

Project Part 1 Group Members

Sai Pranav Datrika - A05187372

Yeshwanth Gowd Sreerama - A05166105

Karthik Reddy Thimmadi - A05173439

Requirement Gathering

- We initiated on working our methodologies by going through the project overview and the ideologies that we had acquired through gleaning from various websites like Angie's list and yelp. Professor had also explained regarding the project specifications during class hours.
- We had attentively read through the project description and highlighted the vital keywords that helped us in gathering the required entities and its attributes.
- Extra information for our project had been gathered from websites that helped us get a better understanding regarding the project specifications. The websites are listed below.

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

<https://www.constantcontact.com/small-business/advertising>

<https://.govalpak.com/where-to/advertise>

- We had also gained insights from an application called 'Rent Records' that one of our project mates (Sai Pranav Datrika) had modelled to understand on how to acquire the data for entities and building its ER diagram.
- We had also gained some practical knowledge on how a business advertising application(<https://www.yext.com>) works by directly interacting with one of the advertising- company and gained practical insights.
- We had conducted sessions physically and virtually to gather insights based on our experience and knowledge on understanding the project overview in detail.
- Each project mate had worked on building the entities and attributes individually and later gathered the data that are more in common and suits our best interest.
- We had also researched on how google search works for specific areas as the information panel for this application is similar to that of google information panel.
- Similar to how facebook search click works we had gotten an idea on how to model our **Customer_Clicks** entity as it provides the user with preferences based on his search history.
- Also similar to how google search engine cookies work, we had taken the ideology similar to it by using the **Recet-Access** entity.

Business Rules/Constraints

- Features are everchanging and must have the facility to add or remove features at given time.
- Any business can have multiple services in multiple locations with unique address and unique ID.
- Business in which the user prefers must have the priority from the usage of history of data.
- While searching for a business, the user should be shown all the list of subcategories of the business that it provides.
- The information in which the user creates while creating an account should be stored by the business for future purpose.
- •The information panel in which the user is shown should include all the required details (Image of the storefront, contact details, opening and closing hours etc).
- Access to connect-pay will be provided based on the user subscription preference.
- Each entity of the business and user side should consist a primary key which gets all the attributes related to that entity.
- The employees in who does that are acquired with super user if from which they will be able to access the required attributes for the provided service.
- Every Role in Who_Does_That_Employees must have a manager.
- If a customer or business want to access the who does that app features then they need to get approval from the employee of who does that app.
- Every business should at least provide one service for the customer.
- Who does that app should release features according to the business and customer level of subscriptions.
- Who Does that app should notify the business or customer when their subscription is about to end at least 7 days prior.
- Each Category of business should consist of sub categories.
- Who does that employees should not authorize the business to edit the reviews by the customers
- There should be a hierarchy in the level of subscriptions (Basic, Standard, Premium).

