PROJECT REPORT

ON

Online social network application “Who Does That”

Submitted By:

Sangeetha Eswaran. Net ID: s_e229
Rancy Thankachan. Net ID: r_t295
Jyotsna Shahi. Net ID: j_s1294

Under the Guidance Of:

Dr. Anne Hee Hiong Ngu
Department of Computer Science
Texas State University
San Marcos, Texas 78666

Table of Contents

Requirement Gathering:

- First, we went through the Project Specification Document and WDT PowerPoint slide provided by our professor.
- We read the project description carefully, identifying the nouns and business rules, and highlighted the keywords, which helped define entities and their attributes.
- We went through the following websites and analyzed them to gather the required information.
  - For example,
    https://www.yelp.com/
    https://www.angieslist.com/
- We thought from the perspective of end-users and created an account in https://www.yelp.com/ to search for the local businesses in our area based on location, category, and specialty, which are an essential part of our project.
- When we searched for a business which had more locations, it allowed us to see all the locations and also, we had the option to select nearby location.
- We also manually change the geographic area we are searching using the map and view the business.
- When we searched for a particular service and submitted a form, Yelp sent a request to all businesses nearby and allowed us to connect to the businesses, and they were able to reach us with their quote, thereby allowing us to schedule an appointment with the business we like.
- We also reviewed and rated a business that we visited before and saw how it was shown in Yelp.
- We visited the business page, and from there, we were able to share the business on Facebook and Twitter.
- We also saved the business to see whether it shows it in the profile page's favorites.
- We also thought from the business point of view and talked to the business owner who has a business account in Yelp and asked how reviews help their business and the Yelp's pros and cons.
- When we created a business account, it asked for a pin code and the business name to check whether it was already on Yelp. It showed the template and asked for the business category, address, website, and email address and allowed us to continue; otherwise, it might have asked us to claim the business listed there.
- We checked the same way in Angie's list also. It is mainly for homeowners who need service in plumbing, landscaping, painting, Heating, Roofing, Remodeling, House Cleaning.
- Only the members of Angie's list were able to give reviews.
- With all the information and knowledge, we gathered from these two websites, we discussed our project requirements, shared our ideas, and finalized all the project requirements.

Pros and Cons of Yelp and Angie’s List

1.Yelp

Yelp is a social site for everything local where we can search restaurants, hotels, dentists, and gyms and allow us to get professional services from landscapers, contractors, electricians, and plumbers.

Pros:

- Yelp provides a lower cost option to customer interaction than its competitors.
- It allows local businesses to gain positive reviews from customers in that area, attracting more customers in the future.
- Yelp displays 1-2 ads per search for a consumer. So, the local businesses get noticed at the local level, which may not be available through other online marketing options.
- The businesses in Yelp get much exposure since it has 150 million unique visitors per month.

Cons:

- Sometimes, several poor reviews that may not be authentic get displayed, and good reviews are not displayed, which is not good for the businesses.
- Yelp wants businesses on their site to commit to a 1-year contract. Hence removing a paid presence before that period can be a difficult process.
- Enhanced listings on Yelp are not affordable for small businesses.
- Businesses cannot remove reviews from Yelp if it seems false, and it will affect their business.
- The cost-per-click on a Yelp Ad is high and is around $10 per click.
- In order to generate a significant number of views and clicks, the budget needs to be around $1800, which is not affordable for small businesses.

Angie's List:

It is an online directory that allows users to read and publish crowd-sourced reviews of local businesses and contractors.

Pros:

- Users can grade a business in report-card style on A to F's scale, with A being the best and F being the worst.
- Businesses on this site give discounts often, which is good for consumers.
- Only member reviews count for the overall rating of the service providers.

- Businesses cannot pay to be on Angie's list, so reviews come from real people, which implies they are more accurate and unbiased.

Cons:

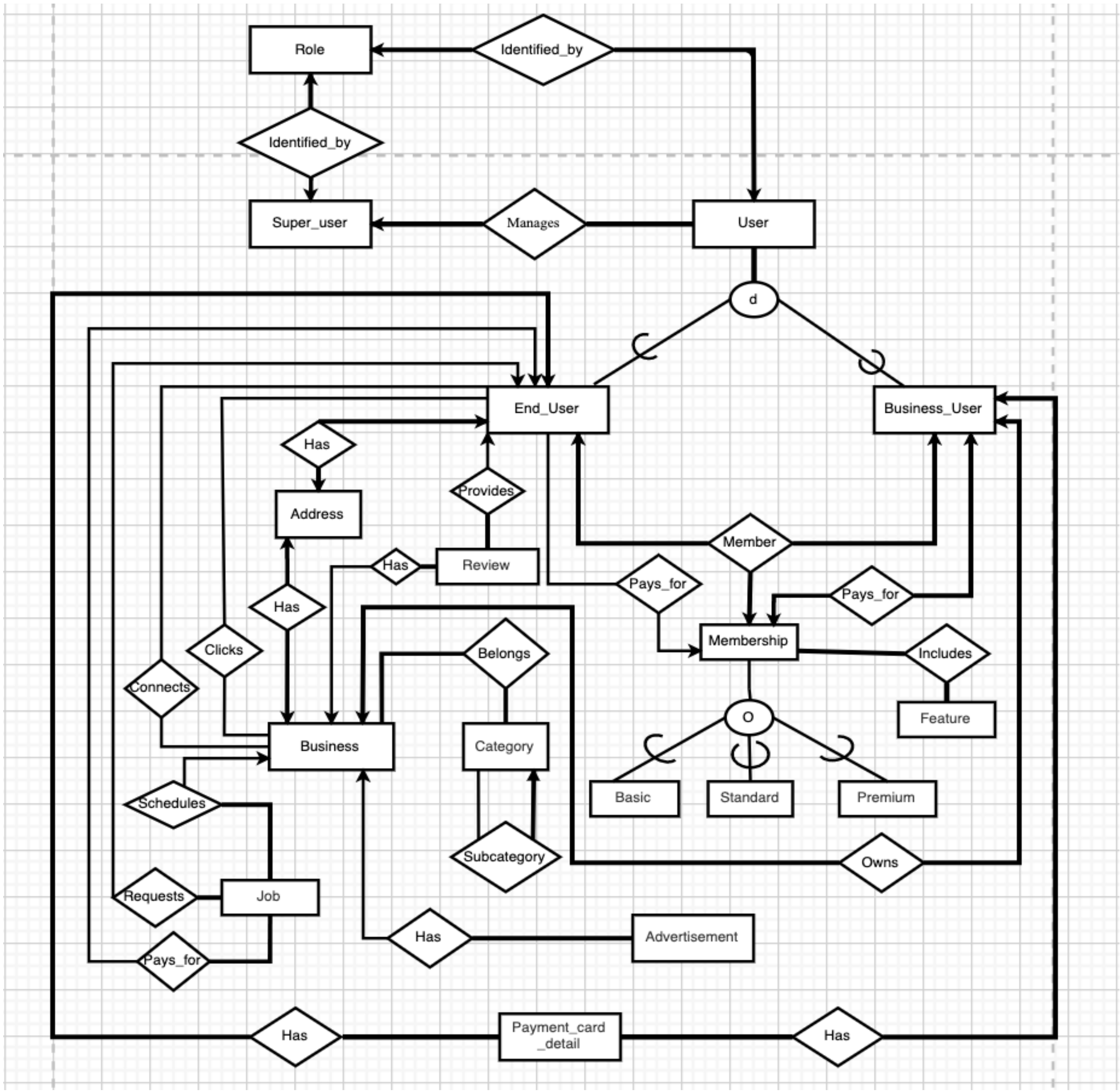
- Businesses need a decent number of reviews to be seen by the customers. The more, the better. Even if the business is good and only has fewer reviews, it will not be seen by customers.
- Businesses pay to be seen on the list even though Angie's list claims businesses are not paid to be on their list.
- Businesses cannot control negative reviews listed on their page in Angie's list, which will hinder their ability to get customers from Angie's list.
- Only member reviews impact businesses' grade scale and this impacts businesses if they do not get hired by Angie's List members.

