

Group-7

Project 1- C5332

1. Stephen Paul
 2. Janki Patel
 3. Sai Hemanth Shishir Reddy
- 

Requirement Gathering :

- First, we referred to the project presentation and the PPT given by the professor to know the whole functionality of the application.
- We read the document and kept note of all the important entities and attributes necessary for the project functionality.
- To get a better understanding , we visited below sites which are similar to our project .

https://www.yelp.com/user_details?userid=XE_RxIRKDBYPObakRNUpHA

<https://www.angieslist.com/>

- By going through the above sites in detail, we could easily identify the main Entities and all the possible attributes.
- We discussed with each other and shared our ideas and finalized the project requirement.

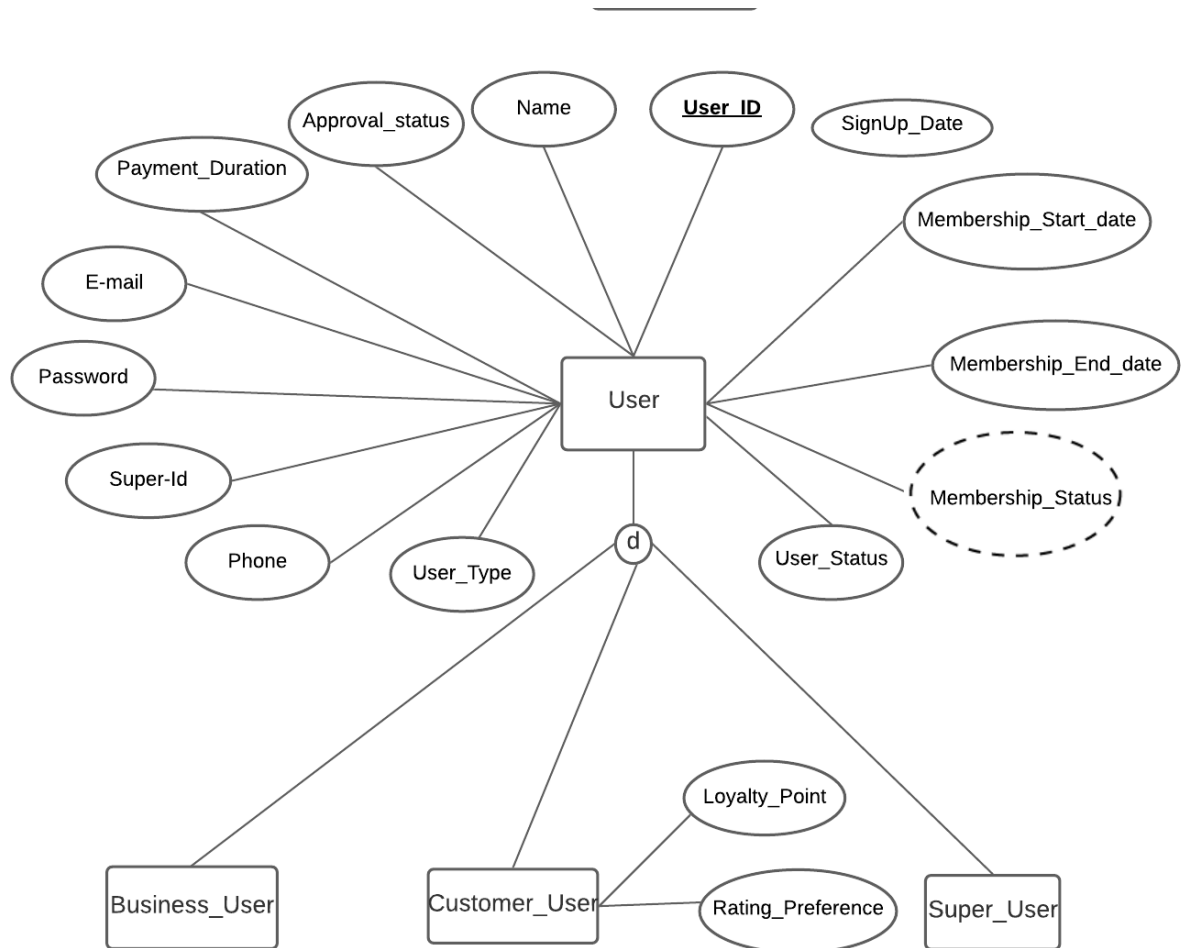
Listing of business rules/constraints:

1. There is one online service application named "Who Does That?".
2. There are three types of users: Business-user, End User(Customer) and Super-User(WDT Staff).
3. The user ID must be unique for every user.
4. Both businesses and users must first sign up in order to use Who Does That?.
5. Both users/customers and business have to choose one of the membership of Who Goes That.
6. Super user of Who Does That App, must have access to both Business and User accounts.
7. User/Business must not start using a system before his account has been approved.
8. Membership features must be modifiable.
9. Membership detail is private information for business.
10. Payment information is compulsory to all users at the time of sign-up for the membership-payment and maintaining the membership.
11. Each feature must have a unique ID.
12. A Business user can create an account only if it does not exist.
13. Each Business user must have access to reviewer information.
14. The limitation of review access depends on membership-plan.
15. The access of reviews must be sorted by time from latest to oldest.
16. One business user is the owner of only one business.
17. All Business categories and specialities must have Unique ID and must not be changed.
18. All predefined categories must reside in the database as more can be added at any point and must not require a code change.