E-R Diagram

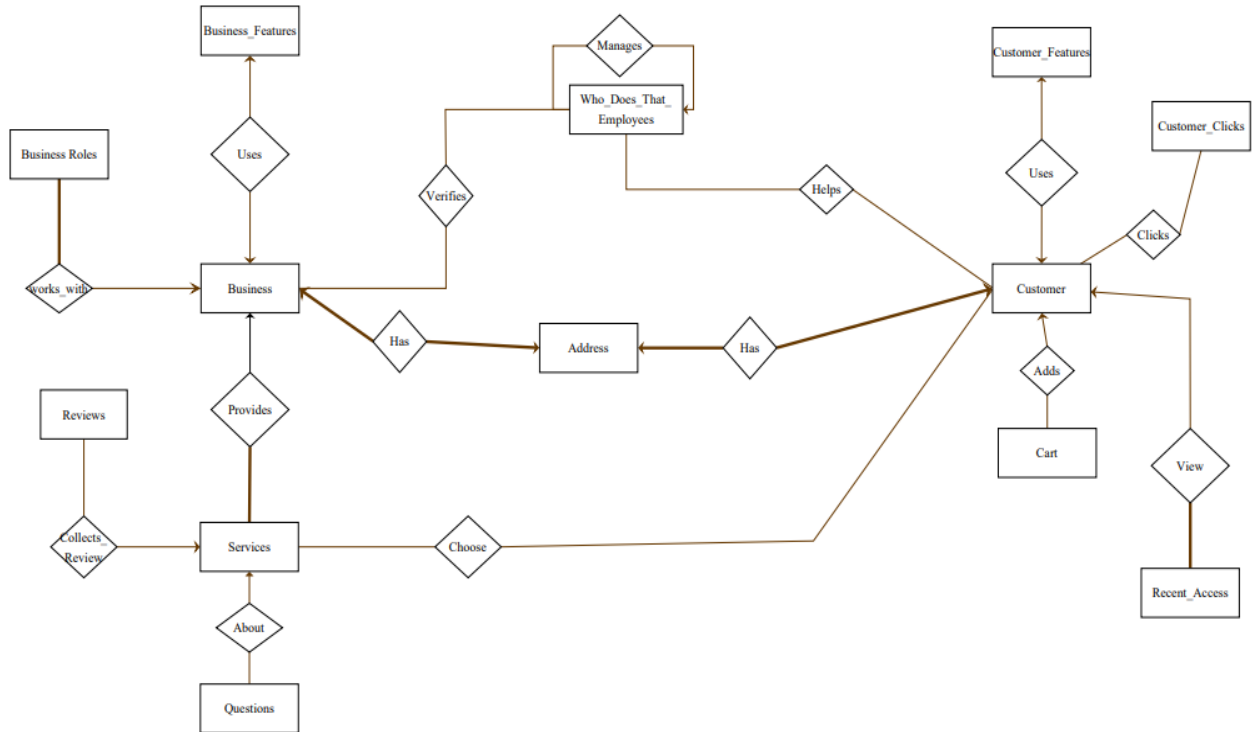
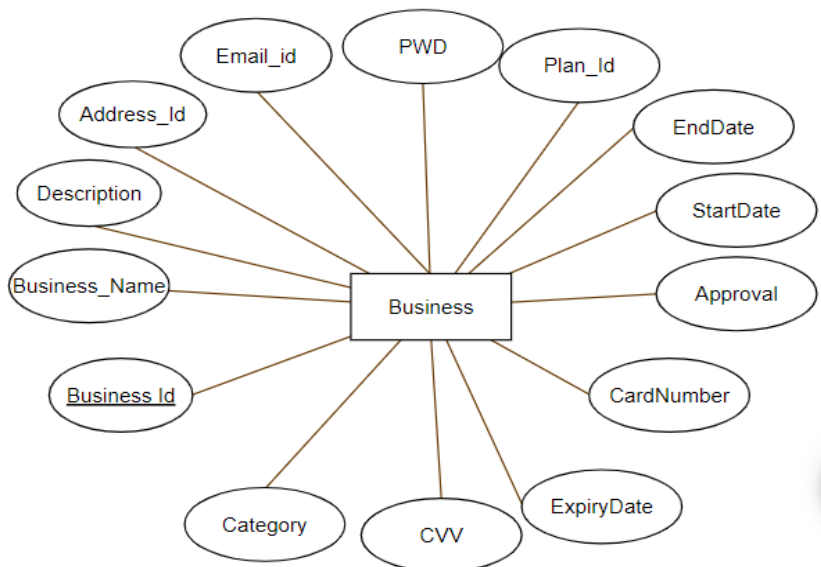


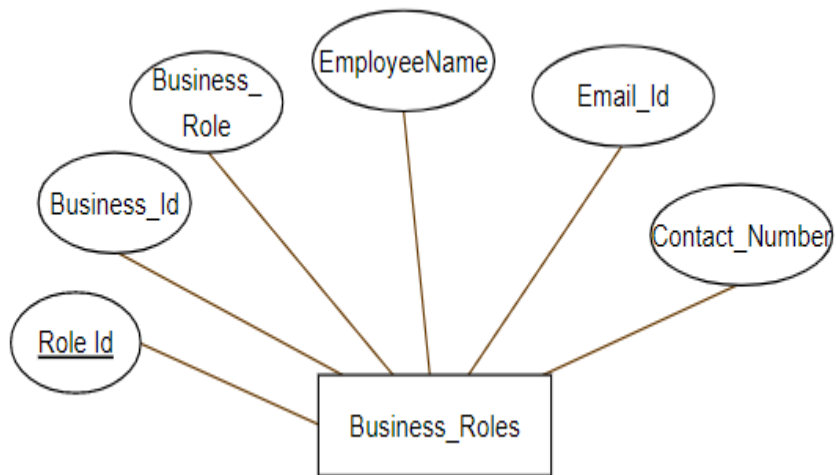
Fig 1: The Overall E-R Diagram for Who Does that