Business Rules:

- " Who Does That? "is a mobile app designed to easily connect users with local businesses and services.
- WDT will have two account types: Business user and End User.
- Businesses and end users will have to register with WDT to use the services.
- WDT will provide three membership levels such as Basic, Standard, and Premium for businesses and end users to choose from.
- Each membership will have a name, description, cost, duration, and many features. Also, a feature can belong to more than one membership.
- Business and end users must be able to pay for their selected membership by providing payment information.
- Super user is the application admin who validates both business users' and end users' authenticity once they sign up on the app.
- Super user can add or remove any feature from the membership.
- The WDT application can have many Business users.
- Each business can belong to one or more predefined categories, and each category can have one or more subcategories.
- Each business can provide services in one or more locations.
- Each business will have one address associated with the business.
- Businesses can have many advertisements.
- Business users should provide details of business during the creation of a business account.
- The business user can connect with end users, respond to reviews, schedule jobs, etc., based on the membership level.
- Each business can view all of its ongoing jobs and the employees who are working on them.
- WDT can have many end users.

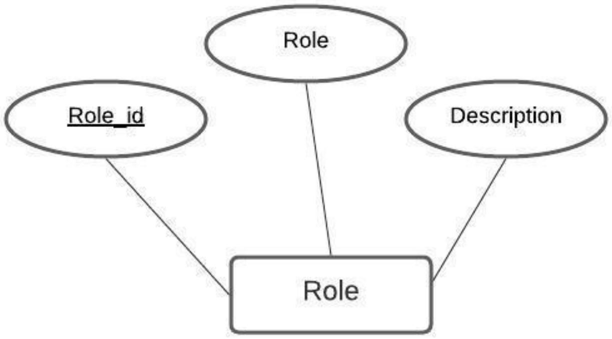
- End users will be able to search for a business based on their preferences such as categories, subcategories, and location/zip code, and the result set can be sorted either by distance, User rating, or alphabetically.
- One end user will have at least one address.
- Each end user can have many favorite businesses.
- End user can request jobs from businesses.
- Each end user can provide many reviews for many businesses.
- Whenever an end user clicks on a business, the action is captured to track a particular business's traffic.

E-R Diagram:

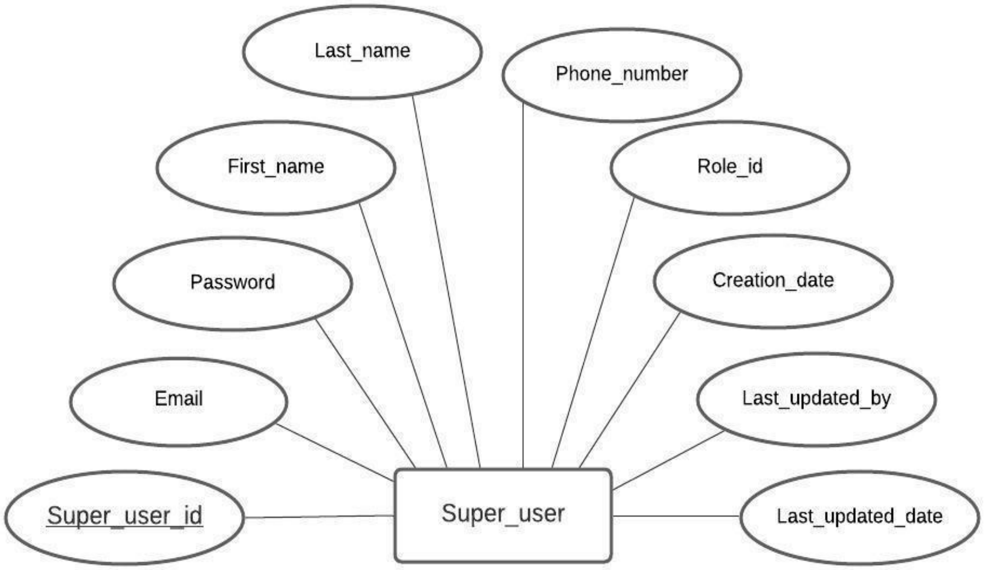


E-R Design of Individual Entity:

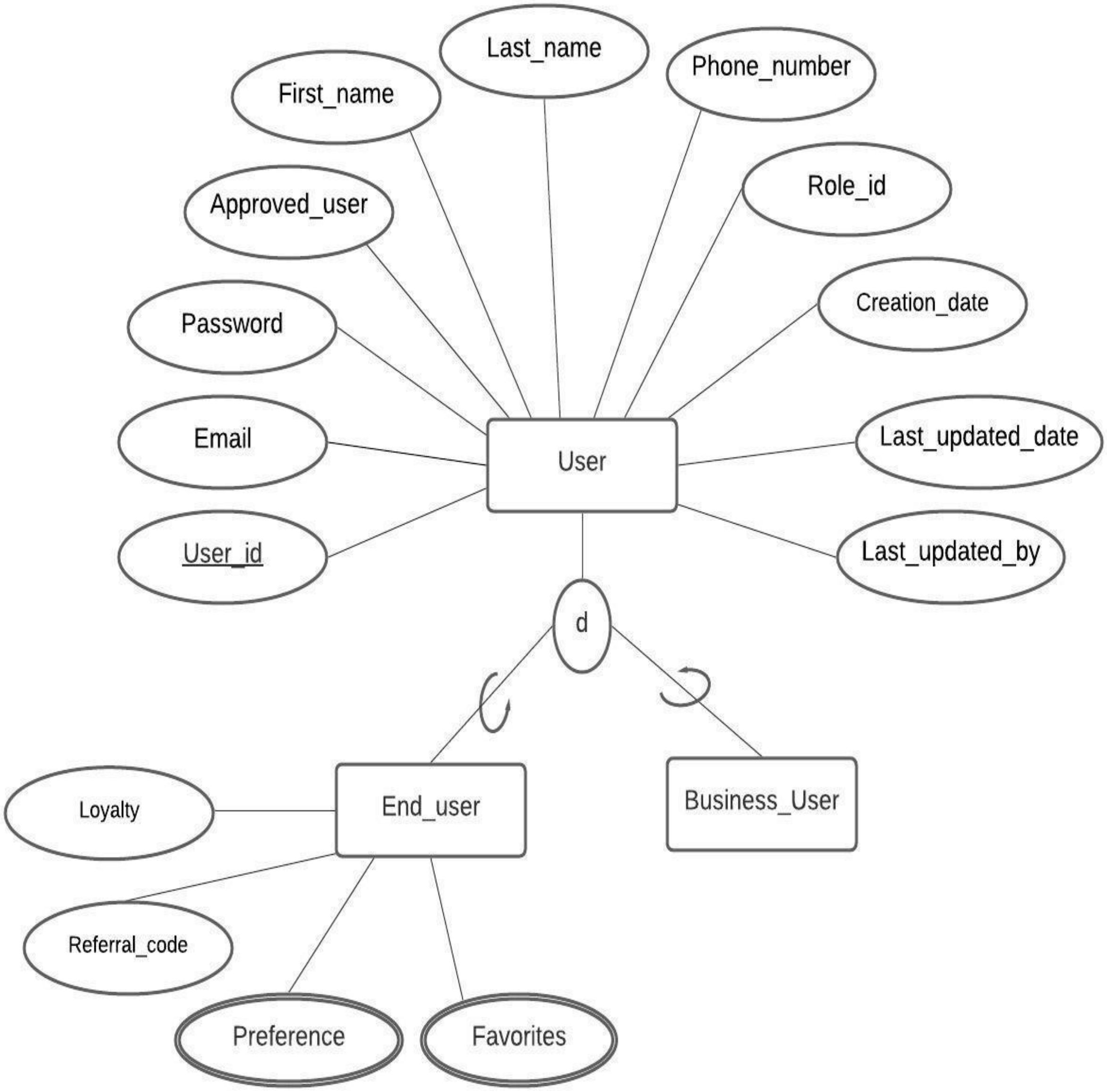
1. **Role**



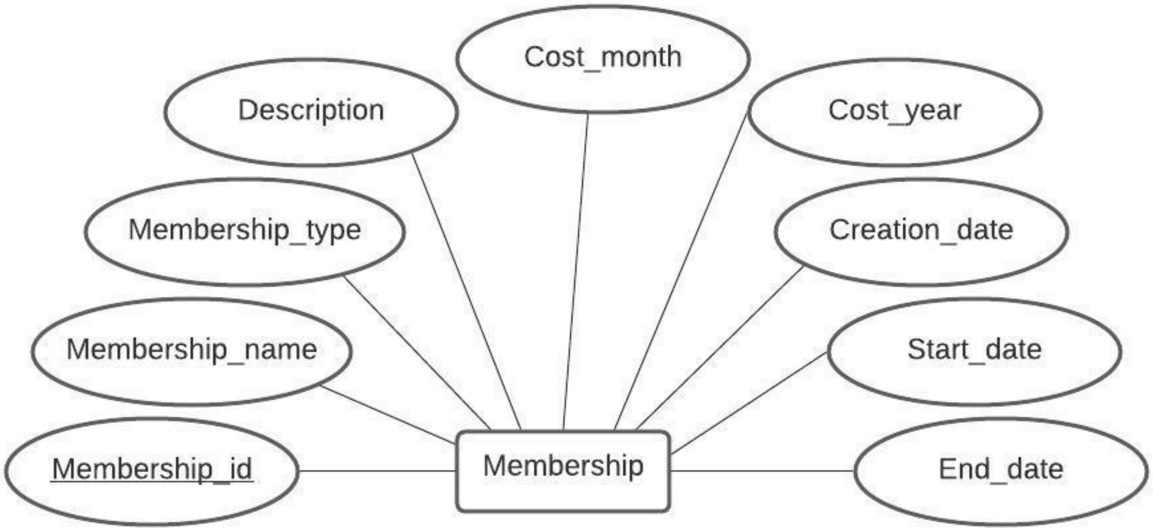
2. **Super_User**



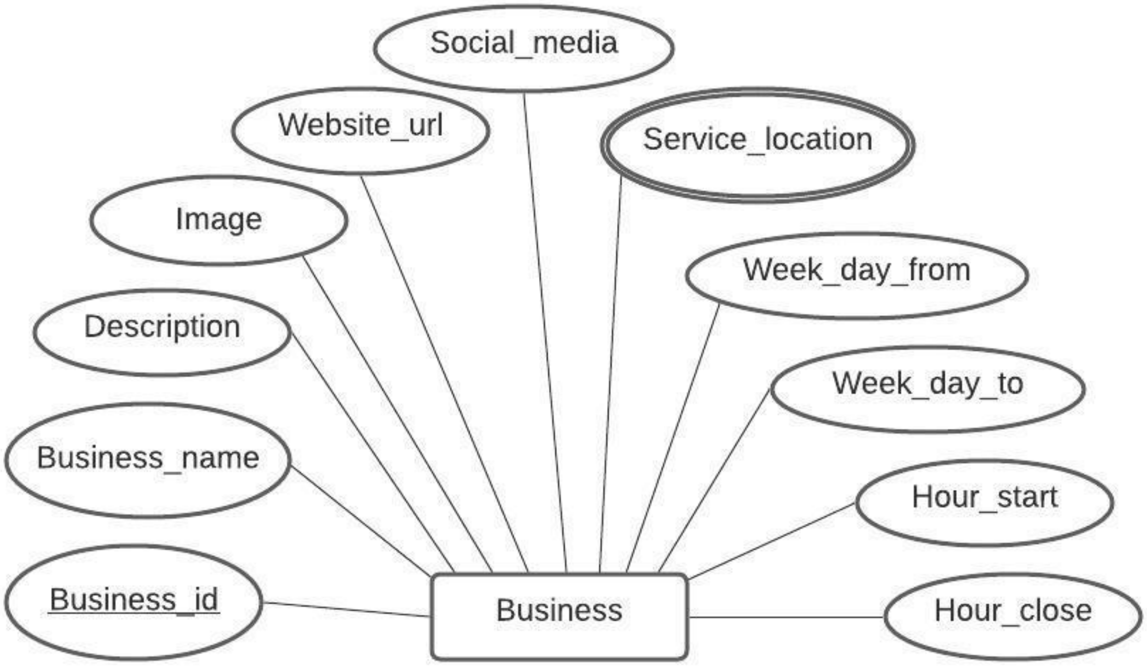
3. User


Last_name
Phone_number
First_name
Role_id
Approved_user
Creation_date
Password
Email
User
Last_updated_date
User_id
Last_updated_by
d
Loyalty
End_user
Business_User
Referral_code
Preference
Favorites

