

Overview

This is the report for Team 1 for Project Delivery 1

Teammates:

member	email address
Susmita Biswas	sb78@njit.edu
Kranthi Gunuru	kg62@njit.edu
Shawn Cicoria	sc2443@njit.edu

Assumptions

1. Customer is using credit card only for online reservation, hence total participation in uses relationship and min-max is (1,N)
2. Not every customer writes reviews, hence partial participation of customer and min-max is (0,N)
3. A hotel can have zero reviews, thus partial participation in Writes_For relationship and min-max is (0,N)
4. ratings cannot be 0 or less

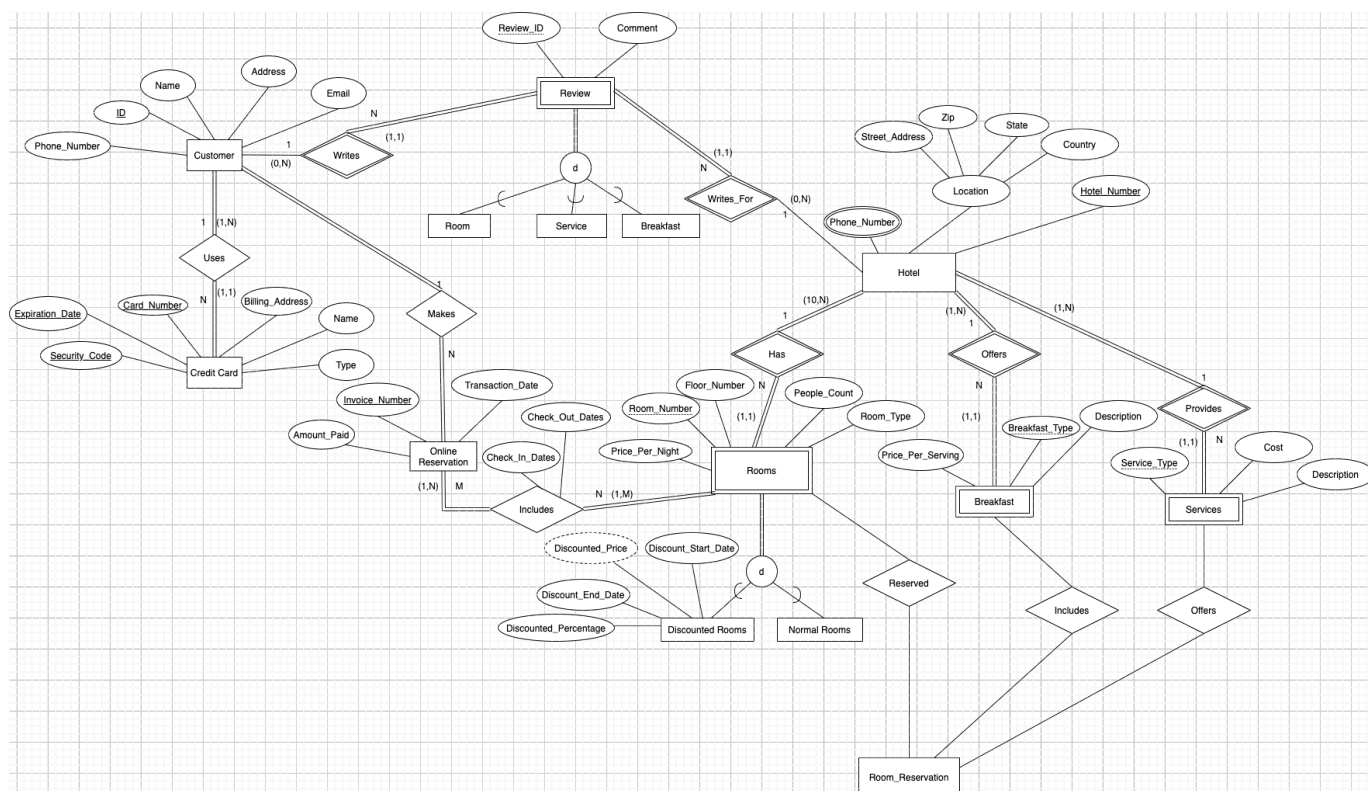
Constraints

- booking for a room cannot overlap dates for same room for any other reservation - "not allow reservations for the same room with overlapping reservation periods"
- room discounts - "every year" - either 365 between or use calendar year
- breakfast orders cannot exceed people count per room

Challenges

- how are room rates changed over time; how does this affect discounts periods.

Enhanced Entity Relationship Diagram (EERD)



Extracted Requirements

Utilizing a review of the requirements document and to provide succinct capture of requirements.

Key requirements extracted from Requirements document

- hotel
 - hotel number (key identifier)
 - phone numbers (N) (many permitted)
 - location (1)
 - street address
 - country
 - state
 - zip
 - number of rooms (rooms ≥ 10) **can use count(room)**
 - room (10:N)
 - room number (identifier at the hotel) - hotel number + room number => key
 - type of room
 - single
 - double
 - deluxe
 - suite
 - occupancy count
 - textual description (dependant upon type of room!)
 - floor number
 - price (per night)
 - room discount (0:N)
 - percent of normal price
 - start / end dates
 - year (requirement is 'only once per year' – **TODO assumption either calendar year, or every 365 days?**)
 - breakfast (meals)
 - type + hotel (key identifier) - "a breakfast offered by hotel is identified by it's type"
 - continental
 - english
 - american
 - italian

- french
 - price (per serving)
 - description
- service
 - type + hotel (key identifier) - "a service provided by a hotel is identified by it's type"
 - parking
 - laundry
 - airport dropoff / pickup
 - cost
- customer
 - ID (number? alpha?)
 - name
 - address (1) (location can be a type? and we can re-use?)
 - street address
 - country
 - state
 - zip
 - phone number
- reservation
 - customer id
 - invoice number (key)
 - reservation date
 - amount paid
 - booking (1:N) - this allows multiple rooms per reservation - "at least one room reservation in one or different hotels for same/diff ... dates".
 - hotel + room
 - checkin date
 - checkout date
 - number of days (**this can be derived using = [checkout-date - check-in date]**)
 - payment (1)
 - credit card type - (visa|mastercard|discover|amex)
 - credit card name
 - credit card number
 - expiration date
 - security code
 - billing address (location can be a type? and we can re-use?)
 - street address
 - country
 - state

- zip
 - phone number
- breakfast included (0:N)
 - breakfast + type (0:N) (hotel + type)
 - orders (0: room person count) - via "booking" within reservation
 - breakfast + type
 - number of orders (**derivable via count(orders)**)
 - **cost? derivable but does not allow change over time**
- service booking (0:N) - "**may also contain** one or more services; a room reservation can contain one or more service types"
 - type + hotel (the key of service)
- reservation cost (derivable)
 - breakfast included -> hotel:breakfast:type.cost
 - service booking -> hotel:service.cost
- review
 - review id
 - customer id - **potentially derive via reservation**
 - hotel id - **this and customer id + category** should be the "uniqueness" needed. this would allow a customer only 1 review at time for the hotel + category
 - ~~reservation id~~ - **review for which they made a reservation - not to be used**
 - booking specific???
 - category: must be the following
 - room review **only for a room and hotel:room:number**
 - service review - **only for 1 customer and hotel:service:type**
 - breakfast review - **only for 1 customer and hotel:breakfast:type**
 - rating: numeric $0 < r \leq 10$

key on customer id + category - as customer can write a review for each category

Notes