E-R Design Of Each Entity

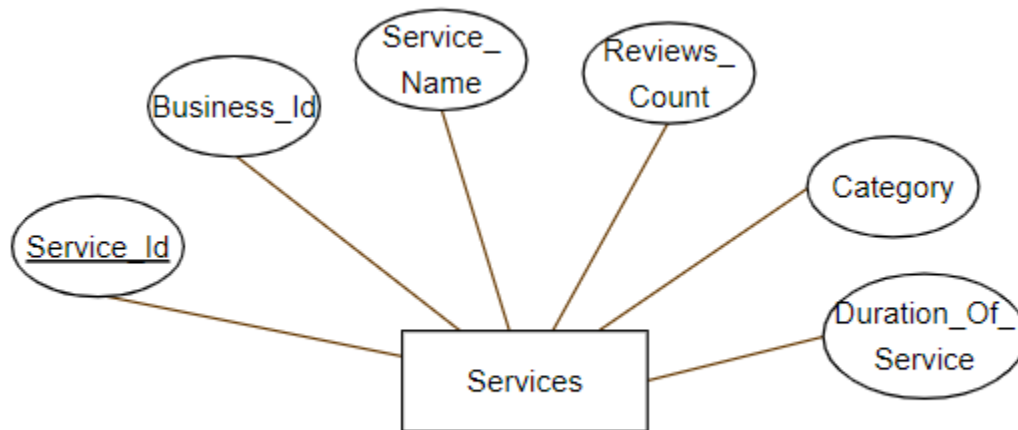
1. Business



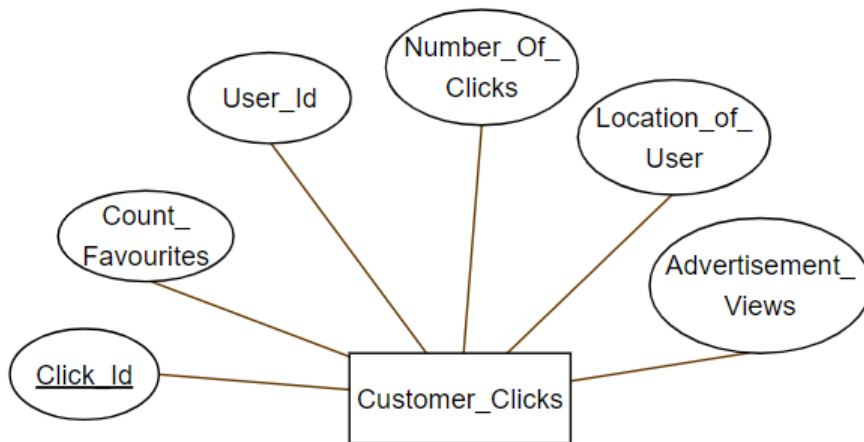
2. Business_Roles



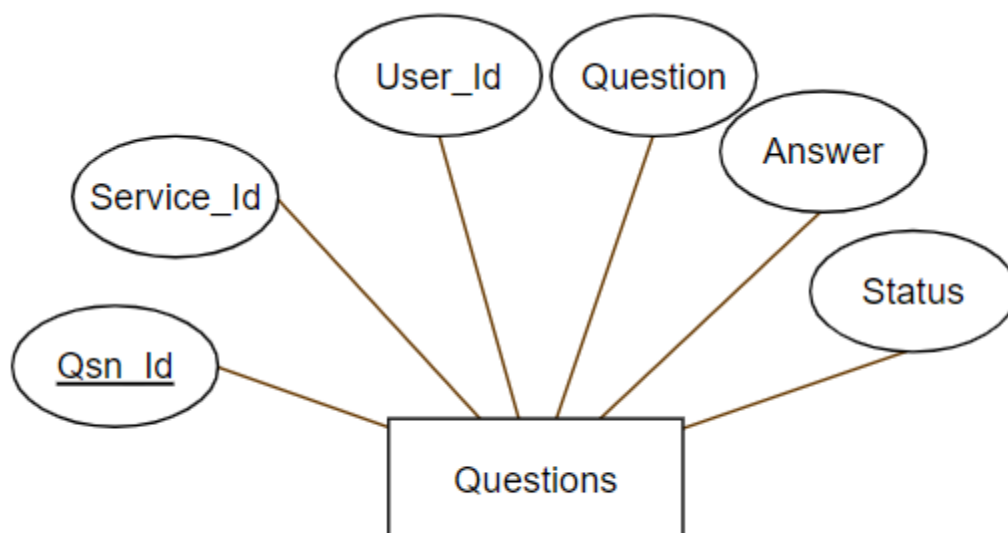
3. Services