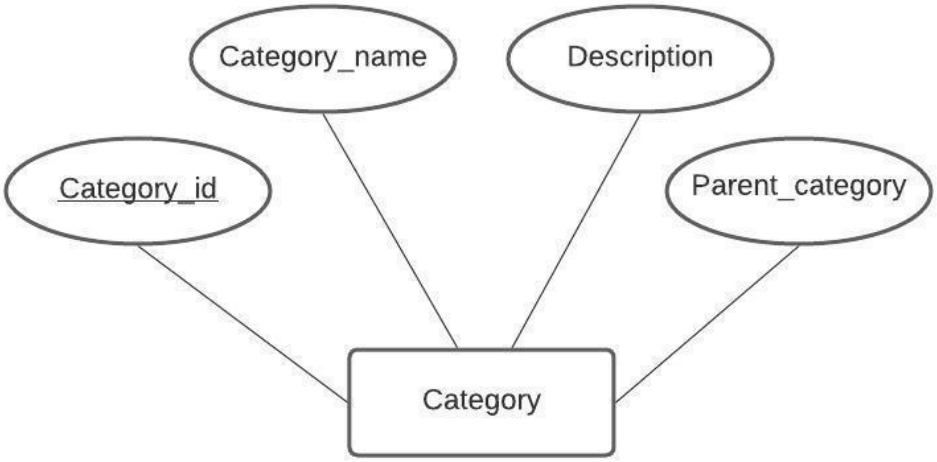
4. Membership



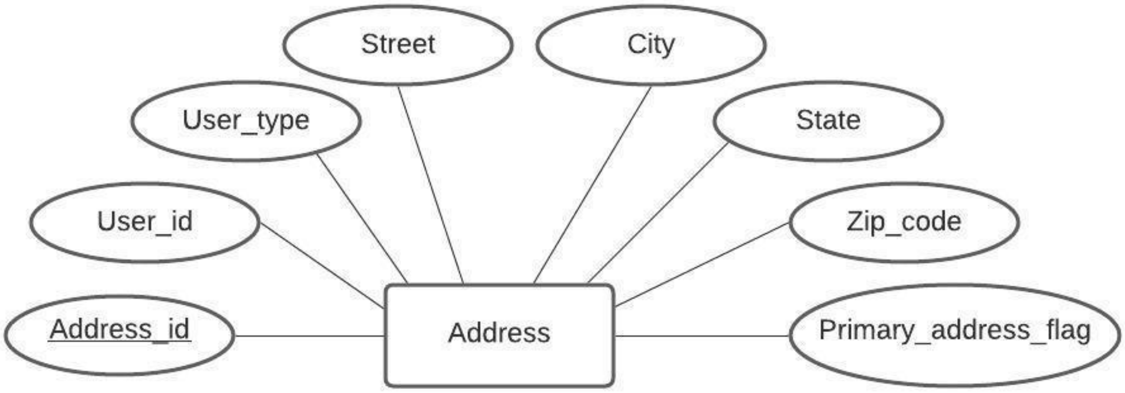
5. Business



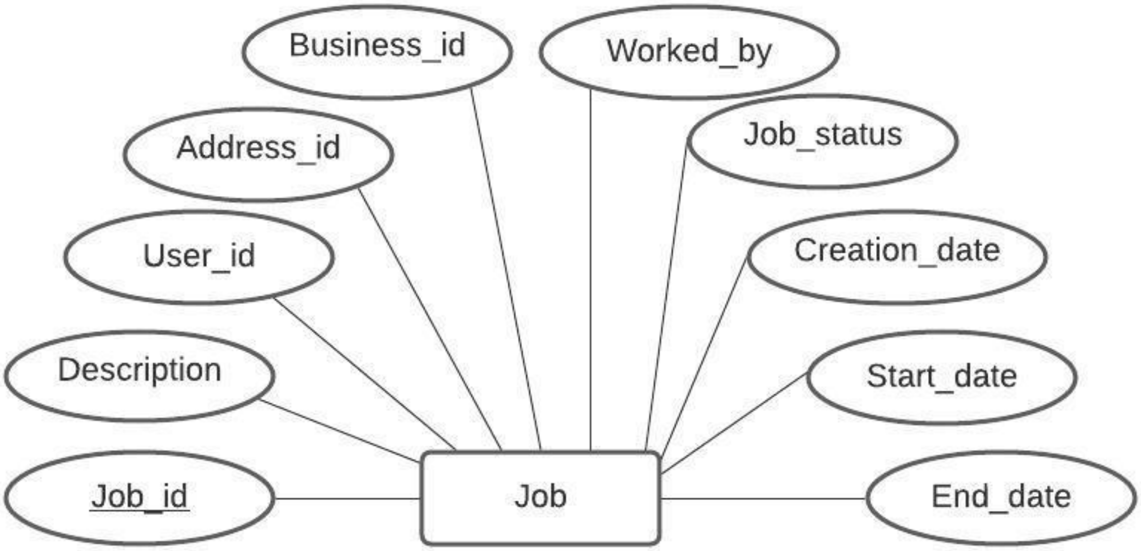
6. Category



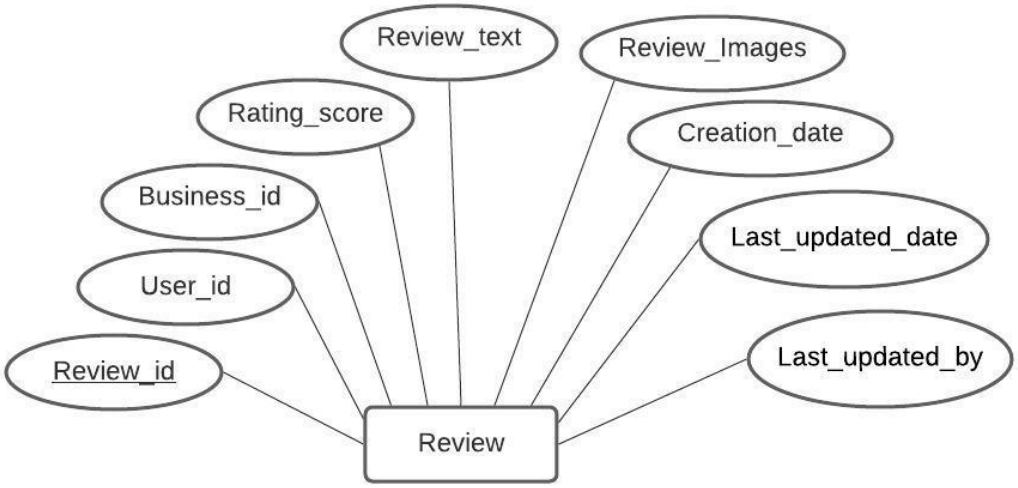
7. Address



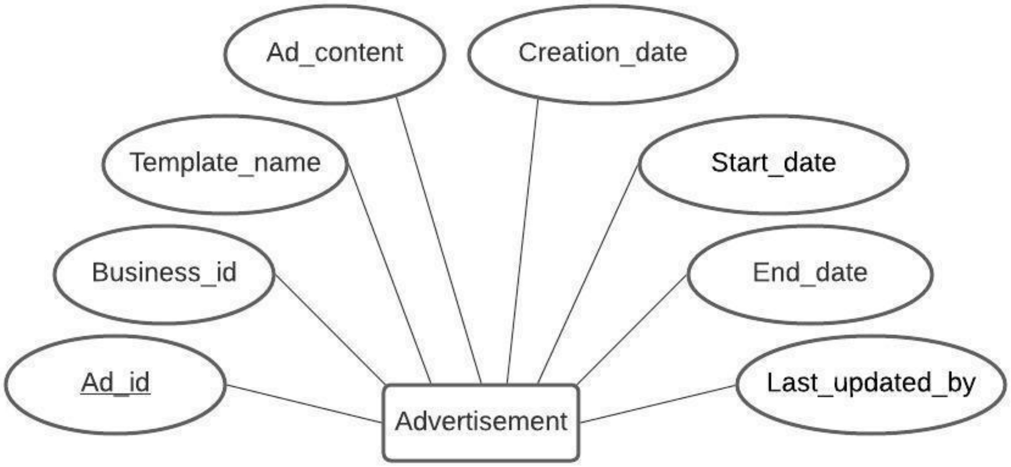
8. Job



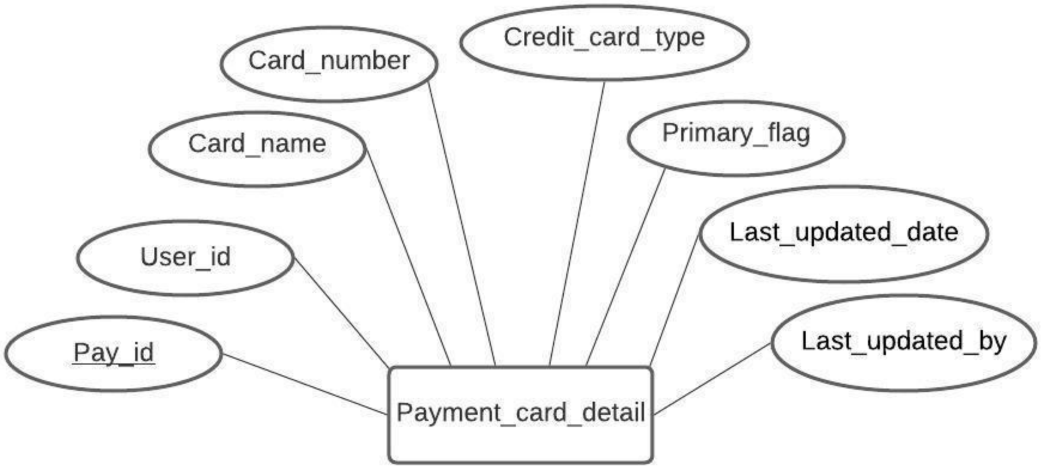
9. Review



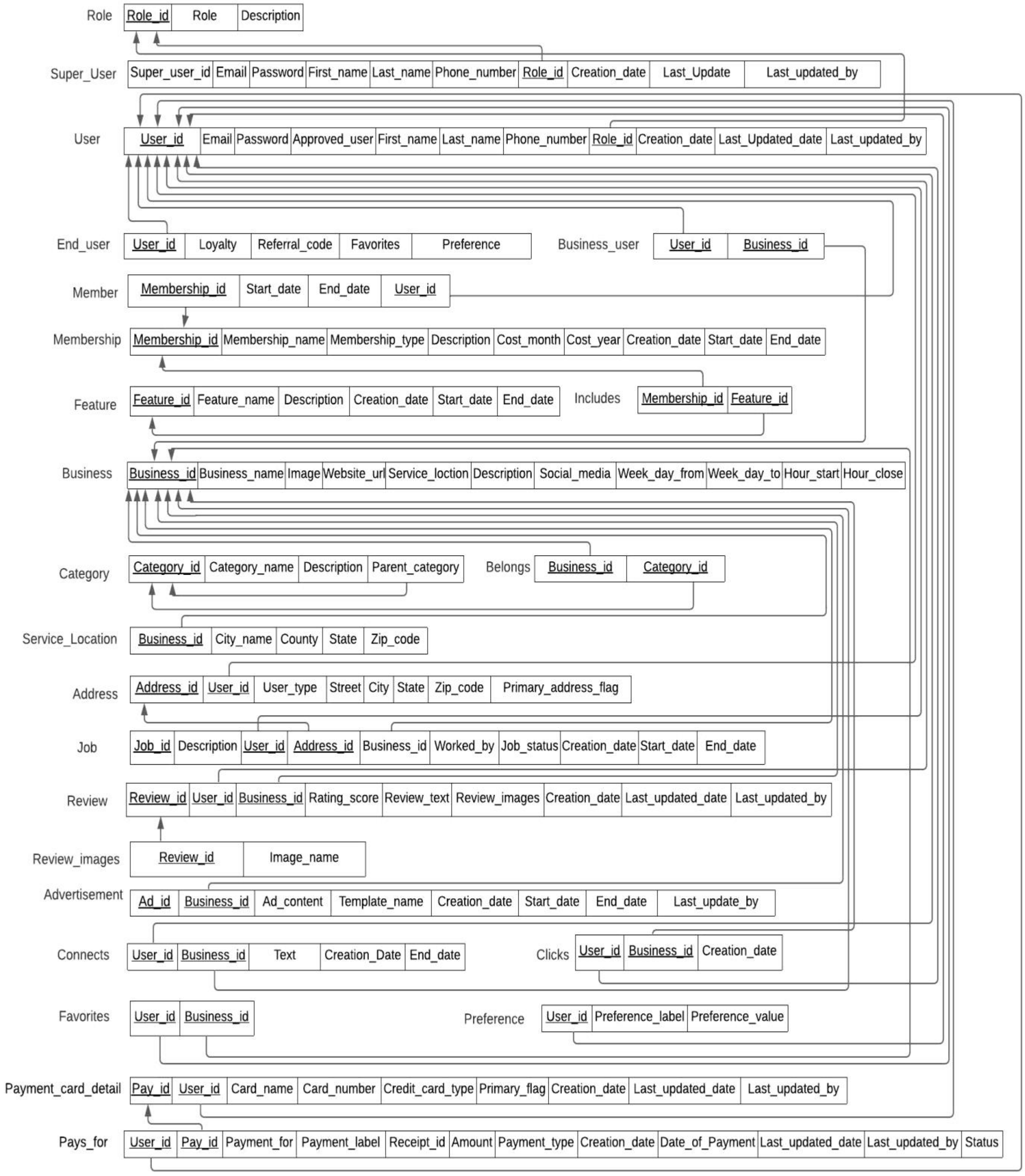
10. Advertisement



11. Payment_card_detail



Relational Schema Diagram:



Role

Role_id	Role	Description

Super_User

Super_user_id	Email	Password	First_name	Last_name	Phone_number	Role_id	Creation_date	Last_Update	Last_updated_by

User

User_id	Email	Password	Approved_user	First_name	Last_name	Phone_number	Role_id	Creation_date	Last_Updated_date	Last_updated_by

End_user

User_id	Loyalty	Referral_code	Favorites	Preference

Business_user

User_id	Business_id

Member

Membership_id	Start_date	End_date	User_id

Membership

Membership_id	Membership_name	Membership_type	Description	Cost_month	Cost_year	Creation_date	Start_date	End_date

Feature

Feature_id	Feature_name	Description	Creation_date	Start_date	End_date

Includes

Membership_id	Feature_id

Business

Business_id	Business_name	Image	Website_url	Service_loction	Description	Social_media	Week_day_from	Week_day_to	Hour_start	Hour_close

Category

Category_id	Category_name	Description	Parent_category

Belongs

Business_id	Category_id

Service_Location

Business_id	City_name	County	State	Zip_code

Address

Address_id	User_id	User_type	Street	City	State	Zip_code	Primary_address_flag

Job

Job_id	Description	User_id	Address_id	Business_id	Worked_by	Job_status	Creation_date	Start_date	End_date

Review

Review_id	User_id	Business_id	Rating_score	Review_text	Review_images	Creation_date	Last_updated_date	Last_updated_by

Review_images

Review_id	Image_name

Advertisement

Ad_id	Business_id	Ad_content	Template_name	Creation_date	Start_date	End_date	Last_update_by

Connects

User_id	Business_id	Text	Creation_Date	End_date

Clicks

User_id	Business_id	Creation_date

Favorites

User_id	Business_id

Preference

User_id	Preference_label	Preference_value

Payment_card_detail

Pay_id	User_id	Card_name	Card_number	Credit_card_type	Primary_flag	Creation_date	Last_updated_date	Last_updated_by

Pays_for

User_id	Pay_id	Payment_for	Payment_label	Receipt_id	Amount	Payment_type	Creation_date	Date_of_Payment	Last_updated_date	Last_updated_by	Status

Relational Schema Definition and Domain of all Attributes:

Role (**Role_id**, Role, Description)

Super_user (**Super_user_id**, Email, Password, First_name, Last_name, Phone_number, **Role_id**, Creation_date, Last_updated_date, Last_updated_by)

User (**User_id**, Email, Password, Approved_user, First_name, Last_name, Phone_number, **Role_id**, Creation_date, Last_updated_date, Last_updated_by)

End_user (**User_id**, Loyalty, Referral_code, Favorites, Preference)

Business_User(**User_id**, **Business_id**)

Member (**Membership_id**, Start_date, End_date, **User_id**)

