Project Outline & Details

Team Members
Atlas Brodin (atlas.brodin@nyu.edu),
Katie Carlson (kc5180@nyu.edu),
Avril Luo (hl5629@nyu.edu),
Yanxi Chen (yc7497@nyu.edu)

Relevant Links
GitHub Repo: https://github.com/atl4sxR/Hotel_Last_Resort
LucidChart: Drafting
https://lucid.app/lucidchart/cbbf102e-99cb-4775-ab29-e671778adeb9/edit?page=0_0&invitationId=inv_abcd78d0-0b43-46ec-833b-d7597a16defe#
LucidChart ERD:

Figma:

Milestones
Project Milestone 1 - Due: Thursday 23 October

- ERD (PDF) should show all tables and relationships
- The Draft Analysis Report should show an understanding of the whole business and indicate team assignments.

Project Milestone 2 due on Tuesday, November 11th

- Proposed information for management.
- Wireframe (mockup) of website.

Final Project due on Thursday, December 4th

- Entity Relationship Diagram (ERD)
- Data Export of Database
- Analysis Report
- PyCharm Project Folder
- Team Evaluation Document (Optional)

Project Milestone 1
20 Tables in the ERD

- Hotel/Building
- Wing
- Floor
- Event
- EventRoom
- Reservation
- ReservationRoomAssignment
- Billings
- RoomType
- RoomSpace
- RoomRate
- Room
- BedType
- RenovationReconstruction
- CustomerType (Guest/Host)
- Host

- Guest
- GuestNumber

Draft Analysis Report

- Assumptions made about the ERD
- Team Contributions
  - Atlas: Guests, Data Analysis Report
  - Katie: Rooms
  - Avril: Events, Data Analysis Report
  - Yanxi: Reservations, Billings, Facilities

Room:

-Types:

-Sleeping Rooms: one or two double beds (regular, XL, queen, king)
-Rated by how many people are sleeping in it
-Must have toilet
-Smoking or non-smoking
-Suite Rooms: regular sleeping room + working room
-Amount of space
-Meeting Rooms:
-Toilet facilities or no toilet facilities
-Rated by how many people can sit around the tables
-Can become a sleeping room, only if it has toilet facilities

-Location:

-Building
-Wing
-Floor

-Room Composition/adjacency/status:

-Facilities

-Facility type: Pool, gym, etc.
-Facilities: Location/capacity
- Reader: Readers are placed in various facilities such as every meeting room, restaurant, and health club; Each card reader has two slots coming (entering) and going (leaving the facility).
- Card: Each guest receives a plastic card with a PIN. As they move about within the hotel, they have the option of running the card through readers indicating their current location.
-Card activity: the card can be used to inform the computer of their current location.

-Reservations

-Allocated from 4 pm to 12 noon the next day; allows early or late extensions for up to 2 hours
-Can be made up to 2 years in advance (or more for major functions)
- Room Schedule (schedules use of meeting rooms; Time, Date, Host)
- Smoking Preference (for rooms)

- Reservation_Detail (specific room, bed types, preferences, number of guests)
- Event Table (meeting/conference/gathering,)

-Guests

-Types:

-Guests: stay in the overnight sleeping rooms
-Can switch between rooms during the same visit
-Can split up the billing of a room as long as there is a responsible party
*Responsible Party: billed for usage of facilities and services at the hotel
-Hosts: could be an individual/organization that hosts a meeting/meal in a meeting room

- Host Table (uniqueID, room type, room preference, room proximity)
- Guest Table (paired to Host Table, includes uniqueId, host/guest, affiliation, roomNumber, roomType, charges)

-Event

- host: guest/ billed party/ both
- duration
- several facilities will be used in conjunction
- estimated attendance
- estimated number of guests

-Usage of Allied Services

- Sleeping in Room
- Meeting in Room
- Meals
- Phone
- Extra Room Charge (set up/delivery)
- Business (scanning, etc)
- in Complex Charges (retail)
- Services (masseuse, etc.)

-Billings

- Billing Account (tracks billings for guests, groups, events)
- Charge (individual charges)
- Charge_Type (room, food, drink, service, etc.)
- Payment (person charged + payment maid toward bill)

ERD Notes:

Identity / Parties / Contact

1. **Person**
    - PK: personId
    - attrs: firstName, lastName, email, phone
    - Usage: person record (guests, staff, contacts). Or separate into multiple tables?
2. **Organization**
    - PK: organizationId
    - attrs: name, mainContactPersonId → Person(personId), phone, address
    - Usage: organizations that can host or be billed.
3. **Party**
    - PK: partyId
    - attrs: partyType ('PERSON' or 'ORGANIZATION'), personId → Person(personId) NULL, organizationId → Organization(organizationId) NULL
    - Usage: any billed/host party (avoids duplicating persons/orgs). (Required; referenced by billing/event/reservation.)