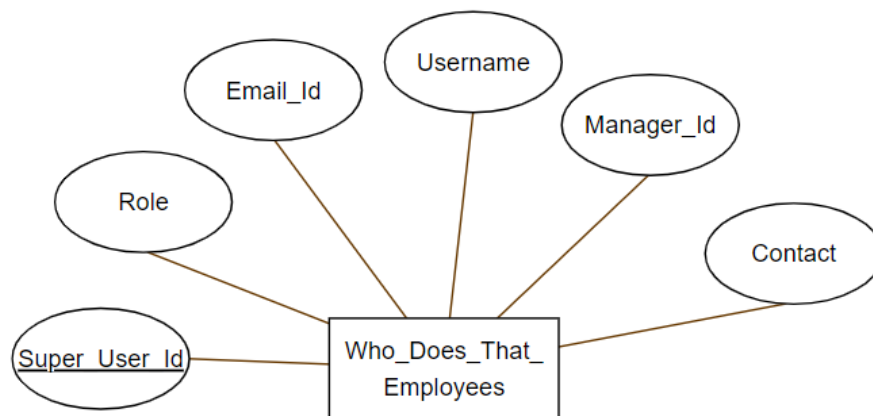
4. Customer_Clicks



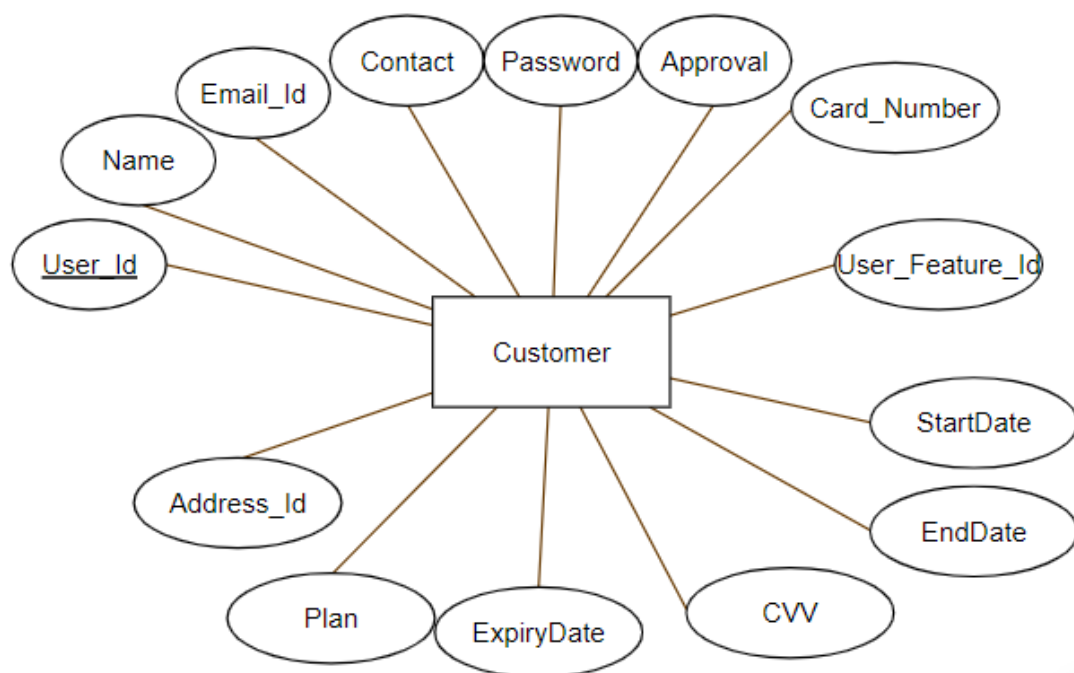
5. Questions



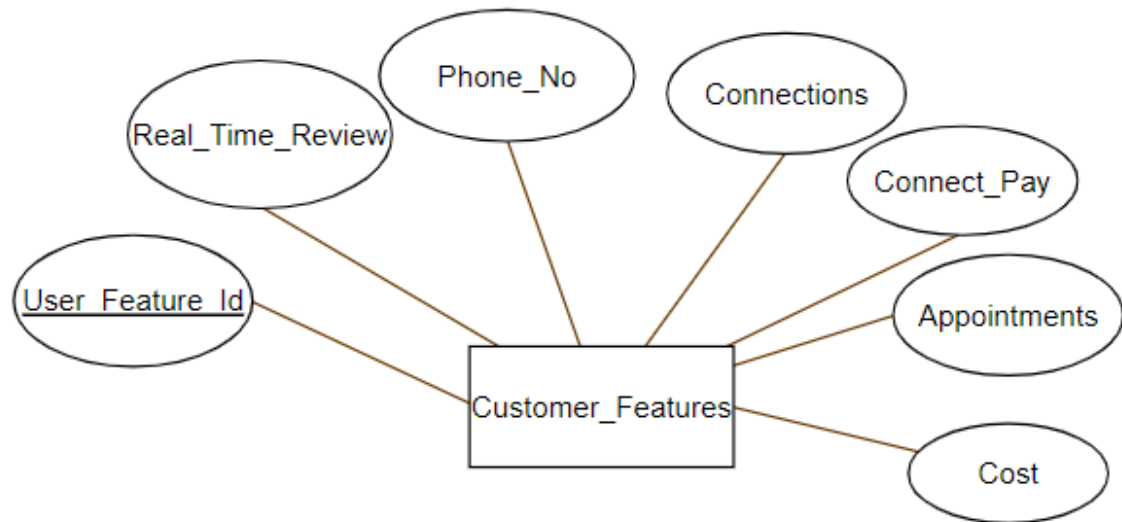
6. Who_Does_That_Employees