Membership (**Membership_id**, Membership_name, Membership_type, Description, Cost_month, Cost_year, Creation_date, Start_date, End_date)

Feature (**Feature_id**, Feature_name, Description, Creation_date, Start_date, End_date)

Includes (**Membership_id, Feature_id**)

Business (**Business_id**, Business_name, Image, Website_url, Service_location, Description, Social_media, Week_day_from, Week_day_to, Hour_start, Hour_close)

Category (**Category_id**, Category_name, Description, Parent_category)

Service_Location (**Business_id**, City_name, County, State, Zip_code)

Address **(Address_id**, **User_id**, User_type**,** Street, City, State, Zip_code, Primary_address_flag)

Job (**Job_id**, Description,User_id, Address_id, **Business_id**, Worked_by, Job_status ,Creation_date, Start_date, End_date)

Review (**Review_id**, **User_id**, **Business_id**, Rating_score, Review_text, Review_images, Creation_date, Last_updated_date, Last_updated_by)

Review_images (Review_id, Review_name)

Advertisement (**Ad_id**, **Business_id**, Ad_content,Template_Name, Creation_date, Start_date, End_date, Last_updated_by)

Connects (**User_id**, **Business_id**, Text, Creation_date, End_date)

Clicks (**User_id**, **Business_id**, Creation_date)

Favorites (**User_id**, **Business_id**)

Preference (**User_id**, Preference_label, Preference_value)

Payment_Card_Detail (**Pay_id**, **User_id**, Card_name, Card_number, Credit_card_type, Primary_flag, Creation_date, Last_updated_date, Last_updated_by)

Pays_for (**User_id**, Pay_id, Payment_for, Payment_label, Receipt_id, Amount, Payment_type, Creation_date, Date_of_payment, Last_updated_date, Last_updated_by, Status)

Domain of all attributes:

Role Types

Int	Role_id	
Varchar2(25)	Role	// Represents roles of User and Super_user
Varchar2(200)	Description	//Description about different roles

Super_user Types

int	Super_user_id	//Primary key, not null
Varchar2(320)	Email	//Super_user email
Varchar2(25)	Password	//Super_user password
Varchar2(50)	First_name	//Super_user first name
Varchar2(50)	Last_name	// Super_user last name
Varchar2(14)	Phone_number	// Super_user phone number
Int	Role_id	// Foreign key for role entity,
Date	Creation_date	//Creation date of Super_user in mm-dd-yyyy
Date	Last_updated_date	//Last updated date of Super_user in mm-dd-yyyy
Int	Last_updated_by	//Last updated by Super_user

User Types

Int	User_id	//Primary key, not null
Varchar2 (320)	Email	//User's email id, not
Varchar2(25)	Password	//Stores User password
Boolean	Approved_user	//Indicates User is approved or not, not null
Varchar2(50)	First_name	// User first name, not null
Varchar2(50)	Last_name	// User last name, not null
Varchar2(14)	Phone_number	// User phone number
Int	Role_id	//Foreign key for role entity, not null
Date	Creation_date	//Creation date of User in mm-dd-yyyy
Date	Last_updated_date	//Last updated date of User in mm-dd-yyyy
Int	Last_updated_by	//Represents Id of user who updates User last