Rooms & Location

4. **Building**
    - PK: buildingId
    - attrs: name, address
5. **Wing**

- PK: wingId
- attrs: buildingId → Building(buildingId), wingCode, description

6. **RoomType**
   - PK: roomTypeId
   - attrs: name, isSleeping BOOLEAN, baseCapacity, bedConfiguration, isSuite BOOLEAN
7. **Room**
   - PK: roomId
   - attrs: roomNumber (unique), wingId → Wing(wingId), floor, roomTypeId → RoomType(roomTypeId), baseRate, status, hasPermanentBeds BOOLEAN, hasToilet BOOLEAN
   - Usage: physical rooms — sleeping or meeting rooms. hasToilet used to decide convertibility.
8. **Room_Adjacency**
   - PK: adjacencyId
   - attrs: roomIda → Room(roomId), roomIdb → Room(roomId), doorBetween BOOLEAN
   - Usage: records which rooms are adjacent

Access card / activity

9. **Access_Card**
   - PK: cardId
   - attrs: issuedToPartyId → Party(partyId), issuedAt, expiresAt, status (ACTIVE, LOST, DEACTIVATED)

- Usage: guest & staff cards.

10. **Card_Reader**
    - PK: readerId
    - attrs: facilityId → Facility(facilityId) NULL (may be room), locationDescription
    - Usage: readers installed at facilities
11. **Card_Activity**
    - PK: cardActivityId
    - attrs: cardId → Access_Card(cardId), readerId → Card_Reader(readeId), swipeTime, direction (IN, OUT), linkedReservationId → Reservation(reservationId) NULL
    - Usage: access logs to infer whereabouts.

Reservation & scheduling

12. **Reservation**
    - PK: reservationId
    - attrs: leadPartyId → Party(partyId), leadGuestPersonId → Person(personId), startDate, endDate, status (PENDING, CONFIRMED, CHECKED_IN, CHECKED_OUT, CANCELLED), created_at, notes
    - Usage: header for sleeping or meeting reservation.
13. **Reservation_Detail**
    - PK: reservationDetailId
    - attrs: reservationId → Reservation(reservationId), roomTypeId → RoomType(roomTypeId), requestedBeds, numGuests, smokingPref,

proximityPref

- Usage: guest requirements (bed type, smoking, counts).

14. **Room_Assignment**
    - PK: assignmentId
    - attrs: reservationId → Reservation(reservationId), roomId → Room(roomId), assignDate, checkinDatetime, checkoutDatetime, isPrimary, extensionMinutes, surchargeApplied
    - Usage: records actual room allocations and extensions (4pm–noon rule, + up to 2-hr free extension).
15. **Timeslot**
    - PK: timeslotId
    - attrs: name (BREAKFAST, MORNING, LUNCH, AFTERNOON, SUPPER, EVENING, NIGHT), startTime, endTime, isEatingSlot BOOLEAN
    - Usage: is_eating_slot used by billing logic.
16. **Reservation_Room_Schedule** *(reservation-centric meeting usage & billing)*
    - PK: reservationScheduleId
    - attrs: reservationId → Reservation(reservationId), facilityId → Facility(facilityId) (meeting room), timeslotId → Timeslot(timeslotId), startDatetime (Separate?), endDatetime, eatingFlag, feeApplied, rateApplied
    - Usage: billing-relevant schedule entries; used to compute eating/noneating rules.
17. **Facility_Schedule** *(facility-centric master availability calendar)*
    - PK: facilityScheduleId
    - attrs: facilityId → Facility(facilityId), startDatetime, endDatetime, usageType (PUBLIC, PRIVATE, MAINTENANCE, EVENT), relatedReservationId →

Reservation(reservationId), relatedEventId → Event(eventId), notes

- Usage: global facility availability (blocks for maintenance or reservations). (Facility schedule must be updated whenever Reservation_Room_Schedule is created.)