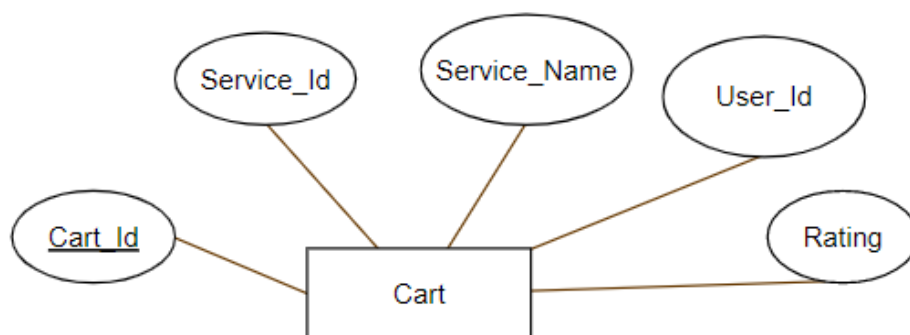
7. Customer



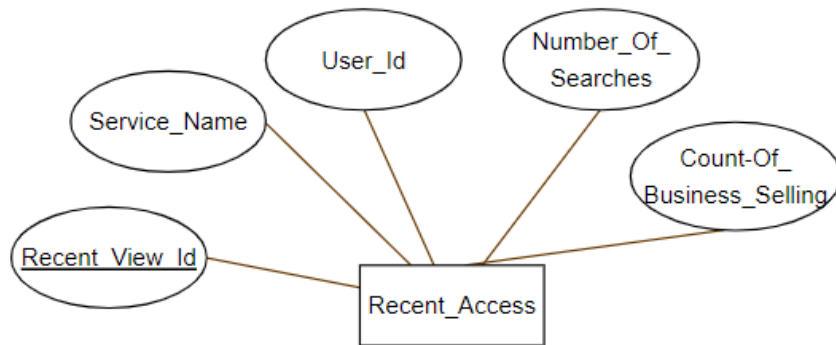
8. Customer_Features



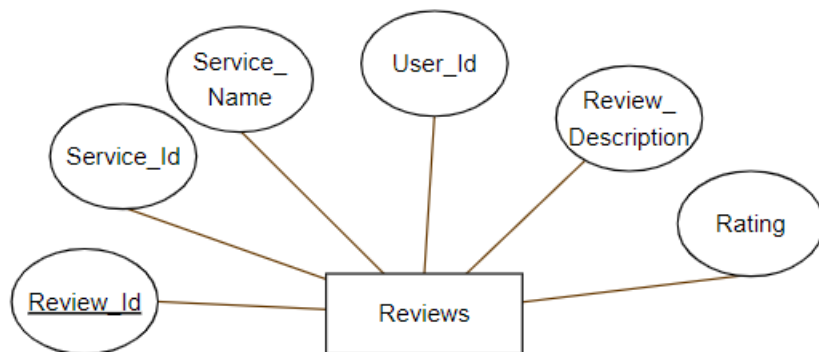
9. Cart



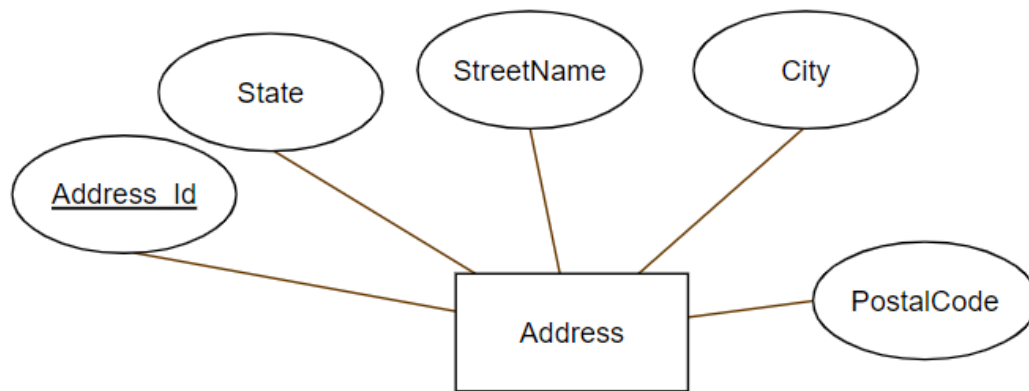