End_user Types

Int	User_id	// Foreign key for User entity
Number(5)	Loyalty	//Represents End_user loyalty points
Varchar2(50)	Referral_code	//Represents End_user referral code
Int	Favorites	//Represents favorite Business_id,multivalued
Varchar2(20)	Preference	//Represents User Preference,multivalued

Business_user Types

Int	User_id	// Foreign key for User entity
Int	Business_id	//Foreign key for Business

Member Types

Int	Membership_id	// Foreign key for Membership entity
Int	User_id	// Foreign key for User entity
Timestamp	Start_date	//Start date of member's Membership in mm-dd-yyyy
Timestamp	End_date	//End date of member's Membership in mm-dd-yyyy

Membership Types

Int	Membership_id	//Primary key, not null
Varchar2(50)	Membership_name	//Gives the name of Membership

Varchar2(1)	Membership_type	//Gives type of Membership
Varchar2(500)	Description	//Description about Membership
Double	Cost_month	//Monthly cost of Membership
Double	Cost_year	//Yearly cost of Membership
Date	Creation_date	//Membership creation date in mm-dd-yyyy
Date	Start_date	//Membership start date in mm-dd-yyyy
Date	End_date	//Membership end date in mm-dd-yyyy

Feature Types

Int	Feature_ id	//Primary key, not null
Varchar2(50)	Feature_name	//Represents Feature name
Varchar2(500)	Description	//Describes about Feature
Date	Creation_date	//Creation date of Feature in mm-dd-yyyy
Date	Start_date	//Start date of Feature in mm-dd-yyyy
Date	End_date	//End date of Feature in mm-dd-yyyy

Includes Types

Int	Membership_id	// Foreign key for Membership entity
Int	Feature_id	// Foreign key for Feature entity

Business Types

Int	Business_id	
Varchar2(50)	Business_name	//Name of Business, not null
Varchar2(100)	Image	//Image of Business, not null
Varchar2(100)	Website_url	// Url of the Business, not null
Varchar2(100)	Service _location	//Service location of Business
Varchar2(500)	Description	//Description of Business
Varchar2(100)	Social_media	//Social media link of Business
Varchar2(10)	Week_day_from	//Start weekday/weekend of Business
Varchar2(10)	Week_day_to	//End weekdayweekend of Business
Time	Hour_start	//Starting hour of Business
Time	Hour_close	//Closing time of Business

Category Types

Int	Category_id	//Primary key, not null
Varchar2(50)	Category_name	//Name of Category
Varchar2(500)	Description	//Description of Category and Subcategory
Int	Parent_category	//Foreign key of Category entity

Belongs Types

Int	Business_id	//Foreign key of Business entity
Int	Category_id	//Foreign key of Category entity

Service_location Types

Int	Business_id	// Foreign key for business
Varchar2(50)	City_name	//Name of city where business provide service
Varchar2(50)	County	//County name of city
Varchar2(50)	State	
Varchar2(50)	Zip_code	

Address Types

Int	Address_id	//Primary key, not null
Int	User_id	// Foreign key for User entity
Varchar2(1)	User_type	//Represents Whether User is Business_User or End_ user
Varchar2(50)	Street	//User’s street
Varchar2(50)	City	//User’s city
Varchar2(2)	State	//User’s state
Varchar2(10)	Zip_code	//User’s zip code
Boolean	Primary_address_flag	// User’s primary address indication

Job Types

Int	Job_id	//Primary key, not null
Varchar2(500)	Description	//Description of Job
Int	User_id	// Foreign key for User entity
Int	Address_id	// Foreign key for Address entity
Int	Business_id	// Foreign key for Business entity
Varchar2(50)	Worked_by	//Name of job handler
Varchar2(50)	Job_status	
Date	Creation_date	//Creation date of Job in mm-dd-yyyy
Date	Start_date	//Start date of Job in mm-dd-yyyy
Date	End_date	//End date of Job in mm-dd-yyyy

Review Types

Int	Review_id	//Primary key, not null
Int	User_id	// Foreign key for User entity
Int	Business_id	// Foreign key for Business entity
Double	Rating_score	//Rating of Business
Varchar2(500)	Review_text	//Text review of business
Varchar2(50)	Review_Images	//Images in Review
Date	Creation_date	//Review date in mm-dd-yyyy
Date	Last_updated_date	//Last updated review in mm-dd-yyyy
Int	Last_updated_by	//Represents Id of user who update Review last

Review_images Types

Int	Review_id	//Foreign key, not null
Varchar2(50)	Image_name	//Represents Review Image name

Advertisement Types

Int	Ad_id	//Primary key, not null
Int	Business_id	// Foreign key for Business entity
Varchar(500)	Ad_content	//Advertisement content
Varchar2(25)	Template_name	//Template name of Advertisement
Date	Creation_date	//Creation date of Advertisement
Date	Start_date	//Start date Advertisement
Date	End_date	//Expiry date of Advertisement
Int	last_update_by	//Represents Id of user who update last

Connects Types

Int	User_id	// Foreign key for User entity
Int	Business_id	// Foreign key for Business entity
Varchar2(250)	Text	//Text to connect
Date	Creation_date	//Date of creation in mm-dd-yyyy
Date	End_date	//End date of connect in mm-dd-yyyy

Clicks Types

Int	User_id	// Foreign key for User entity
Int	Business_id	// Foreign key for Business entity
Date	Creation_date	

Favorites Types

Int	User_id	// Foreign key for User entity
Int	Business_id	// Foreign key for Business entity

Preference Types

Int	User_id	// Foreign key for User entity
Varchar2(20)	Preference_label	//Gives the Preference type
Varchar2(5)	Preference_value	//Value of Preference

Payment_card_details Type

Int	Pay_id	
Varchar2(25)	Card_name	//Name of card holder
Number	Card_number	//Card number of credit card
Varchar2(10)	Credit_card_type	//Type of card user using
Boolean	Primary_flag	//Gives whether card is default or not
Date	Creation_date	//Payment card detail creation date in mm-dd-yyyy
Date	Last_updated_date	// Payment card detail last update in mm-dd-yyyy
Int	Last_updated_by	// Payment card detail last update by user

Pays_for Type

Int	Payment_for	//For what user paying for
Varchar2(25)	Payment_label	//For what the payment is for
Int	Receipt_id	//Receipt of payment
Double	Amount	//Amount paid
Varchar2(50)	Payment_type	//Mode of payment
Date	Date_of_Payment	//Payment date in mm-dd-yyyy
Date	Last_updated_date	//last payment update in mm-dd-yyyy
Int	Last_updated_by	//last updated by user
Varchar2(50)	Status	//Payment status

Design Decision and Domain of all Attributes (for E-R Diagram):

Entity: **Role**	
Primary key attribute: Role _id	
Attributes	Description
Varchar2(25) Role	Helps to identify whether he/she is Super_ user or Business_user or End_user
Varchar2(200) Description	It gives description of different roles played by the user.
● Each User is **Identified_by** a Role. [One to One]. ● Every Role should belong to a User. So Role is in **Total Participation.** ● Super_user, who is admin of WDT is **Identified_by** a Role as well. [(One to One]. ● All Super_users should have a role. So Role is in **Total Participation.**	

Entity: **User**	
Primary Key attribute: User_id	
Attributes	Description
Varchar2(320) Email	Represents User's email address
Varchar2(25) Password	Stores User's password
Booelan Approved_user	To check whether the User is approved or not
Varchar2(50) First_name	Represents User's first name
Varchar2(50) Last_name	Represents User's last name
Varchar2(14) Phone_number	Represents User's phone number
Int Role_id	To identify whether the User is a Business_user or End_user
Date Creation_date	It represents User details creation date
Date Last_updated_date	It represents User details last updated date
Int Last_updated_by	It represents Id of the User or Super_user who updated User details last.
● We have **User** entity using the disjoint concept in inheritance. ● **User** has two subclasses named Business_User, and End_User. ● A Super_user **Manages** all Users.[One to Many]. So User is in **Total Participation.**	

● Each User is **identified_by** a Role. [One to One]. ● All users should be identified by one of the roles. **So User is in Total Participation.**

Entity: **Super_user**	
Primary Key attribute:User_id	
Attributes	Description
Varchar2(320) Email	Represents Super_user's email address
Varchar2(25) Password	Stores Super_user's password
Varchar2(50) First_name	Represents Super_user's first name
Varchar2(50) Last_name	Represents Super_user's last name
Varchar2(14) Phone_number	Represents Super_user's phone number
Int Role_id	Represents Super_user's Role_id
Date Creation_date	It represents Super_user details creation date
Date Last_updated_date	It represents Super_user details last updated date
Int Last_updated_by	It represents Id of the User or Super_user who updated the record last
● A Super_user **Manages** all Users (Business_users and End_users) [One to Many]. ● All Users should be managed by a Super_User. So Super_user is in **Total Participation**. ● Each Super_user is identified by a role.[One to One]. ● All Super_users must be identified by a role. So Super_user is in **Total Participation**.	