Events

18. **Event**
    - PK: eventId
    - attrs: eventName, hostPartyId → Party(partyId), billingAccountId → Billing_Account(billingAccountId), startDatetime, endDatetime, estimatedGuestCount, notes
    - Usage: event header (host can be guest or billed party).
19. **Event_RoomUse** *(bridge Event–Room)*
    - PK: eventRoomUseId
    - attrs: eventId → Event(eventId), roomId → Room(roomId), timeslotId → Timeslot(timeslotId), date
    - Usage: many-to-many link for events using rooms.
20. **Event_FacilityUse** *(bridge Event–Facility)*
    - PK: eventFacilityUseId
    - attrs: eventId → Event(eventId), facilityId → Facility(facilityId), timeslotId → Timeslot(timeslotId), date,
    - Usage: which facilities an event uses (pool patio, restaurant, etc.).

Facility & services

21. **Facility_Type**

- PK: facilityTypeId
- attrs: name (POOL, GYM, MEETING_ROOM, RESTAURANT, OUTDOOR), description, defaultCapacity

22. **Facility**
    - PK: facilityId
    - attrs: facilityTypeId → Facility_Type(facilityTypeId), name, buildingID→ Building(buildingId), wingId → Wing(wingId), floor, capacity, hasToilet, status
    - Usage: meeting rooms, pool patio treated as facility (hasToilet used for convertibility).
23. **Service_Type**
    - PK: serviceTypeId
    - attrs: description, basePrice, chargeTypeId → Charge_Type(chargeTypeId)
    - Usage: allied services (printing…)
24. **Service_Usage**
    - PK: serviceUsageId
    - attrs: serviceTypeId → Service_Type(serviceTypeId), guestPersonId → Person(personId), billingAccountId → Billing_Account(billingAccountId), facilityId → Facility(facilityId), usageDatetime, qty, amount, performedByPersonId → Person(personId)
    - Usage: records allied services consumed and posts charges.

Billing / Charges / Payments

25. **Billing_Account**
    - PK: billingAccountId

- attrs: responsiblePartyId → Party(partyId), leadGuestPersonId → Person(personId), openDate, closeDate, status, currency
- Usage: aggregates charges & payments for a visit or event. There must be a responsible party.

26. **Billing_Account_Participant** *(was paymentParticipants)*

    - PK: participantId

    - attrs: billingAccountId → Billing_Account(billingAccountId), partyId → Party(party_id), sharePercent, isResponsible

    - Usage: split percentages for account; at least one isResponsible = TRUE must exist.

27. **Charge_Type**

    - PK: chargeTypeId

    - attrs: code (ROOM, FOOD, SERVICE, PHONE, RETAIL), description, default_taxable.

28. **Charge**

    - PK: chargeId

    - attrs: billingAccountId → Billing_Account(billingAccountId), reservationId → Reservation(reservationId), guestPersonId → Person(personId), chargeTypeId → Charge_Type(chargeTypeId), serviceUsageId → Service_Usage(serviceUsageId), facilityId → Facility(facilityId), quantity, unitPrice, taxAmount, amount, postedAt

    - Usage: every billable transaction line.

29. **Charge_Allocation**

    - PK: allocationId

    - attrs: chargeId → Charge(chargeId), participantId → Billing_Account_Participant(participantId), amount, createdAt

- Usage: splits a charge into participant shares (must reference participant for the same billing account).

30. **Invoice**
    - PK: invoiceId
    - attrs: billingAccountId → Billing_Account(billingAccountId), invoiceDate, invoiceStatus, totalAmoun, dueDate, notes
    - Usage: single invoice per billing account by default.
31. **Invoice_Line**
    - PK: invoiceLineId
    - attrs: invoiceId → Invoice(invoiceId), chargeId → Charge(chargeId), lineAmount, description
    - Usage: maps invoice lines to charges.
32. **Payment**
    - PK: paymentId
    - attrs: billingAccountId → Billing_Account(billingAccountId), invoiceId → Invoice(invoiceId), amount, paymentMethod, paidByPartyId → Party(partyId), paidAt, paymentStatus, applieAllocationId → ChargeAllocation(allocationId)
    - Usage: payments applied to accounts/invoices or specific allocations.
33. **Refund?**
    - PK: refundId
    - attrs: paymentId → Payment(paymentId), billingAccountId → Billing_Account(billingAccountId), amount, refundDate, reason
34. **Deposit** *(reservation-level deposit)*

- PK: depositId

- attrs: reservationId → Reservation(reservationId), billingAccountId → Billing_Account(billingAccountId), amount, depositDate, refundable, status, expiration_date

- Usage: records deposit requirement and ties to Payment via Reservation_Payment.

35. **Reservation_Payment?**

- PK: reservationPaymentId

- attrs: reservationId → Reservation(reservationId), paymentId → Payment(paymentId), amount, paymentDate, type, refundable

- Usage: ties payments to reservations specifically (deposits, pre-auth, final).