LAST RESORT HOTEL MANAGEMENT SYSTEM

By **Group H**

- Skye Xi
- Cicci Shao
- Sue Su
- Lein Wu

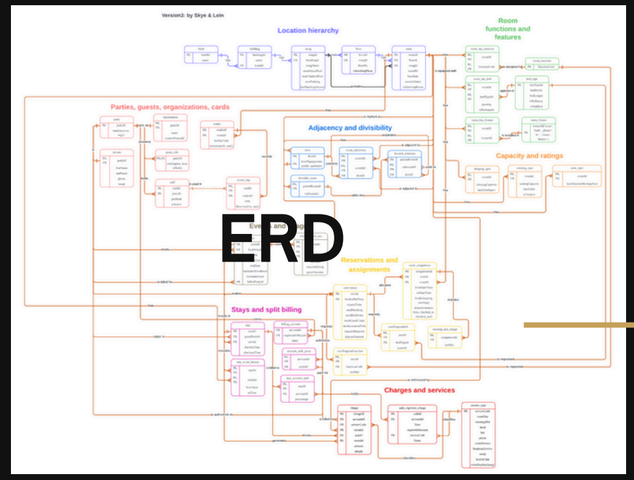
ERD

5 Logical Sections:

- Location Hierarchy
- Room Functions and Features
- Parties and Guests
- Reservations and Assignments
- Charges and Services

Lucid Chart

SCHEMA DESIGN




```sql
CREATE TABLE room (
    roomId INTEGER PRIMARY KEY,
    floorId INTEGER,
    roomNumber TEXT,
    baseRate REAL,
    currentStatus TEXT, -- Clean, Dirty, Occupied, OOO
    isSmokingRoom BOOLEAN,
    FOREIGN KEY(floorId) REFERENCES floor(floorId)
);
```

Database

1. ERD → Schema Implementation
   - Use relational databse SQLite
   - Created all tables from ERD in schema.sql
     - Users / Rooms / Reservations / etc.

SCHEMA DESIGN

```sql
-- five floors
INSERT INTO floor (floorId, wingId, floorNo) VALUES
(1,1,1), (2,1,2), (3,1,3), (4,1,4), (5,1,5),
(6,2,1), (7,2,2), (8,2,3), (9,2,4), (10,2,5);

-- room type
INSERT INTO room_function VALUES ('SLP', 'Guest Room'),
('MTG', 'Meeting Salon'), ('STE', 'Grand Suite');
INSERT INTO bed_type VALUES (1, 'King', 2), (2, 'Queen', 2),
(3, 'Twin', 1), (4, 'California King', 2);
INSERT INTO service_type VALUES ('ROOM', 'Room Charge'), ('FOOD', 'Dining'),
 ('SPA', 'Wellness'), ('EVENT', 'Banquet'), ('MISC', 'Concierge');
```

2. Schema → Populate structural datas

- populate.sql inserts:
  - hotel structure (building / wing / floor)
  - room types & fixtures
  - organizations & basic accounts
- create solid foundation for more complex data insertion.

SCHEMA DESIGN

3. python script → Populate datas
   - reservationData.py
   - Randomized:
     - id
     - date & duration
     - status
     - ~10% Booked → Cancelled
   - Generate extra charges:
     - FOOD
     - SPA
     - MISC

```python
data = pd.DataFrame({
    'partyId': np.random.randint(1, 101, size=NUM_ROWS),
    'day_offset': np.random.randint(0, max_offset_days, size=NUM_ROWS),
    'stay_days': np.random.randint(MIN_STAY, MAX_STAY + 1, size=NUM_ROWS)
})

data['startDate'] = [
    START_DATE_RANGE + timedelta(days=o) for o in data['day_offset']
]

data['endDate'] = [
    s + timedelta(days=d) for s, d in zip(data['startDate'], data['stay_days'])
]
```

randomization made by pd

```python
cur.execute("""
    INSERT INTO charge (accountId, serviceCode, amount, dateIncurred)
    SELECT
        b.accountId,
        'ROOM',
        rm.baseRate * (julianday(r.endDate) - julianday(r.startDate)),
        r.startDate
    FROM reservation r
    JOIN billing_account b ON b.partyId = r.partyId
    JOIN room rm ON rm.roomId = r.roomId
    WHERE r.roomId IS NOT NULL;
""")
```

insert into db

QUERIES

Main purposes

- support daily hotel operations
- provide trend analysis for management

HOW WE STRUCTURE QUERIES

1.Top revenue clients
2.Room utilization
3.Monthly revenue
4.Service revenue breakdown
5.Cancellation stats
6.Guest demographics
7.Average stay length
8.Peak occupancy by weekday