Entity: **End_user**	
Primary key attribute: User_id	
Attributes	Description
Number(5) Loyalty	To store loyalty points earned by end users by connecting with business, writing honest reviews, or doing referrals to WDT. These

	points can be redeemed while paying businesses.
Varchar2(50) Referral_code	It represents the static referral code of the End_user.
Int Favorites	It represents favorite Business_id of the End_user
Varchar2(20) Preference	It denotes User’s preference(For eg: Rating, Category,Subcategory)

- Each End_user has a membership. [One to One].
- All End_users should be a **Member** and must have membership. So, End_User is in **Total participation.**
- Except Basic members, an End_user **Pays_for** a membership.
- Since Basic membership is free for end users, not all end users **Pays_for** a membership, So End_User is in **Partial participation.**
- One end user **Provides** one or more reviews.[One to Many].
- Not all end users **Provides** a review. So End_User is in **Partial Participation.**
- Each End user has a Address.[One to One].
- All End_users must have an Address. So End_User is in **Total Participation.**
- More than one End_user **Clicks** any number of businesses. [Many to Many].
- Not all End_user **Click**s on business. So End_User is in **Partial Participation.**
- More than one End_user **Connects** to more than one Business.[Many to Many]
- Not all End_user **Connects** to business. So, End_User is in **Partial Participation**
- One end user can **Requests** any number of jobs.[One to Many].
- Only some End_users request Job from the Business. **So** End_User is in **Partial Participation.**
- Each End_user **Pays_for** one or more jobs they have requested.[One to Many].
- End_users **Pays_for** requested Job. **So End_User** is in **Total Participation.**
- Each End_user can have more than one **Payment_card_detail**.[One to Many].
- All End_users must have a Payment_card_detail. So End_User is in **Total Participation.**

Entity: **Membership**	
Primary key attribute: Membership_id	
Attributes	Description
Varchar2(50) Membership_name	It gives the name of the Membership whether it is Basic, Standard or Premium.
Varchar2(1) Membership_type	It denotes whether the Membership is for End_user or Business_user. It will store “E” for End_user and “B” for Business_user

Varchar2(500) Description	It gives description of the different memberships.
Double Cost_month (Units are in dollars and cents)	Represents cost for the Membership per month.
Double Cost_year (Units are in dollars and cents)	Represents cost for the Membership per year
Date Creation_date	It denotes Membership creation date
Date Start_date	It denotes existence of this particular type Membership
Date End_date	It denotes how long this particular Membership type exists.

- We generalized **Membership** entity using overlapping concept.
- We have three subclasses named Basic, Standard and Premium under Membership superclass.
- Each end user is a **Member** of a Membership.[One to One].
- Each business user is a **Member** of a Membership.[One to One]
- All Membership should belong to either Business_user or End_user. So Membership is in **Total Participation.**
- End_user **Pays_for** a Membership**.**
- Basic Membership is free for End_Users**.** So it is in **Partial Participation.**
- Each Business_user **Pays_for** a Membership.[One to One].
- All Memberships should be paid by Business_users. So Membership is in **Total Participation**
- All Memberships **Includes** one or more Features and one Feature may belong to more than one Membership [Many to Many].
- All Membership includes at least one Feature. So Membership is in **Total Participation**

Entity: **Feature**	
Primary key attribute: Feature_id	
Attributes	Description
Varchar2(50) Feature_name	Represents Feature's name
Varchar2(500) Description	It describes more about Features
Date Creation_date	It denotes Feature created date
Date Start_date	It denotes Feature starting date
Date End_date	It denotes Feature expiry date

- One Feature can be included in more than one Membership, and Membership **Includes** many Features.[Many to Many].
- All Features should belong to one or more Membership. So Feature is in **Total Participation.**

Entity: Address	
Primary Key attribute:Address_id	
Attributes	Description
Varchar2(1) User_type	It denotes whether the User is Business_user or End _user
Varchar2(50) Street	Represents street address of the User
Varchar2(50) City	Represents User’s city name
Varchar2(2) State	Represents User’s state
Varchar2(10) Zip code	Represents User’s zip code
Boolean Primary_address_flag	To identify whether the User’s Address is primary or not
● Each End_user **Has** an Address.[One to One] ● End_Users must have an Address.So Address is in **Total Participation.** ● A Business **Has** an address.[One to One] ● Business must have an Address So Address is in **Total Participation.**	

Entity: **Business**	
Primary key attribute: Business_id	
Attributes	Description
Varchar2(50) Business_name	Represents Business name
Varchar2(100) Image	Represents store front image of the Business
Varchar2(100) Website_url	It represents a Business website url. Clicking on this url will leave WDT and takes to Business website.
Varchar2 (100) Service_location	It represents service location of the Bus iness.This attribute is multivalued as the Business may have many service locations.
Varchar2 (500) Description [<500 chars]	To describe more about the Business like how long they are in business and what are their specialties and it should be less than 500 characters.
Varchar2 (100) Social_media	It represents Social_media link. It is multivalued attribute as there is more than one Social_media (For eg: Facebook,Twitter).
Varchar2(10) Week_day_from	It represents starting week day/week end of when service is provided.(For eg:Mon, Sat)
Varchar2(10) Week_day_to	It represents ending week day/week end of when service is provided. (For eg:Fri, Sun)
Time Hour_start	It represents opening time of the Business
Time Hour_close	It represents closing time of the Business

- Every Business_user **Owns** a Business. [One to One].
- All the Businesses must be owned by some Business_user. So Business in **Total Participation.**
- Each Business_User Has Payment_Card_Detail.[One to One].
- One or more businesses can have one or more Categories and each category may belong to more than one Business. [Many to Many].
- Business should **Belongs** to at least one category. So, Business is in **Total Participation.**
- Business **Has** one or more reviews**.**[One to Many]
- Businesses may or may not have reviews. So Business is in **Partial Participation**.
- Each Business **Has** an Address.[One to One].
- All the Businesses should have an Address. So Business is in **Total Participation**.
- One or more End_users **Clicks** one or more Businesses. [Many to Many].
- Not all businesses are clicked by End users. So Business is in **Partial Participation**.
- One or more Businesses **Connects** to one or more End_users.[Many to Many].
- Not all Businesses connect to End_users. So Business is in **Partial Participation**.
- One Business **Schedules** any number of jobs.[One to Many].
- Not all businesses schedule jobs to End_users. So it is **Partial Participation**.
- Each Business **Has** one or more advertisements.[Many to Many]
- Not all businesses have advertisements on social media or business pages. So Business is in **Partial Participation**.

Entity: **Category**	
Primary key attribute: Category_id	
Attributes	Description
Varchar2(50) Category_name	It represents the Category name to which the Business belongs.
Varchar2(500) Description	Description of Category and also subcategories if available.
Int Parent_category	If Category is parent category itself, then the value will be NULL; otherwise Subcategory will have parent category id.

