We have applied a very minimalistic approach, where we have tried our best to provide the customer with as much information about our restaurant in the most convenient to the customer. We have mapped our entire menu as an entity which displays every food and drink we have it along with the Todays special. We have also made use of the ISA hierarchy to divide the menu into vegetarian and non-vegetarian to save time and effort. One alternative to this could also have been to ask the user to enter the dish that they want and display whether our restaurant has that dish or not, however we believe that the first approach is more convenient to the user and also more informative.

We also have provided customers with the facility to book a table in the restaurant to ensure that he/she does not need to wait on arrival. The customer on the other hand can also choose to eat at home and choose to get the food delivered to him at home. We Use an ISA hierarchy to show that a dine-in customer is also a customer whereas someone who orders online is also a customer.

After eating at our restaurant the customers also have the choice to give some ratings about the restaurant. The customer can choose to give rating through his/her customer id through which the readers reading the reviews can read his name or as a complete stranger i.e. if he/she does not want to expose himself/herself. We have used another ISA hierarchy to show that a review that came from both sources is valuable reviews for the restaurant.

We have also chosen to display information about the delivery staff that is tasked with the delivery of your food, given you choose to order food to your home.

All in all we have tried our level best to make this project as informative and complete yet easy to use at the same time.

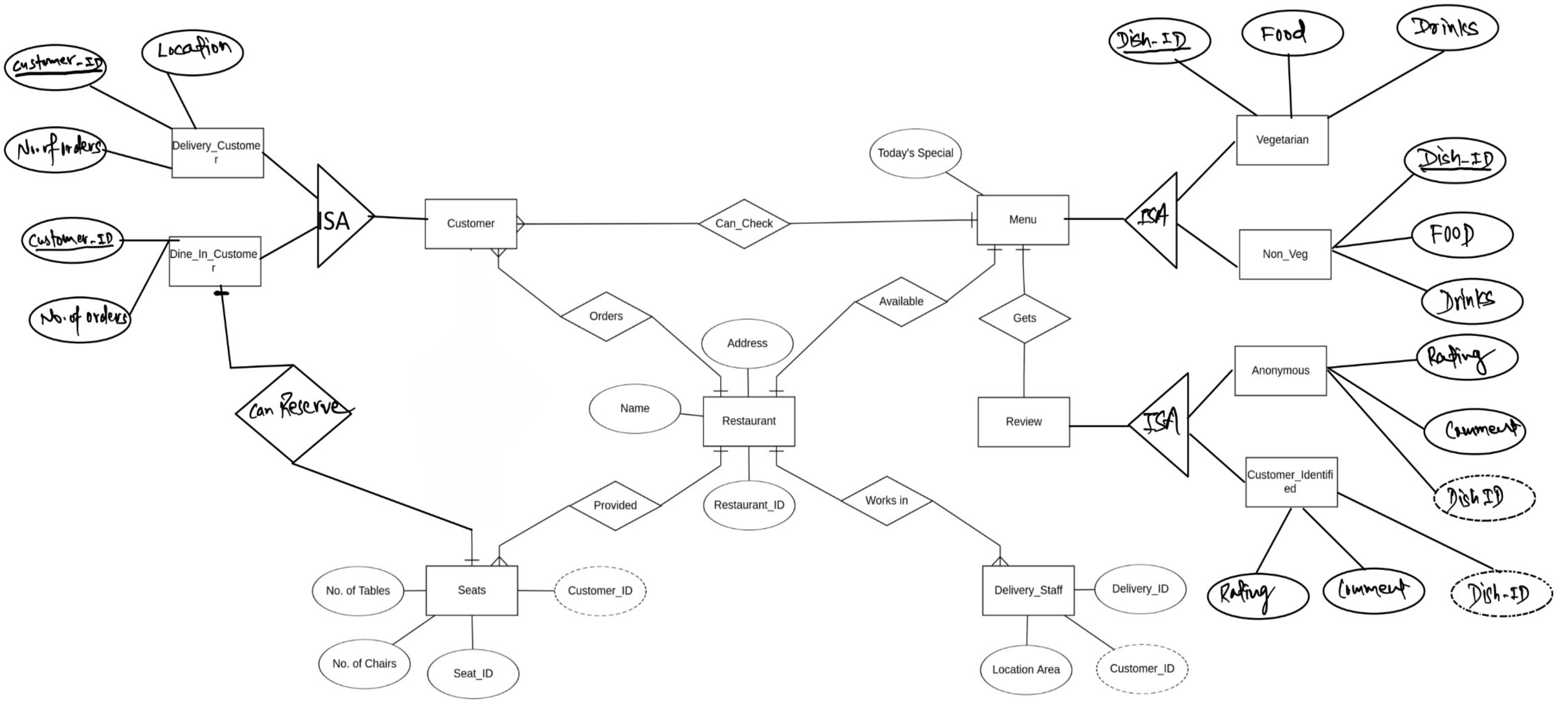

Fig: ER Diagram

```
CREATE TABLE Resturant(
Name VARCHAR(255) NOT NULL,
Address VARCHAR(255) NOT NULL,
Resturantid INTEGER NOT NULL,
PRIMARY KEY(resturantid)
);

CREATE TABLE Customer(
CustomerID INTEGER NOT NULL,
Orders INTEGER NOT NULL,
Location VARCHAR(255) NOT NULL,
PRIMARY KEY (CustomerID)
);

CREATE TABLE Seats(
TableID INTEGER NOT NULL,
CustomerID INTEGER NOT NULL,
Availabletables INTEGER NOT NULL,
Availableseats INTEGER NOT NULL,
PRIMARY KEY (TableID),
FOREIGN KEY (CustomerID) REFERENCES customer(CustomerID)
);

CREATE TABLE DeliveryStaff(
DeliverystaffID INTEGER NOT NULL,
CustomerID INTEGER NOT NULL,
Location VARCHAR(255) NOT NULL,
PRIMARY KEY(DeliverystaffID),
FOREIGN KEY (CustomerID) REFERENCES customer (CustomerID)

);

CREATE TABLE Reviews (
CustomerID INTEGER NOT NULL,
Dishid INTEGER NOT NULL,
Comment VARCHAR (255),
Rating INTEGER,
FOREIGN KEY (CustomerID),
FOREIGN KEY (Dishid) REFERENCES Menu(Dishid)
);

CREATE TABLE Menu(
Dishid INTEGER NOT NULL,
Food VARCHAR(255) NOT NULL,
Drinks VARCHAR(255) NOT NULL,
Todaysspecial VARCHAR(255) NOT NULL
);
```