QUERIES

Operations Dashboard

Occupancy	Arrivals Today	Departures	Revenue Today
20.6%	4	0	$14800.0

TODAY'S ARRIVALS

RESV ID	GUEST NAME	STATUS	ROOM	ACTION
#20	**Ginny Weasley**	Due In	W302	CHECK IN
#43	**Count Dooku**	Due In	E103	CHECK IN
#59	**Count Dooku**	Due In	E305	CHECK IN
#162	**Boba Fett**	Due In	E506	CHECK IN

QUERIES

real-time room status

Rooms Management (63)

Search Room # or Wing... | All Statuses | SEARCH | RESET

ROOM #	LOCATION	TYPE	RATE	STATUS
E101	East Wing / Floor 1	Guest Room	$350.0	CLEAN
E102	East Wing / Floor 1	Guest Room	$350.0	OCCUPIED
E103	East Wing / Floor 1	Guest Room	$350.0	CLEAN
E104	East Wing / Floor 1	Guest Room	$380.0	DIRTY
E105	East Wing / Floor 1	Guest Room	$380.0	CLEAN
E106	East Wing / Floor 1	Guest Room	$400.0	CLEAN
E201	East Wing / Floor 2	Guest Room	$350.0	OCCUPIED
E202	East Wing / Floor 2	Guest Room	$350.0	CLEAN
E203	East Wing / Floor 2	Guest Room	$350.0	CLEAN
E204	East Wing / Floor 2	Guest Room	$380.0	OCCUPIED
E205	East Wing / Floor 2	Guest Room	$380.0	CLEAN
E206	East Wing / Floor 2	Guest Room	$400.0	OCCUPIED

Reservations (200)

Guest Name, ID or Org... SEARCH + NEW

RESERVATIONID	DISPLAYNAME	CHECKIN	CHECKOUT	STATUS	ACTION
#77	Tesla Motors ORG	2025-12-31	2026-01-06	Booked	CHECK IN
#12	Drax Destroyer PERSON	2025-12-30	2026-01-06	Booked	CHECK IN
#60	Rex Captain PERSON	2025-12-29	2026-01-03	Booked	CHECK IN
#114	Sirius Black PERSON	2025-12-29	2026-01-08	Booked	CHECK IN
#16	Steve Rogers PERSON	2025-12-28	2026-01-07	Booked	CHECK IN
#24	Aragorn Elessar PERSON	2025-12-28	2026-01-03	Booked	CHECK IN
#54	Gamora Zen PERSON	2025-12-27	2026-01-03	Booked	CHECK IN
#26	Albus Dumbledore PERSON	2025-12-26	2025-12-28	Booked	CHECK IN
#68	Kanan Jarrus	2025-12-26	2026-01-02	Booked	CHECK IN

Scroll to view more reservations ↕

Guests & Parties (100)