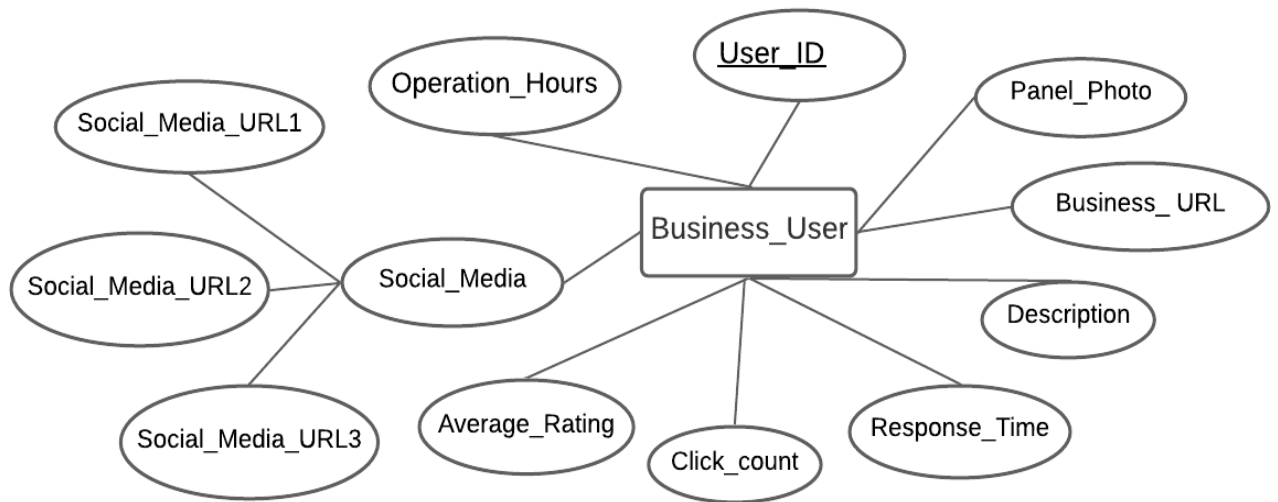
19. There can be one or more specialities in one business category.
20. There can be one or more locations to which the business provides services.
21. All Premium Customers must have access of other reviewers to clarify Reviews.
22. The business must be able to respond privately to the customer for review discussion.
23. The premium Business must have access of User information for marketing.
24. Each customer can have many coverage areas.
25. More than one business can provide service in the same area.
26. Each business will have an In-app Calendar derived from schedules of services and appointments.
27. A current job must be accessible by the customer and the Business employee who is handling it.
28. Only The Associated Business will have access to change the particular job status.
29. The search result must be in map view and list view.
30. Businesses can be listed in various orders: alphabetically, by rating, closest first, etc. with Premium- Level businesses being listed first.
31. The map View must include the Employee availability of the business.
32. A search result shown on a map must be relative to the user's current location.
33. Long-Clicking on the Business Information panel will prompt the user to add the Business to their list of Favourite Businesses.
34. The list view of search must be filterable based on search criteria.
35. Each Customer has more than one Favourite Businesses.
36. In search, if the category is omitted, the speciality must be ignored.
37. The End-user will have full control of their account
38. Premium level business must be listed first in the search list.

E-R Design of Each Entity:

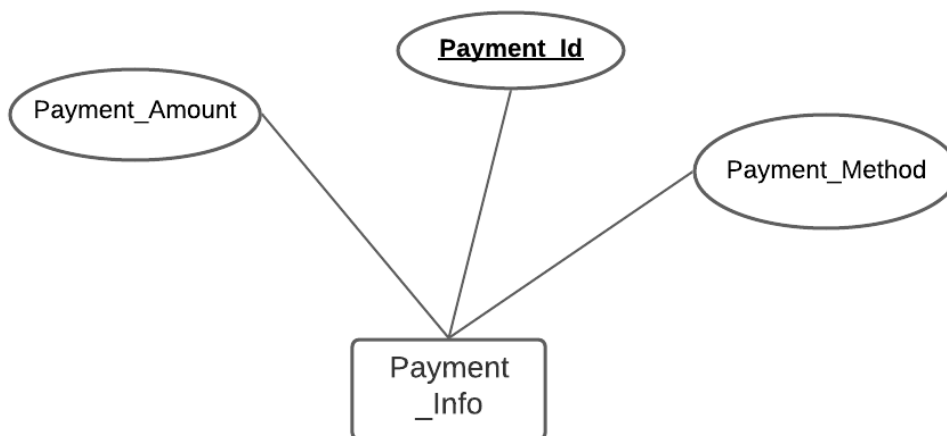
1. User:



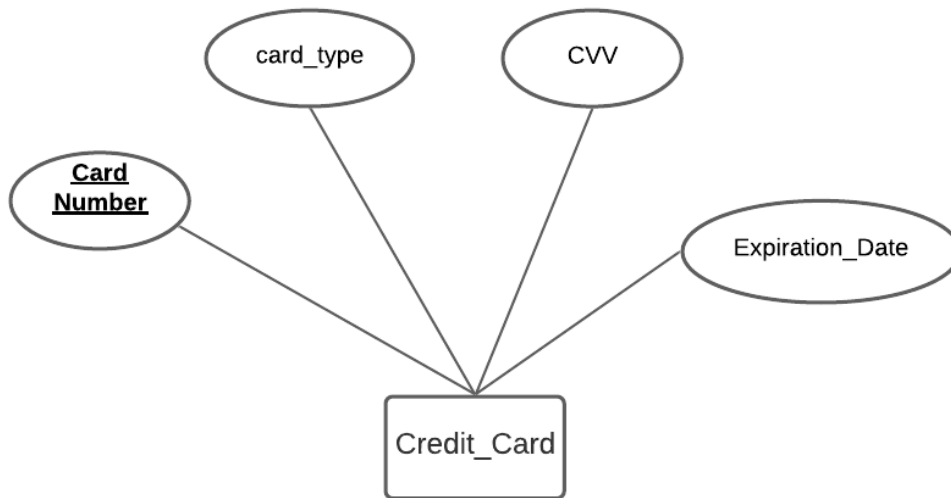
2. Business User:



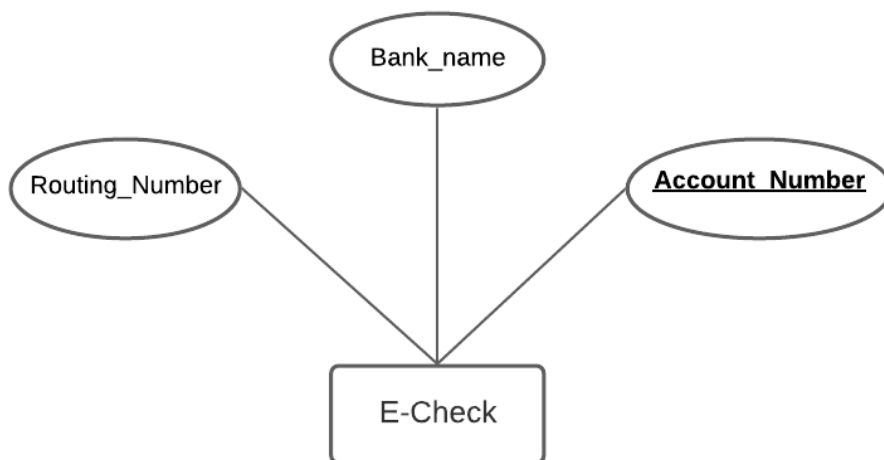
3. Payment Information:



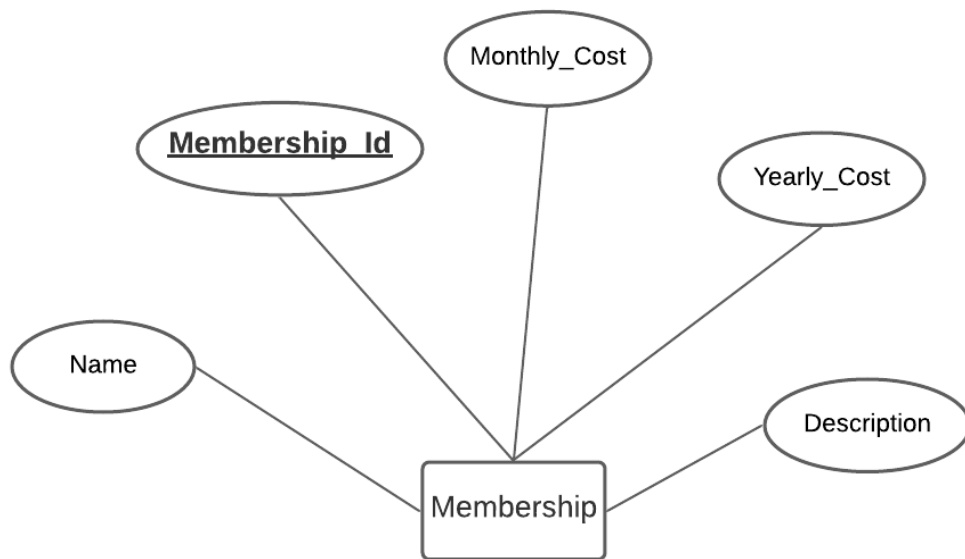
4.Credit Card:



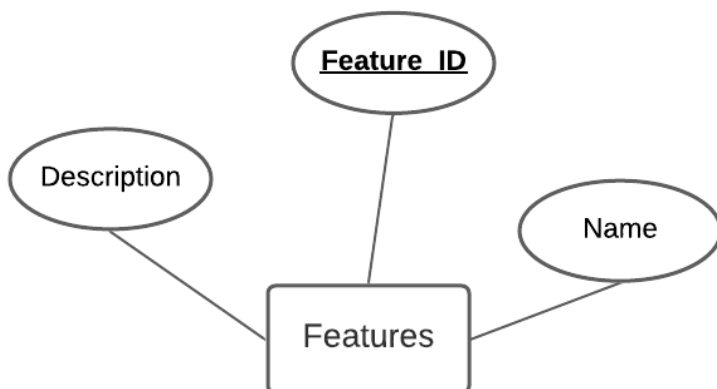
5.E-check:



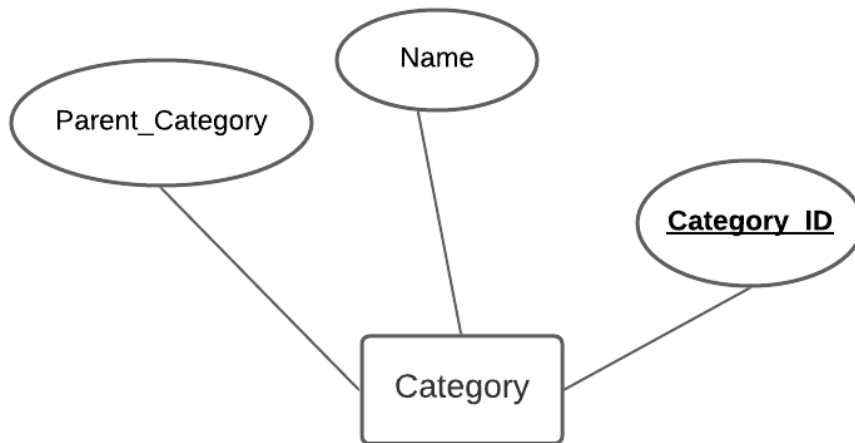
6.Membership:



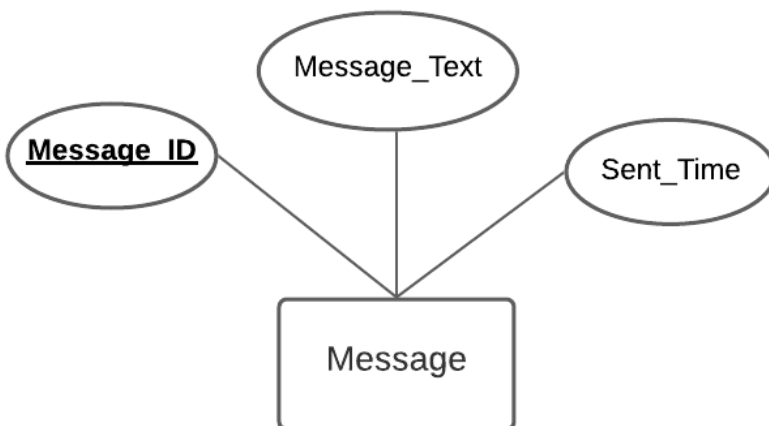
7.Feature:



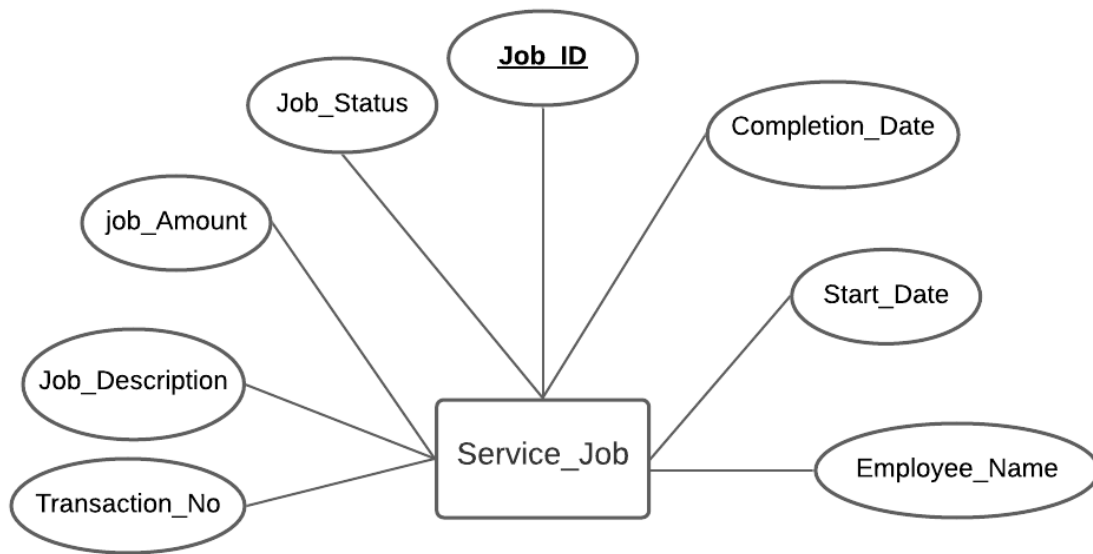
8.Category:



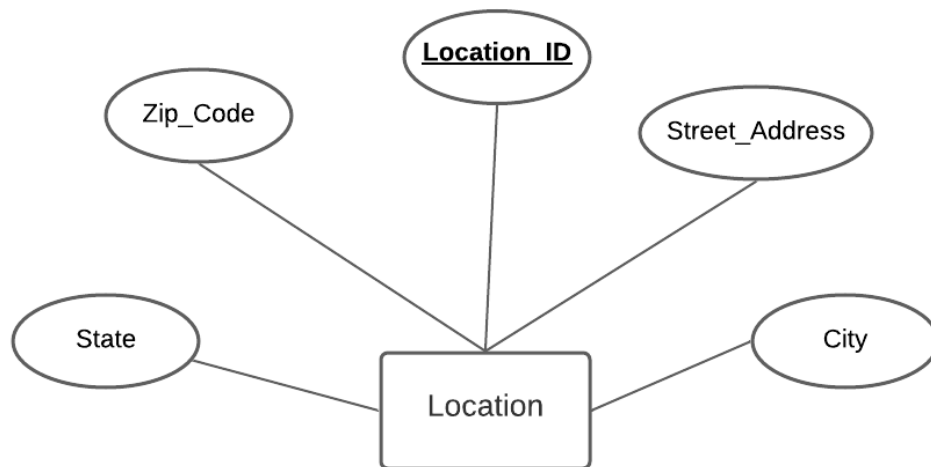
9.Message:



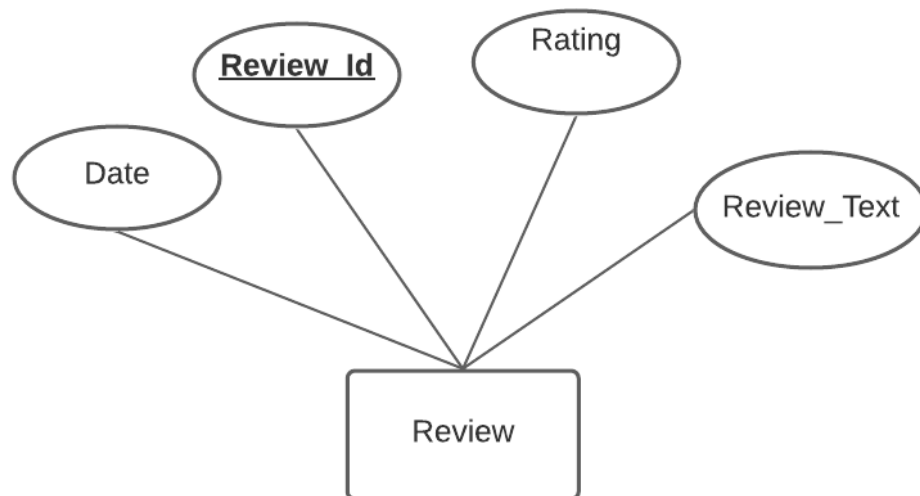
10.Job:



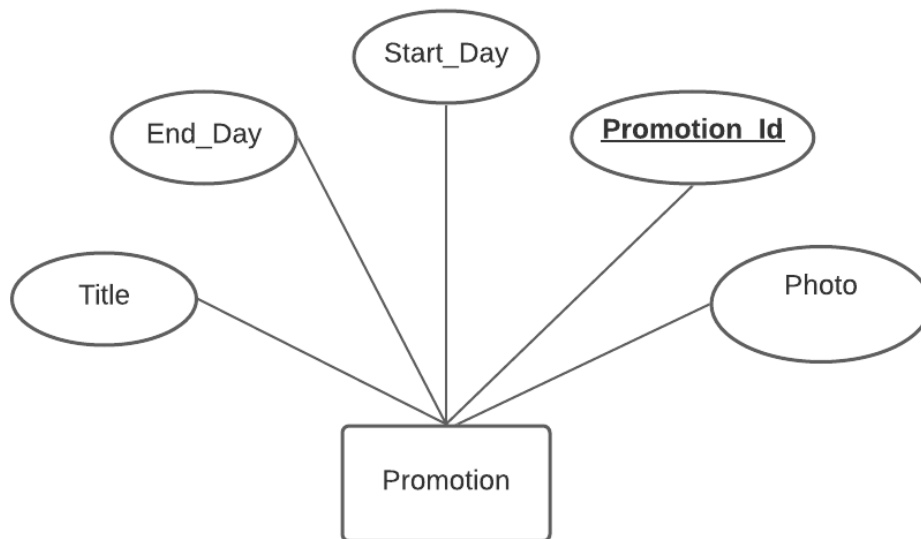
11.Location:



12.Review:



13.Promotion:



Domains of all attributes

User Types :

Int	User_Id	
String	Name	
String	Email_ID	
String	Password	
Int	Location_Id	//Foreign key for Location Entity
Int	Membership_Id	//Foreign key for Membership Entity
String	SignUp_Date	//Account creation date mm-dd-yyyy
Boolean	Approval_status	//To know account is approved or not
String	Membership_start_Date	//Membership Start date mm-dd-yyyy
String	Membership_End_Date	//Membership Start date mm-dd-yyyy
String	Payment_Duration	//Payment Due date for next membership

Int	Super_Id	//Id of super user who approved
String	Phone	
String	User_Type	//Indicates Customer, Business, Super user
String	Business_URL	//URL that contains Business details
BLOB	Panel Photo	//Storefront image of business
String	Description	//Information about business <500 chars
String	Operating_Hours	//Operating hours and days of business
String	Social_Media_URL1	//URL of business social media account
String	Social_Media_URL2	
String	Social_Media_URL3	
Float	Average_Rating	//Represents the average rating of business
Int	Click_Count	//Total count of URL clicks
Float	Response_Time	//Time business takes to respond to customer
Int	Loyalty_Point	//Loyalty points customer has earned
Float	Rating_Preference	//To set the preference for customer

Membership Types :

Int	Membership_Id	
String	Name	
Float	Monthly_Cost	//Monthly cost of membership plan in dollars
Float	Yearly_Cost	//Yearly cost of membership plan in dollars
String	Description	//Details about the membership

Feature Types :

Int	Feature_Id	
String	Name	
String	Description	//Information about features

Membership_Feature Types :

Int	Membership_Id	//Foreign key of membership entity
Int	Feature_Id	//Foreign key of feature entity

Payment_Info Types :

Int	Payment_Id	
Int	User_Id	//Foreign key of User entity
Int	Payment_Amount	//Transaction amount
String	Payment_Method	//indicates chosen payment method

Credit Card Types :

Int	Card Number	
String	Card Type	//Indicates type of card
Int	CVV	
String	Expiration_Date	
Int	Payment_Id	//Foreign key of Payment_info entity

E-Check Types :

Int	Account_Number	
String	Bank_Name	
Int	Routing_Number	
Int	Payment_Id	//Foreign key of payment info entity

Category Types :

Int	Category_Id	
String	Name	
Int	Parent_Category	//Foreign key of Category entity

Business_Category Types :

Int	Business_Id	//Foreign key of User entity
Int	Category_Id	//Foreign key of Category entity

