

## 1. Business Overview

Last Resort Hotels (LRH) is a global chain expanded via acquisitions (e.g., Club Med) and new luxury properties, forming multi-building complexes with wings, floors, and a diverse inventory of sleeping rooms, suites, and meeting spaces. As scale grows, the organization needs integrated, up-to-date information on rooms, reservations, events, services, and billing to serve customers effectively and manage operations.

## 2. Problem Statement

- Data silos and lack of central visibility into availability and usage across rooms and facilities.
- Complex scheduling: day-based sleeping stays vs. time-slot meeting usage, with adjacency and special room configurations.
- Flexible but auditable billing (split charges, event-linked concessions, surcharges/discounts, deposits based on customer qualifications).
- Real-time updates (housekeeping/repair) and optional location/access information (key cards, readers) to inform service.

## 3. Objectives

- Single source of truth for inventory (buildings → wings → floors → rooms, features, adjacency).
- Manage sleeping reservations (daily windows, extensions), meeting reservations (slot-based, discount/comp rules), and events (hosted multi-room activities).
- Track guests/hosts and unify billing under a party-based abstraction with itemized services.
- Record services/transactions for timely, verifiable bills and management analytics.
- Support operational workflows (housekeeping/repair status, access logs) and future UI integrations.

## 4. Entity & Relationship Summary

- Facilities: building, wing, floor, room (+bed config, features, adjacency).
- Parties & Billing: party (person/org), person, organization, billing\_account.
- Reservations/Events: sleep\_reservation ↔ sleep\_res\_room; meeting\_reservation ↔ meeting\_res\_room\_slot; event (hosted by a party).
- Pricing/Services: pricing\_rule, service\_type, service\_txn.
- Ops/Access: card, reader, access\_log, housekeeping\_log.

## 5. Assumptions & Constraints (initial)

- A room's `room_type` determines allowable uses: rooms with permanent beds cannot be assigned as meeting-only. Suites are modeled as sleeping rooms with feature flags.
- Meeting-room slot pricing: eating slots are full rate; non-eating slots are half-rate; one non-eating slot is free per paid eating slot (recorded as `discount_applied` or governed by `pricing_rule`).
- Sleeping stays are day-based (4PM–12PM standard); extensions  $\leq 2$  hours are no-charge; longer extensions produce a fixed surcharge (pricing rule).
- Each event has exactly one host party; guests attending the event are associated through their reservations.
- Billing splits for sleeping rooms are supported via multiple `service_txn` rows pointing to different `party_id` values under the same `billing_account` (or via a future `split_billing` table if needed).
- Room adjacency is symmetric; we'll enforce canonical pair storage and a view to surface bidirectional edges.
- Room numbers are unique per (wing,floor); wings are unique per building; buildings are globally named.
- Access logging is optional (privacy-respecting); enabled per property.

## 6. Team Assignments

- ERD design & constraints: All
- Data dictionary & seed data plan: Derek
- Query design (5–8 mgmt queries): Katherine
- Web wireframe + DB integration: Jun
- Draft Report authoring & GitHub packaging: Isabelle

## 7. Next Steps / Milestone Plan

1. Create schema DDL from this spec; stand up dev DB; seed with sample data ( $\geq 75$  customers,  $\geq 150$  reservations spanning  $\geq 1$  quarter).
2. Define 5–8 queries (occupancy, revenue by source, top customers, housekeeping).
3. Build web page: query result tables + one chart.