- We used the recursive relationship to show category and subcategory in our E-R Diagram.
- One or more categories **Belongs** to one or more businesses[Many to Many].
- All categories should belong to one or more businesses. So Category is in **Total Participation** with Business.
- A Category can have one or more subcategories.[One to Many]

- All Subcategories of the Business must belong to a Category. So Subcategory is in **Total Participation.**

Entity: **Payment_card_details**	
Primary key attribute: Pay_id	
Attributes	Description
Varchar2(25) Card_name	It represents the name on the card
Number Card_number	It represents credit card number
Varchar2(10) Credit_card_type	It represents credit card type
Boolean Primary_flag	It denotes whether credit card is default card or not
Date Creation_date	It denotes creation date of payment card details
Date Last_updated_date	It denotes last updated date of payment card details
Int Last_updated_by	Represents User_id or Super_user_id who update payment card details last

- Each business user and the end user **Has** one or more payment card detail. [One to Many].
- Both business users and end users must have at least one payment card detail. So Payment_card_detail is in **Total Participation.**

Entity: **Review**	
Primary key attribute: Review_id	
Attributes	Description
Double Rating_score	It denotes rating score
Varchar2(500) Review_text	It denotes the review text provided by the end user.
Varchar2(50) Review_images	It represents image that is posted along with the review.It is multivalued as it may contain more than one image.
Date creation_date	It denotes Review created date.
Date last_updated_date	It denotes Review updated date.
Int last_updated_by	Represents User or Super_user's Id who update the Review last

- An End user **Provides** one or more Reviews. [One to Many]
- All Reviews must be provided by End_users. So, Review is in **Total Participation.**
- A Business **Has** one or more Reviews.[One to Many].

- All Reviews should belong to Businesses. So Review is in **Total Participation.**

Entity: **Job**	
Primary key attribute: Job_id	
Attributes	Description
Varchar2(500) Description	It describes about the Job provided by the Business
Varchar2(50) Worked_by	It gives name of the employee who is handling the Job
Varchar2(50) Job_status	It gives information about the Job status to know whether it is in progress or closed or open or rescheduled.

- A Business **Schedules** one or more Jobs.[One to Many].
- A Business Schedules all requested Jobs by the End_User. So it Job is in **Total Participation.**
- An End_ user **Requests** one or more jobs.[One to Many].
- All the scheduled jobs are requested by the End_user.. So Job is in **Total Participation**.
- An End_user **Pays_for** one or more Jobs.[One to Many]
- All the requested Jobs are paid by End_user. So it is in **Total Participation**.

Entity: **Advertisement**	
Primary key attribute: Ad_id	
Attributes	Description
Varchar2 (500) Ad_content	It represents advertisement content of the business
Varchar2(25) Template_name	It denotes template name of the advertisement
Date Creation_date	It denotes creation date of the advertisement
Date Start_date	It denotes the date of advertisement that go live
Date End_date	It denotes the expiry date of the advertisement.
Int Last_updated_by	Represents Id of the User who update advertisement last

- A Business **Has** one or more Advertisements.[One to Many]
- All the Advertisements must belong to businesses. So Advertisement is in **Total Participation**

Description for mapping E-R diagram to Relational Schema:

1. **How you handled n:m relationships?**

   - The “Includes” relationship is handling the relationship between “Membership” and “Features”. In this relationship, the membership_id and feature_id are included as foreign keys.
   - The “Clicks” relationship is handling the relationship between “End_user” and “Business”. In this relationship, User_id and Business_id are included as foreign keys.
   - The “Connects” relationship is handling the relationship between “End_user” and “Business”. In this relationship, User_id and Business_id are included as foreign keys.
   - The “Belongs” relationship is handling the relationship between “Business” and “Category”. In this relationship, Business_id and Category_id are included as foreign keys.

2. **How you handled ternary relationships?**

   The “Member” relationship is handling the ternary relationship among “End_user”, “Business_user” and “Membership”. In this relationship, the membership_id and User_id are the foreign keys.

3. **How you handled recursive relationships?**

   The “Subcategory” relationship between “Category” and “Subcategory” is recursive. We chose to make the relationship recursive as the categories and subcategories follow BOM structure. In this relationship, Category_id is the foreign key.

4. **How have you map multi valued attributes to the Relational Model?**

   - “Favorites” is a multivalued attribute in “End_user” as users can have many favorite businesses.
   - “Service_Location” is a multivalued attribute in “Business” as business can provide service in multiple locations.
   - “Review_Images” is a multivalued attribute in “Review” as review can have many images associated with it.
   - “Social_Media” is a multivalued attribute in “Business” as Business can have multiple social media pages.

5. **How you handled inheritance?**

   - “User” is the entity set having disjoint subsets, “End_user” and “Business_user”. We use the attribute “Role_id” associated with Role from “Role” entity for identifying End_users and Business_users. We modeled this way based on our understanding from the Professor's lecture slides.

- Membership is an entity having overlapping subclasses, Basic, Standard and Premium. Initially, we had made it as ISA but after analysis we made it as an overlapping subclasses.

6. **Mention any constraints used in the project which cannot be implemented.**

   In the ER diagram, we have represented "Membership" as an overlapping inheritance, but we are unsure how to represent the individual subclasses in schema.

Problems faced during the design of E-R Diagram and its Mapping:

- We took a lot of time deciding on mapping total participation and cardinality.
- Deciding on whether we need to model some as an entity or multiple attributes. Example: Locations, Preferences.

Group Meeting Log:

Date	**Time**	**Participants**	**Activity**
1/22/2021	2:00-3:30 pm	Rancy Sangeetha Jyotsna	First Project meeting. We went through the requirements doc of "Who Does That" application and brainstormed potential ideas on how to develop the conceptual design. We agreed on sticking to just creating the DBMS and following the requirements outlined on the professor's website. We decided to review Yelp and Angie's list extensively before the next meetup.
1/25/2021	3:00-5:00 pm	Rancy Sangeetha Jyotsna	Date of the next meeting was decided. We listed project related questions to be asked in the class. Also, we decided to make further notes on the requirements for the next discussion. We also discussed the sample Specification document shared by the Professor. We discussed about relevance of map, request/messages, reviews
2/29/2021	2:00-3:15 pm	Rancy Sangeetha Jyotsna	We discussed various business rules we came up with, both from the specification document and the Professor's PowerPoint slide. We scheduled the next meeting to document requirements gathering and business rules.

2/8/2021	12:00-2:00 pm	Rancy Sangeetha Jyotsna	We all three came up with different Business rules. Then we did a brainstorming session to discuss the rules and professor's requirements and consolidated all the rules in one document.
2/8/2021	5:00-8:00 pm	Rancy Sangeetha Jyotsna	We met again to discuss and finalize the business rules and document the features of Yelp and Angie's list.
2/10/2021	1:30-2:30 pm	Rancy Sangeetha Jyotsna	Entities were identified and we came up with individual ER diagrams. We presented each other's ER diagram and discussed various variations. Based on our discussion we decided to update ER diagrams that we came up with.
2/12/2021	2:00-5:00 pm	Rancy Sangeetha Jyotsna	We reduced unimportant entities in the ER diagram and finalized ER diagrams. Also, we identified attributes for the identified entities.
2/22/2021	3:00-5:00 pm	Rancy Sangeetha Jyotsna	We finished drawing each entity and its attributes and also domains of all attributes. We discussed how we can reduce entities still if they have options to be included as attributes.
2/23/2021	2:00-4:00 pm	Rancy Sangeetha Jyotsna	We re-checked ER-diagram and fixed all cardinalities, we Created a relational Schema
2/24/2021	3:00-8:00 pm	Rancy Sangeetha Jyotsna	Finished the relational Schema Diagram. Constructed relational domain tables, added relational diagram domains and definitions, and reformatting the whole report
2/26/2021	1:00-3:30 pm	Rancy Sangeetha Jyotsna	Finalized relational domain tables
2/27/2021	3:00-5:00 pm	Rancy Sangeetha Jyotsna	Added description for mapping ER diagram. Reviewed the Project report and commented on potential mistakes.
2/28/2021	4:00-8:00 pm	Rancy Sangeetha Jyotsna	Reviewed the ER diagram and Relational schema for any gaps and updated the project report accordingly.

3/1/2021	10:00-12:00 pm	Rancy Sangeetha Jyotsna	Reviewed section of the project report before submission.