Review Types :

int	Review_Id	
Int	Customer_Id	//Foreign key of User entity

Int	Business_Id	//Foreign key of User entity
TIMESTAMP	Date	//Timestamp of review posted
Float	Rating	//Rating from 1 to 5 stars
String	Review_Text	

Service/Job Types :

Int	Job_Id	
Int	Customer_Id	//Foreign key of User entity
Int	Business_Id	//Foreign key of User entity
String	Employee_Name	//Employee assigned for the service/request
String	Job_Status	//Indicates various stages of Job progress
Float	Job_Amount	//Indicates cost in dollars for the service requested
String	Job_Description	//provide information about service request
Int	Category_Id	//Foreign key of category entity
TIMESTAMP	Start_Date	//Start Timestamp of job
TIMESTAMP	Completion_Date	//Completion Timestamp of job

Message Types :

Int	Message_Id	
String	Message_text	//Keep record of text message
TimeStamp	Sent_time	//Defines timestamp of message
Int	Sender_Id	//User Id of the Sender
Int	Receiver_Id	//User Id of Recipient

Connection Types :

Int	Business_Id	//Foreign key of User entity
Int	Customer_Id	//Foreign key of User entity

Favourite Types :

Int	Business_Id	//Foreign key of User entity
Int	Customer_Id	//Foreign key of User entity

Appointment Types :

Int	Appointment_Id	
Int	Busniess_Id	//Foreign key of User entity
Int	Customer_Id	//Foreign key of User entity
String	Date	

Promotion Types :

Int	Promotion_Id	
Int	Business_Id	//Foreign key of User entity
String	Title	
BLOB	Photo	//Indicates promotional banner
String	Start_Date	//Represents start date mm-dd-yyyy
String	End_Date	//Represents end date mm-dd-yyyy

Location Types :

Int	LocationId	
String	Street_Address	
String	City	//To define which city
String	State	
Int	Zip Code	//Represents Zip code
Int	Business_Id	// Foreign Key of Business Entity
Int	Customer_Id	//Foreign Key of Customer Entity

Entity : User	
Primary Key : User_Id	
Attributes	Description
String Name [< 500 chars]	To keep record of personal information of each user (Business,Customer,Super)

String Email_ID [< 150 chars, xyz@gmail.com]	
String Password [< 250 chars]	
String SignUp_Date [< 15 chars 01/02/2021]	To keep the record of account initial created date.
Boolean Approval_status [1 is Approved, 0 is Denied]	To know the status of account created if it is approved or denied by Super User
String Membership_start_Date [< 15 chars 01/02/2021]	To Define the start date of the membership.
String Membership_End_Date [< 15 chars 01/02/2021]	To Define the completion date of the membership.
String Payment_Duration [Monthly, Yearly]	To define the duration of the users payment for membership renewal.
Int Super_Id	To know the user Id of the super user who approved or rejected the account.
String Phone [< 15 chars, (123)-456-7890]	To keep record of phone number of individual user.
String user_Type [< 50 chars]	To define the type of user among business, customer and super user.
Business_User->String Business_URL [< 500 chars]	URL that redirects customer users to panel that contains Business details
Business_User->Binary Large Object Panel photo [<1KB]	Defines image of storefrontTo redirect the user to a business page which contains detailed information
Business_User->String Description [< 500 chars]	A detailed description of what the business is/does and information to sell itself to customers.

Business_User->String Operating_Hours [< 100 chars Mon-Fri,7 a.m - 8 p.m]	To know the Functional hours of business for each day of the week.
Business_User->String Social_Media_URL1 [< 500 chars, Typical URL]	Stores an URL which redirects to business social media pages.
Business_User->String Social_Media_URL2 [< 500 chars, Typical URL]	
Business_User->String Social_Media_URL3 [< 500 chars, Typical URL]	
Business_User->Float Average_Rating [Rating on 1 to 5 (can have half stars)]	To define the average rating business received based on user ratings
Business_User->Int Click_Count	To define the number of users who searched and clicked for business
Business_User->Float Response_Time [Units are days]	To define the duration required in order to respond to services or requests.
Customer_User->Int Loyalty_Point	This is an attribute of Customer entity and it defines the overall loyalty points
Customer_User->Float Rating_Preference	This is also an attribute of Customer and it defines the minimum rating preference for his searches.

- We generalized User Entity using overlapping concept
- User may belong to only one subclass(Business/Customer/Super User).
- A User can **subscribe_to only one** Membership, but a membership could have multiple users. [One to Many]
- A User only has one payment information, and a payment information entity can only belong to one user. (**One to one relationship**)
- A User must have a payment method upon signing up for the app , hence they are in **total participation** with payment information.

User must have a single membership. So, User is in **Total participation**. But sometimes the membership may not contain any users at all. So membership is in **Partial Participation**.

- Business must have at least one Category. So Business is in **Total Participation**. But under category there can be any number of business or none. So, it is in **Partial Participation**.
- In our ER diagram, Business **can have multiple** categories[**Many to Many**] relationship.(each business has one or more categories and each category can be of one or more businesses).
- Business **buys** Promotion, but a promotion only belongs to one business. [One to Many]

Business is in **Partial participation** as business can either have zero or one promotion and all promotions are bought by a business. So Promotion is in **Total Participation**.

- Each Business and Customer user has at least one Location associated with them. So User is in **Total participation**.Locations will not be listed unless they are associated with a business_user or customer_user, hence they are in **total participation** with the user.
- Customers can have many **favourite** businesses and a business can be a favourite of many customers. [Many to Many]
- Customers can schedule appointments with multiple businesses and a businesses can have multiple appointments with many customers. Hence, there is a **many to many** relationship.

