

Milestone 1 – Draft Analysis Report

Team member: Lavinia Yang, Xiaofei Yang, Sophie Osunkoya, Jebonnesa Hossain

Draft Analysis Report

1. Business Overview

Last Resort Hotels (LRH) is a luxury hotel chain that has grown rapidly through the acquisition of new properties, including Club Med and other high-end resorts worldwide. As LRH expands, each property now consists of multiple buildings, wings, floors, and hundreds of rooms with unique configurations and amenities. The company also hosts large events such as conferences, weddings, and business meetings. This expansion has introduced significant operational complexity. Managing reservations, guest services, events, and billing manually or across separate systems has become inefficient. A unified database is needed to centralize information across all hotel locations, allowing staff to access accurate, real-time data on rooms, reservations, and events.

2. Problem Statement

- Lack of centralized, integrated data across properties.
- Difficulty tracking room availability, adjacent room relationships, and ongoing renovations.
- Inconsistent reservation and billing processes for guests and organizations.
- Limited ability to connect event scheduling with room and facility usage.
- Slow or incomplete communication between departments such as the front desk, housekeeping, and event planning.
- Difficult to track availability, events, and billing (sleeping by day; meeting by time slots)

3. Objectives

- Track all hotel properties, buildings, wings, floors, and rooms.
- Define and store room details, including amenities, bed types, and capacities.
- Manage guest and organizational information in a unified structure.
- Record reservations, preferences, and room assignments.
- Support complex billing and payment structures for guests and hosts.

- Track and manage events, including facility usage and scheduling.
- Provide accurate, real-time information to staff and management to enhance guest satisfaction and operational efficiency.

4. Entity and Relationship Summary

Property and Room Structure

- **Hotel** – Represents each LRH property.
- **Building** – Each hotel consists of one or more buildings.
- **Wing** – Divides buildings into sections (smoking, poolside, accessible).
- **Floor** – Tracks floor levels and smoking designations.
- **Room** – Represents every sleeping or meeting room.
- **RoomType** – Defines room categories (sleeping, suite, meeting).

Room Configuration

- **BedType** – Lists possible bed sizes and types.
- **RoomBed** – Links rooms and bed types with quantities.
- **Amenity** – Lists all available room amenities.
- **RoomAmenity** – Links rooms with their amenities.
- **Room_Adjacency** – Tracks adjacent rooms with connecting doors.

Customers and Billing

- **Party** – General entity for all customers (individuals and organizations).
- **Guest** – Represents individual guests.
- **Organization** – Represents companies or groups.
- **Account** – Stores billing, payment method, and credit limit information.

Reservations

- **Reservation** – Records all bookings and stay information.
- **ReservationPreference** – Tracks specific guest preferences.
- **ReservationRoom** – Connects reservations to rooms.
- **GuestAssignment** – Tracks which guest occupies which room.

Events

- **Event** – Represents hosted events or meetings.
- **EventFacilityUsage** – Links events to the rooms or facilities used.

4a. Relationship Overview

Hotel → Building (one-to-many)

Building → Wing (one-to-many)

Wing → Floor (one-to-many)

Floor → Room (one-to-many)

RoomType → Room (one-to-many)

Room Configuration

Room → RoomBed (one-to-many)

BedType → RoomBed (one-to-many)

Room → RoomAmenity (one-to-many)

Amenity → RoomAmenity (one-to-many)

Room ↔ Room_Adjacency (many-to-many, self-relationship)

Party → Guest (one-to-many)

Party → Organization (one-to-many)

Party → Account (one-to-many)

Guest → Account (one-to-many, optional)

Organization → Account (one-to-many, optional)

Account → Reservation (one-to-many)

Reservation → ReservationPreference (one-to-one)

Reservation → ReservationRoom (one-to-many)

ReservationRoom → GuestAssignment (one-to-many)

Guest → GuestAssignment (one-to-many)

Room → ReservationRoom (one-to-many)

Party → Event (one-to-many)

5. Assumptions and Constraints

Assumption

- Each room belongs to one floor, and each floor belongs to one wing.
- Each event has one host (either a guest or an organization).
- Rooms cannot belong to multiple wings or floors simultaneously.
- Billing is handled through a single responsible party (an account), even if charges are shared.
- Room adjacency is handled through a separate linking table to prevent duplication.
- Reservation preferences are stored as a single optional record per reservation.
- Each service charge is tied to exactly one reservation or event.
- Event rooms can be used for other purposes once the event ends; no permanent allocation.

6. Team Assignments

- ERD design
 - Sophie: Created the base tables and initial entity structure for the property hierarchy (Hotel, Building, Wing, Floor, Room)

- Jebonnesa: Also created some tables and finalized relationships, defined cardinalities, and ensured consistency between all 21 tables.
- Data dictionary
 - Jebonnesa and Sophie will collaborate to define each table's attributes, data types, primary and foreign keys, and provide concise field descriptions.
- Sample queries
 - Emmelia and Lavinia will work together to create SQL queries that demonstrate how the database can be used to:
 - Retrieve guest and reservation details.
 - Display room availability by date and type.
 - Track event facility usage and related billing.
 - Summarize account balances for guests or organizations.
- Draft report writing
 - Emmelia: finished the overview and linked the report to our current ERD
 - Lavinia: finished the objectives and the assumption and refined the report
- Presentation prep
 - All four of us plan on prepping the presentation together

7. Next Steps / Milestone Plan

- We plan to create in SQL the database and try to insert some data into it (we plan to insert 20 data entries)
- Built more queries to test the model (implement the JOIN queries)
- Refine the report
- Mock up the website
- Give instructions to the management team