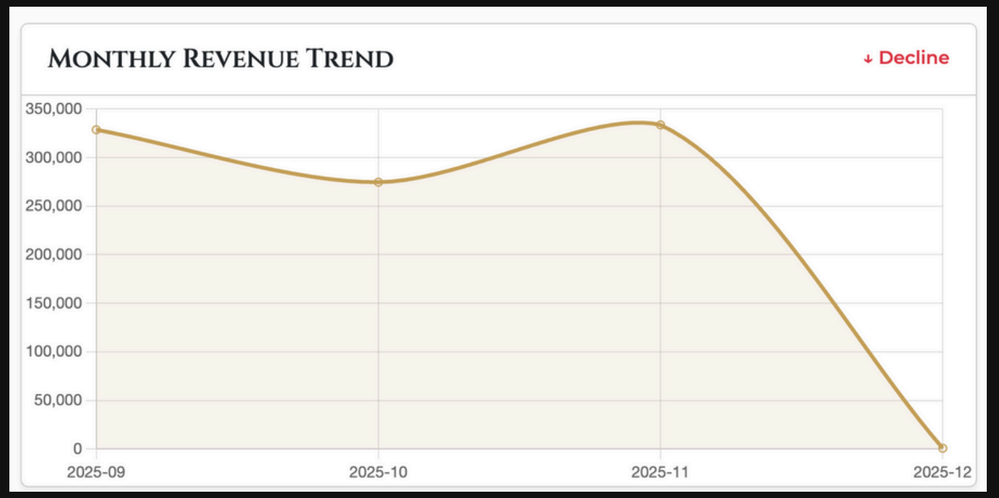
Name, Email or Org... SEARCH

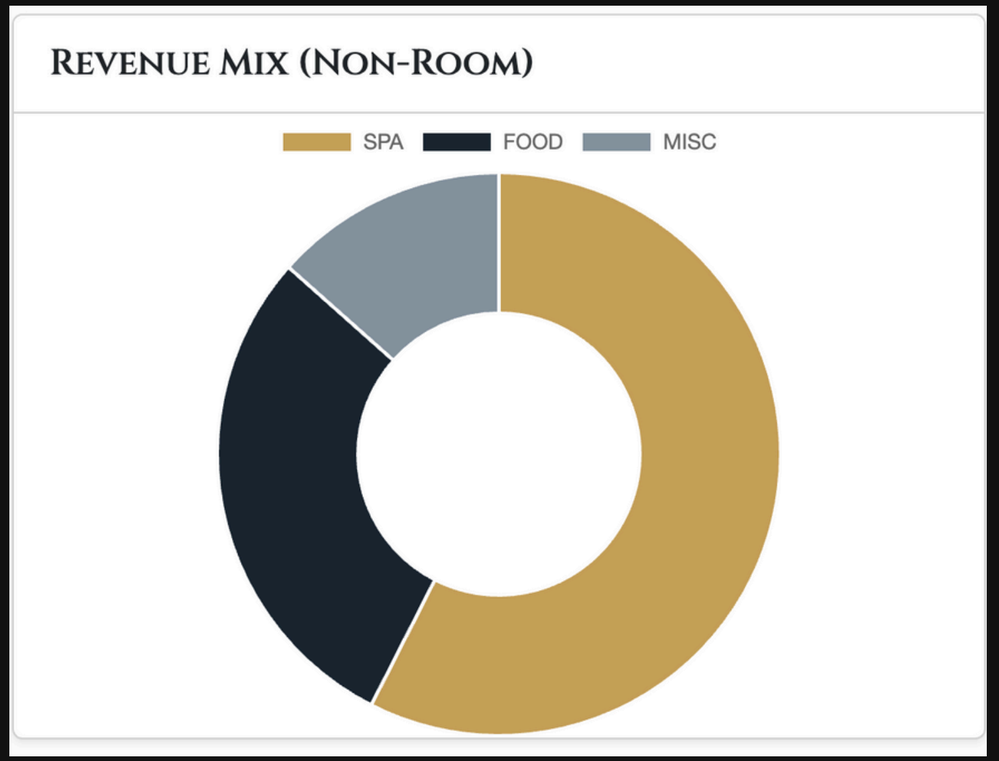
ID	NAME / ORG	TYPE	EMAIL	PHONE
1	Tesla Motors	Organization	contact@tesla.com	212-555-0101
2	LVMH Group	Organization	info@lvmh.com	212-555-0102
3	Vogue Magazine	Organization	events@vogue.com	212-555-0103
4	Harvard University	Organization	admin@harvard.edu	617-555-0104
5	James Bond	Person	james.bond@mi6.uk	007-007-0007
6	Tony Stark	Person	tony@stark.com	212-555-2384
7	Bruce Wayne	Person	bruce@wayne.com	212-555-7893
8	Peter Parker	Person	peter@parker.com	212-555-2388
9	Clark Kent	Person	clark@dailyplanet.com	212-555-3853
10	Diana Prince	Person	diana@themyscira.com	212-555-3784
11	Natasha Romanoff	Person	natasha@shield.gov	202-555-0111
12	Steve Rogers	Person	steve@shield.gov	202-555-0112
13	Wanda Maximoff	Person	wanda@avengers.com	212-555-0113
14	Vision Android	Person	vision@avengers.com	212-555-0114