Customers can have any number of businesses or none as favourites and each business can be part of many or zero customers' favourite list. Therefore both are in **Partial participation**.

Customers may have no appointments scheduled with businesses, hence customer is in partial participation.

Businesses may not have any appointments scheduled with customers hence they are in partial participation.

- Customer can **request many** Service_Jobs [One to Many] and Business **Provides many** Service_Jobs to Customers which is also a One to Many relation.
- User's can interact with each other i.e, can **send** and **receive** messages [One to Many] based on membership users enroll.
- Customer **writes** Review [One to Many] and the business is **evaluated** by Review. [One to Many]

Customers can write at most one review for the service. So, Customer is in **Partial participation**. But review is in **Total participation** as review written should belong to a customer.

- User **has** Payment Information. [One to Many]

Users must have at least one payment information as the user makes payment for membership, so user is in **Total Participation** and Payment Information is also in **Total Participation**.

Entity : Membership	
Primary Key : Membership_Id	
Attributes	Description
String Name [< 500 chars]	To define the name of a membership plan.
Float Monthly_Cost [Units in Dollars]	To keep the monthly cost of membership
Float Yearly_Cost [Units in Dollars]	To keep the Yearly cost of membership
String Description [< 1000 chars]	To define the basic information and details of the membership
<ul style="list-style-type: none"> • Membership has many features and features can be in many memberships . [Many to Many] 	

Membership must have at least one feature. So, Membership is in **Total participation**. But some Features could not be present in any membership. So feature is in **Partial Participation**.

Entity : Feature	
Primary Key : Feature_Id	
Attributes	Description
String Name [< 500 chars]	The Name of the feature
String Description [< 1000 chars]	To define the basic information and details of the Feature.
<ul style="list-style-type: none"> • We have taken feature entity to represent the Membership Features. • As per our project requirement , each feature can be modifiable and removed from the membership,we have made it as a separate entity. • In our diagram, we have a relationship between Membership and feature. • Each Membership contains features [Many to Many]. (one membership contain more than one features and one feature is in more than one membership. • Feature_id along with Membership_id will make the record unique so it will be our composite key. • Each membership has some features but not all features are in a membership. Therefore features is in partial participation with membership. 	

Entity : Payment_Info	
Primary Key : Payment_Id	
Attributes	Description

Int Payment_Amount [Units in Dollars]	To keep the record of transaction amount.
String Payment_Method [< 100 chars]	To define the mode of payment used for transaction..
<ul style="list-style-type: none"> Payment_Info stores only one Credit Card and one Echeck.[One to One] <p>Payment_Info must at least have either Credit card or Echeck or(some payment method added in later). So, payment_info is in Total Participation.</p>	

Entity : Credit Card	
Primary Key : Card Number	
Attributes	Description
String Card Type [< 100 chars]	To know the type of card (Master/Visa/American Express).
Int CVV	To keep the record of card CVV
String Expiration_Date [< 15 chars 01/02/2021]	To keep the record of card expiration date
<ul style="list-style-type: none"> Credit card information is used for making payments when users subscribe for membership and Customers request for services. Payment_Info stores only one Credit Card and a Credit Card can only belong to one payment plan/info. Therefore there is a one to one relationship. 	

Entity : E-Check
Primary Key :

Attributes	Description
String Bank_Name [< 200 chars]	To know the name of bank
Int Account_Number	To keep the record of bank account number.
Int Routing_Number	To keep the record of Routing Number
<ul style="list-style-type: none"> E-Check information is used for making payments when users subscribe for membership and Customers request for services. Payment_Info stores only one e-check and an e-check can only belong to one payment plan/info. Therefore it is a one to one relationship. 	

Entity : Category	
Primary Key : Category_Id	
Attributes	Description
String Name [< 100 chars]	To define category name.
Long Int Parent_Category_Id [< 100 chars]	To Define the parent category of each category.

- As per our project requirement, Category is required as a separate entity to allow user to find the business using general category rather than the business name.
- In our ER diagram, we have represented category as a recursive relationship.
- Each category can be a general category [i.e Automobile] with parent_category NULL.
- Also each category can be a sub-category/Speciality [Air Conditioning, Brakes, General Repair, Tire/Wheel, etc.] which for example has a parent_Category of automobile.
- A category may not have any businesses that fall under it so it is in **partial participation**.
- In our ER diagram, Business **can have multiple** categories[**Many to Many**] relationship.(each business has one or more categories and each category can be of one or more businesses).
- Business_id along with category_id will make the record unique hence it will be our composite key for the many to many relationship.

Entity : Message	
Primary Key : Message_Id	
Attributes	Description
Message_Text [< 1000 chars]	To define the communication done between customer to Business and Current Customer to Previous Customer.
Sent_Time [< 100 chars]	To Define time and date of message.
<ul style="list-style-type: none"> • Message is used for the real-time interaction between business to customer as well as between customers themselves. • Each message is sent by a user. Hence message is in total participation. • Each message must also have a recipient. Hence message is in total participation with the user that receives the message. • In our ER diagram, Message is in relationship with user. As message can be sent by business or customer. • Each message is sent by one user , but one user can send multiple messages. Therefore, there is a one to many relationship. 	

