



Product Dissection for Zomato: Food Delivery and Dining

Company Overview:

Zomato was founded by Deepinder Goyal and Pankaj Chaddah in the year 2008 providing food at your doorstep with just a click. Zomato has a user friendly interface, providing fast delivery service from multiple restaurants , it has emerged as a top leading platform in food delivery apps.

Product Dissection and Real-World Problems Solved by Zomato:

Zomato's restaurant discovery feature allows users to explore nearby restaurants based on various criteria such as location, cuisine type, price, and user ratings. Users can browse through restaurant names, view menus, photos, and read reviews to make decisions.

Zomato has a wide range of restaurants that users can choose from with a variety of cuisines that users like . In addition to that there is an option of user ratings and reviews, so whenever a user orders food , they can check which restaurant has positive reviews and can select from that thus saving them a lot of time in search of good restaurants.

Zomato offers users to track their orders in real-time, can check updates of delivery status

Conclusion: Zomato's product design has solved real-life problems by creating a platform where people can explore food, connect with others, and express their food interests.

Case Study: Real-World Problems and Zomato Innovative Solutions

Problem 1: Busy lifestyle

Real-World Challenge: As we know how fast life is and people have busy lifestyles where they don't have time to cook even regular meals, especially working people or old-age people who can't cook by themselves.

Zomato's Solution:

Zomato has provided a solution that users won't have to leave their homes, they can simply get their favorite food delivered at their doorstep with just a few clicks. Also it will save their time to go to restaurants. Even if they want to order for other people who are in a different city they can also do that easily.

Problem 2 : Food Waste Reduction

Real-World Challenge: Restaurants often face the issue of food wastage due to overproduction or unsold food stock

Zomato's Solution:

Zomato introduced the "Feeding India" initiative, a program aimed at reducing food wastage by partnering with restaurants to donate surplus food to those in need. Restaurants can use the Zomato app to identify excess food stock, which is then collected and distributed to local charities and shelters. This initiative not only helps to minimize food wastage but also supports people in need.

Problem 3: User's Preferences and limited options

Real-World Challenge: Users who have some restrictions in diet due to health issues, some users who are either vegan or eat only gluten free, they may struggle in finding restaurants as per their choices.

Zomato's Solution:

Zomato has search and filtering options that allow users to find restaurants where they can simply use a filter for vegan, gluten free, spicy, kids-choice or any other specifications as per their need as they have tied up with several restaurants. Users can also customize what they have ordered. This ensures that each and every user will have rich experience in ordering food no matter what preference they have.

Problem 4: Hygiene quality

Real-World Challenge : Users often worry about cleanliness and hygiene of restaurants they visit as they can go through diseases like diarrhea , food poisoning etc.

Zomato's Solution:

Zomato has addressed this issue, whenever a user visits any restaurant they can provide their genuine feedback and rating ,so that restaurant will keep working on their issues which will retain their customers .Also it provides restaurant's inspection reports so that users can have detailed information about it.

Conclusion : Zomato's innovative solutions address real-world challenges faced by users in the dining industry. By providing a platform for discovering top rated restaurants, ordering food, booking table reservations, zomato has enhanced the overall dining experience for users globally.

Top Features of Zomato:

Ø **Multi-City Coverage:** Zomato is available in many cities, so no matter where you are, you can use it to find good places to eat. It's like having a food guide. If you're traveling in a new city and looking for restaurants there,you can easily find it on zomato in one go.

Ø **Order Customization:** Zomato allows users to customize their food orders based on their preferences and requirements. Users can add special instructions,request modifications to ensure that they receive orders according to their need.

Ø **Table Reservations:** In addition to food delivery, Zomato also enables users to make table reservations at restaurants .Users can book tables for dining in advance.

Ø **Real-Time Updates:** Zomato provides real-time updates and notifications to keep users informed about the status of their orders, promotions, and other relevant information. Users receive alerts about order confirmations, delivery updates,including opening hours, current wait times, and table availability for reservations.

Ø **Discounts:** Zomato features deals, discounts, and special offers allowing users to save money on their orders. Users can take advantage of these promotions to enjoy their favorite food at discounted prices.

Schema Description:

The schema for Zomato involves multiple entities that represent different aspects of the platform.These entities include user, restaurant,reviews,order,payment, menu. Each

entity has specific attributes that describe its properties and relationships with other entities.

User Entity:

The user entity contains information about each user:

- **UserID (Primary Key):** A unique identifier for each user.
- **Username:** The chosen username for the user's account.
- **Email:** The user's email address for account-related communication.
- **Password:** The password for the account.
- **Phone_number:** Phone number of user
- **Address:** The complete address of user for delivery

Restaurant Entity:

Restaurants are the core of Zomato as they are key partners for delivery and dining.

- **Rest_id(Primary Key):** A unique identifier for each restaurant.
- **Rest_name :** The name of the restaurant
- **Rest_address:** Address of the restaurant.
- **rest_number:** The phone number of restaurant for communication
- **Delivery_time:** The expected delivery time given by restaurant to user
- **Rating:** The rating that each restaurant gets by user.

Orders Entity:

Orders entity represents the request of order placed by users:

- **order_id(Primary Key):** A unique identifier for each order.
- **rest_id(Foreign Key referencing Restaurant Entity):** The restaurant associated with order.
- **user_id(Foreign Key referencing User Entity):** The user who placed the order
- **Amount:** The total price of order
- **order_status:** The status of order
- **Date_placed :** The date when order is placed

Payment Entity:

Payment entity represents the transaction details of order placed by users

- **payment_id(Primary Key):** A unique identifier for each payment.

