the access_swipe table in real-time to keep track of when someone has entered which location

2.5 Room Function Assumptions

- One room can have multiple functions such as being a sleeping room and a meeting room. However, when it's booked it can only be used for one function. That's the reason we need the `current_function` table to indicate the `roomType` of this room in the current order.

3. Design Choices:

Based on the provided requirements and our previous assumptions, we made the following design choices:

3.1 Room Organization and Status

- Created a hierarchy of hotel -> building -> wing -> room to represent the physical structure
- Implemented a separate `room_status` table to document the status of a room instead of recording it in the room table itself
- Created a separate `room_adjacency` table to keep track of various connections between rooms like moving walls
- Included a `hasPermanentBed` in room table to make sure rooms with permanent beds cannot be a meeting room

3.2 Customer and Billing

- Distinguished between different customer types (guests, hosts, billing parties)
- Created a dedicated transaction table to handle billing arrangements, allowing the people using the room to split the payment among themselves

3.3 Event and Reservation

- Separated Event and Reservation concepts, as they can exist independently
- Used bridging tables (`event_room` and `reservation_detail`) to handle many-to-many relationships

4. Team Assignments:

Kevin: So far, I have organized the information provided in the description. I have also worked on ERD creating a number of tables (`hotel`, `building`, `wing`, `room`, `room_type`, `room_status`, `reservation`, and `reservation_detail`), and connecting them with relationships. I also worked on this analysis report, recording many of our assumptions and missing pieces of information.

Iven: I have worked on ERD creating and editing tables: hotel, event, reservation, reservation_requirement, event_room, and transaction. Then I identified the cardinality among them with each other and the customer. I edited the analysis report to make sure that changes to the tables are reflected in the report.

Tim: I have worked on ERD, creating and editing tables: room, room_type, current_function, bed, bed_in_room, and room_relationship. I also edited the analysis report to make sure all the assumptions we made about the room are correctly reflected in the report.

Xiangcheng: I have worked on identifying the relationships between transactions and bills, and creating tables related to card assignment and location. I also worked on figuring out what assumptions are necessary on the requirements mentioned on the last page. The key argument we have here is whether a bill_to_transaction table is needed. I'll look more into how to realize the function of combining and splitting the transactions.