- Each message is only received by one user (no group messages), but a user can receive multiple messages. **Therefore, there is a one to many relationship.**

Entity : Service_Job	
Primary Key : Job_Id	
Attributes	Description
String Job_Status [< 100 chars]	To keep track of each Job whether it is pending or completed.
Float Job_Amount	To Define Job charges.
String Job_Description [< 1000 chars]	To describe the problem faced by the customer and work to be done to solve the problem.
String Start_Date [< 15 chars 01/02/2021]	To Define the start date of the job.
String Completion_Date [< 15 chars 01/02/2021]	To Define the completion date of the job.
String Employee_Name [<150 chars]	To define the Business Employee name and contact detail to whom the job is assigned.
Int Transaction_No	To keep track of the order of the transactions done by the business. This is important because some transactions over a certain limit will be charged an extra amount depending on the membership.

- In our ER diagram, job is in relationship with the customer and the business.
- As per project requirement, job is required to keep track of all current jobs as well as to keep record of partly completed jobs and customer detail of to whom particular job is being provided.
- Each job must be requested by a customer , hence job is in **total participation with customer**.
- Each Job must be performed by a business,hence job is in **total participation with business**.
- Each customer can request multiple services, but a particular service job is only performed for one customer. Hence customer **requests** job is a **one to many relationship**.
- Each business can perform more than one job, hence there is a **one to many relationship**.
- Each Job has a sub-category/speciality that it falls under(for ex. Tire change). Hence it is in **total participation with category**.
- Each job falls under one sube-category/speciality, but a sub category can be related to multiple service_jobs, **hence there is a one to many relationship**.

Entity : Location	
Primary key : Location_Id	
Attributes	Description
String Street_Name [< 150 chars]	To Define street name of customer and business
String City [< 100 chars]	To Define city name of customer and business
Long int ZipCode	To Define Zip Code of customer and business
String Country [< 100 chars]	To Define country of customer and business
<ul style="list-style-type: none"> • We Decided to take this entity to reduce redundancy in our database. 	

Entity : Review	
Primary Key : Review_Id	
Attributes	Description
String Date [< 15 chars 01/02/2021]	To define date of the review given.
Float Rating [1 to 5]	To define ratings given.
String Review_Text [< 500 chars]	To describe and share the experience of the customer
<ul style="list-style-type: none"> Review is required as a separate entity for a business to keep track of all the reviews given by the customers. In our ER-diagram, Review is written by customer and it is evaluated by business. Customer can write multiple Reviews, but a review can only be written by one customer [One to Many]. A review can only evaluate one business but a business can be evaluated by multiple reviews. Hence, there is a many to one relationship. Number of reviews accessible by any business depends on the business's membership plan. 	

Entity : Promotion	
Primary key : Promotion_id	
Attributes	Description
String title [< 100 chars]	To represent title of promotion
String Start_Day [< 15 chars 01/02/2021]	To define when the promotion will start
String End_Day [< 15 chars 01/02/2021]	To define when the promotion will end

String Photo	To provide the advertisement banner
<ul style="list-style-type: none"> • Promotions belong to the business for marketing purpose. • Each business has access to promotions as per his membership plan. • Each business can have more than one promotion, but a promotion only belongs to one business hence Business has Promotion is a one to Many Relationship. 	

Description for mapping E-R to relational schema diagram

1. How you handled n:m relationships

For all of our n:m relationships, we created a new relation in the relational schema and used both primary keys of the entities to create a composite primary key for the new relation.

2. How you handled ternary relationships

We had no ternary relationships in our E-R diagram

3. How you handled recursive relationships in your diagram.

We only had one recursive relationship in our diagram and that was the relationship with the parent category and its sub-categories/specialities. We created a new attribute in our relation called parent category that was a foreign key that referenced the category I.D of the parent category(from the same table).

4. How you mapped multi-valued attributes to the Relational Model.

The only multi-valued attribute we had was the social media url for the business_user entity. We separated out the attribute into 3 social media urls for our relational diagram. If the business_User had less than 3 URL's then some fields in the relational diagram would be null.

5. How you handled inheritance.

We made each subclass disjoint. A business account represents a business as a whole and therefore cannot represent a single customer. Similarly a Super_User uses a faculty issued email that is separate from his personal email. A person can sign up for more than one type of account but these accounts themselves will be separate. We grouped all 3 subclasses of user into one relation. We then added an extra field into our relational diagram called User_Type to differentiate between different types of users in our user relation.

6. Constraints not represented in our diagram

A User can only pick a membership that coincides with their type of account. For example, a Business_user cannot pick a membership that only belongs to customer_users.

Date	Time	Participants	Activity
2/10/21	5:00-7:30P.M	all	First project meeting. Discuss the overall requirements gathering and details of the application
2/12/21	5:00-7:30P.M	all	Made a list of entities and attributes and started to form the base for our E.R diagram
2/13/21	2:00-4:30P.M	all	Worked on our E.R diagram and discussed the nuances of each relationship
2/16/21	7:00-9:00P.M	all	Completed most of the E.R diagram and began our mapping of the E.R diagram to relational schema
2/18/21	7:00- 9:00	all	Continued to work on the relational schema

	P.M		
2/22/21	7.00 to 9.00 PM	all	Finished the Relational Schema and began to prepare the final document
2/25/21	7.00 to 8.30 PM	all	Made slight changes to the E.R diagram and updated the relational schema as a result. Ironed out the final details of these two.
2/28/21	2.30 to 4.30 PM	all	Worked on the Final document and explanations, etc.
03/01/21	6.45 PM to 12.00 PM	all	Finished the final document and put everything together for the final submission.