Scroll to view more guests ↕

Management Reports Include

1.Top Revenue Clients
2.Room-Type Utilization
3.Monthly Revenue Trends
4.Service-Type Revenue Breakdown
5.Cancellation Rates
6.Guest Demographics (individual vs. organization)
7.Average Stay Length
8.Peak Occupancy by Weekday



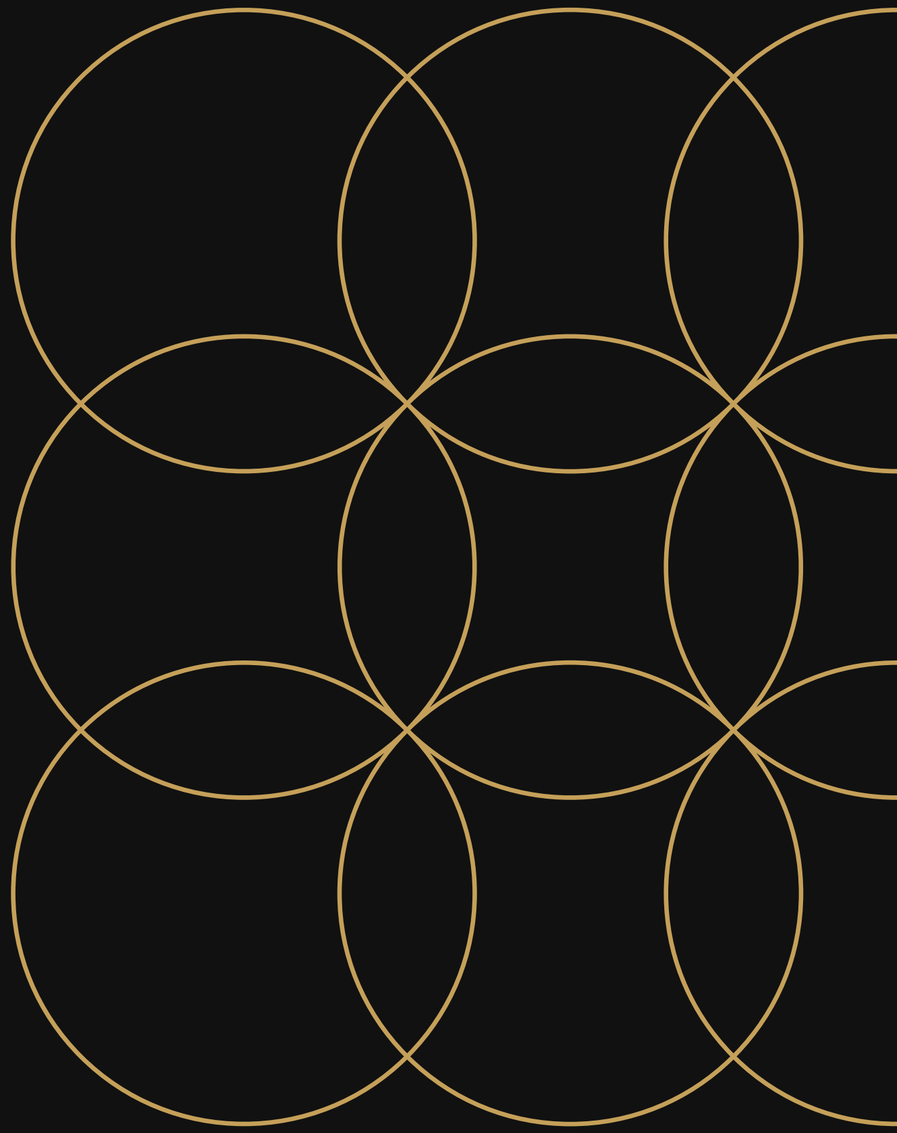


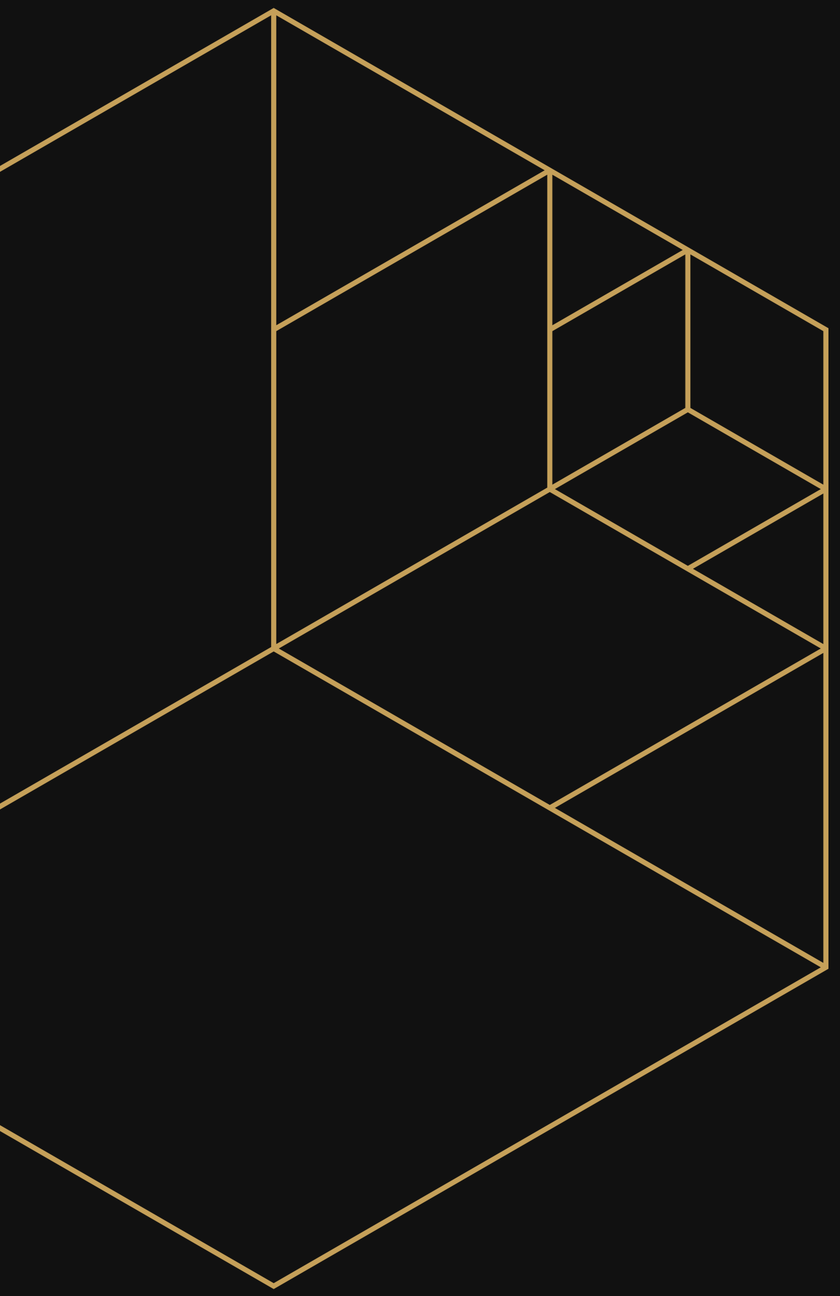
Room Utilization

Type	Total	Occupied	Rate
Grand Suite	12	3	25.(
Guest Room	48	10	20
Meeting Salon	3	0	

LIVE DEMO

Live Demo





CONCLUSION

CONTRIBUTIONS

- Skye Xi
- Cicci Shao
- Sue Su
- Lein Wu