- **payment_method**: The mode through which payment is done like card ,paytm etc
- **order_id(Foreign Key referencing order Entity)**: The order associated with payment
- **bill**: The total price paid for order
- **payment_status**: The status of payment like paid,cod .

Delivery Entity:

Delivery entity ensures each order is delivered or not to the users:

- **delivery_id(Primary Key)**: A unique identifier for each delivery.
- **order_id(Foreign Key referencing orders Entity)**: The order associated with delivery.
- **delivery_status**: The status of delivery (out for delivery, delivered, canceled)
- **expected_delivery_time**: The estimated time for delivery to be completed.

Item Entity:

Item entity represents the items ordered by user

- **item_id(Primary Key)**: A unique identifier for each menu item.
- **Rest_id : (Foreign Key referencing Restaurant Entity)**
- **order_id(Foreign Key referencing Order Entity)**: The order
- **name(Foreign Key referencing User Entity)**: The name of item being ordered
- **quantity**: The total number of items ordered
- **price**: The price of item

Review Entity:

Review entity represents the rating and reviews given by user for restaurants:

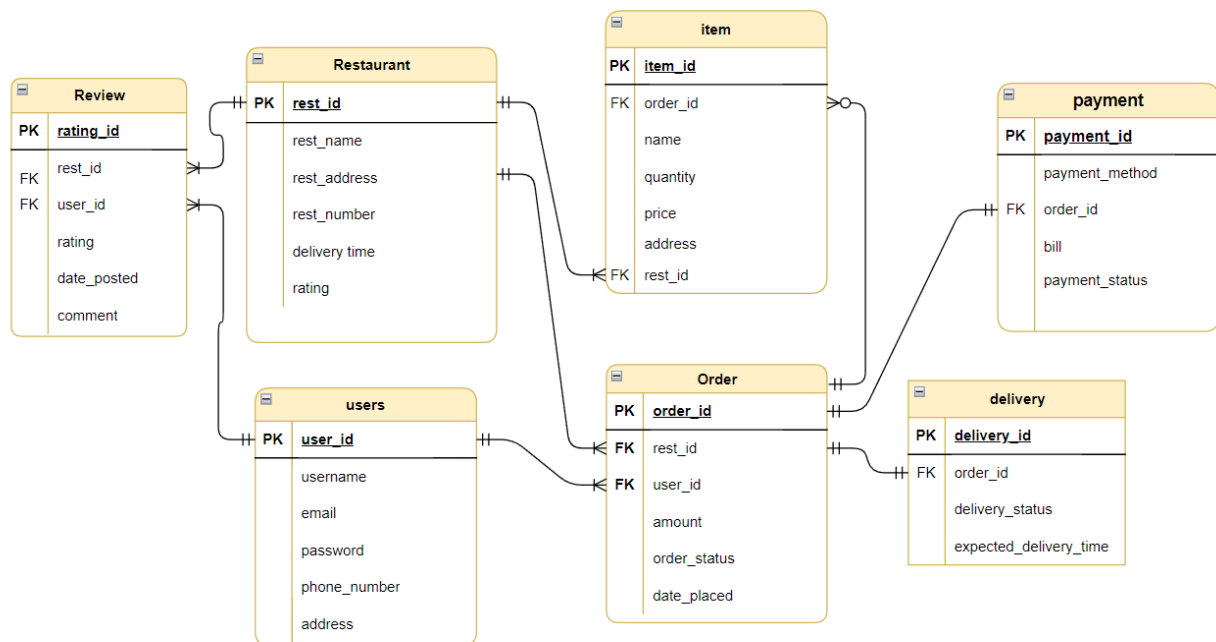
- **rating_id(Primary Key)**: A unique identifier for each rating.
- **user_id(Foreign Key referencing user Entity)**: The user who posted the review
- **rest_id(Foreign Key referencing RestaurantEntity)**: The restaurant being reviewed
- **rating**: The rating given by user on restaurant
- **date_posted**: The date on which user commented review
- **comment**:The review by user on restaurant

Relationships are:

- ∅ **Users place orders**– Each user can place multiple orders and each order is placed by single user
- ∅ **Restaurants receive orders** - Each restaurant can receive multiple orders and each order is placed at a single restaurant.
- ∅ **Payment on order**– Each order is associated with one payment and each payment with one order
- ∅ **Delivery on order**– Each order is linked to one delivery and each delivery with one order
- ∅ **Restaurant have review**– Each restaurant can have multiple reviews by user
- ∅ **Restaurant have items**– A restaurant can have multiple items, and each item belongs to single restaurant
- ∅ **Order having items** - Each order can have multiple item and each item belongs to one order
- ∅ **User post reviews** -Each user can write multiple reviews and each review is associated with one user

ER Diagram:

Let's create an ER diagram that visually represents the relationships and attributes of the entities within the Zomato schema. This diagram will provide a clear understanding of how different parts of Zomato's system are connected and interact with each other.



Conclusion:

In this case study, we've looked at how Zomato's database and diagram are set up. Zomato is like a digital hub where you can find restaurants, order food, and connect with others. Think of it like Instagram, but for food. Zomato keeps track of users, restaurants, reviews, orders, payments, and menus. Zomato's smart technology has helped it become a big player in the food delivery world.