10. Recent_Access



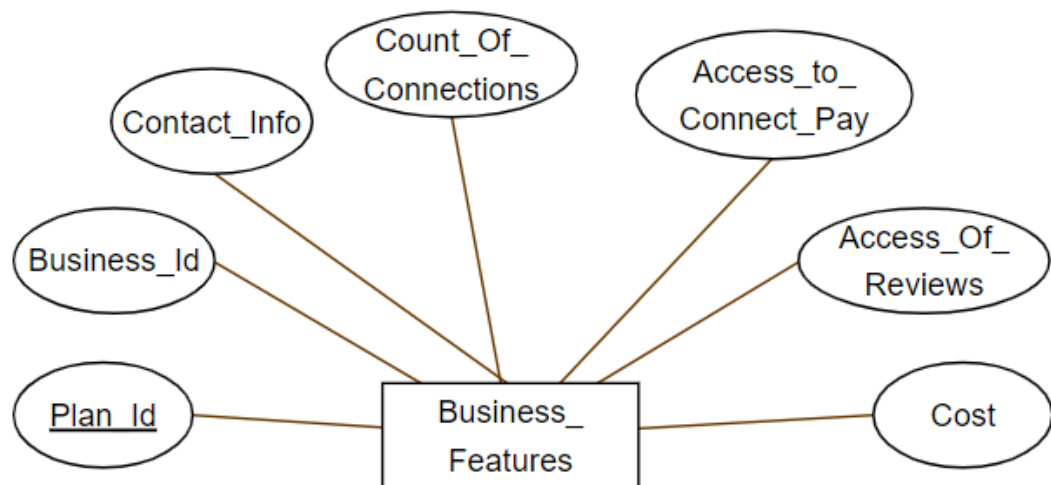
11.Reviews



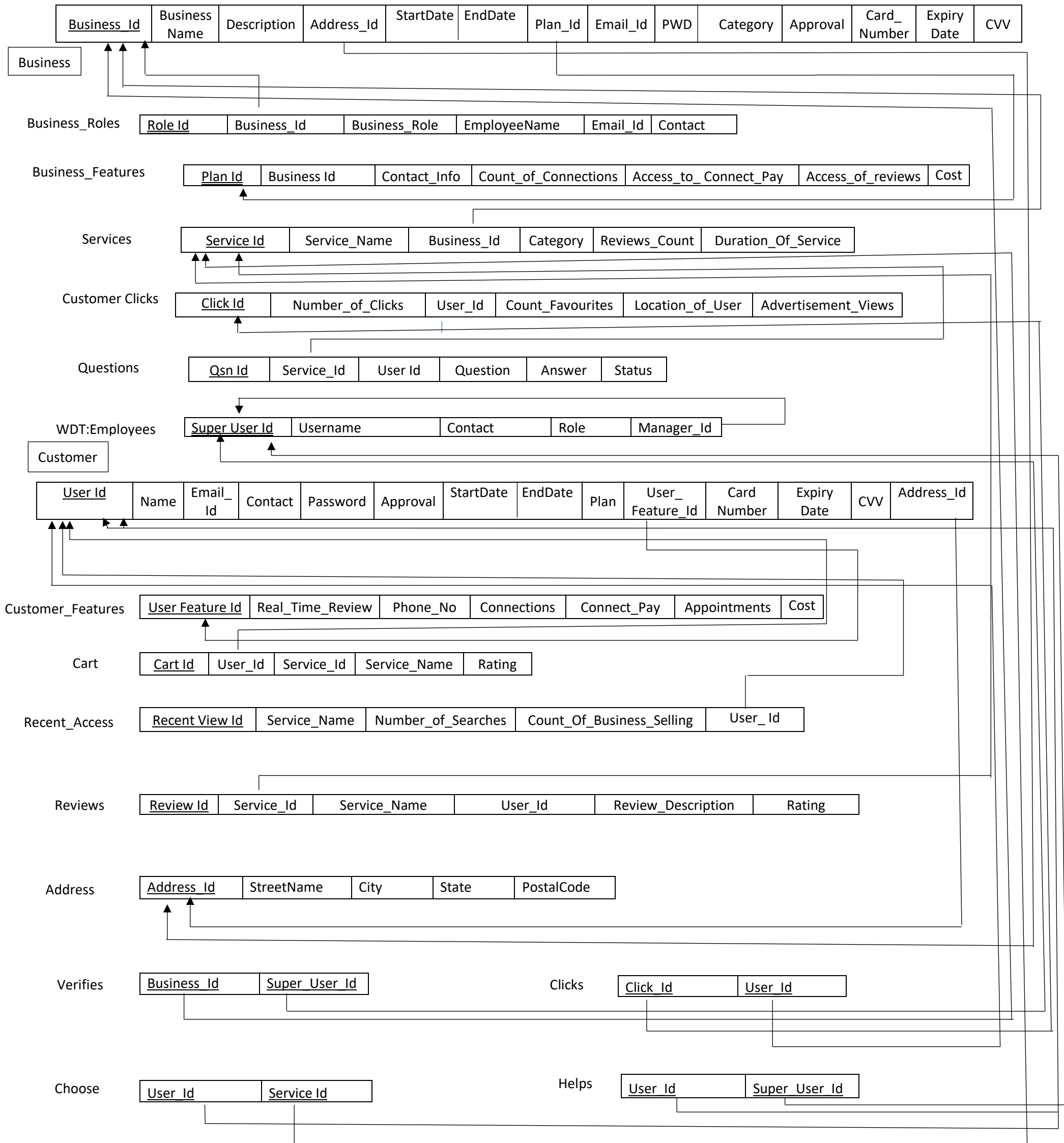
12. Address



13. Business_Features



Relational Schema Diagram



Relational Schema Definition and Domain of all attributes

Business(**Business_Id**, Business_Name, Description, Address_Id, StartDate, EndDate, Plan_Id, Email_Id, PWD, Category, Approval, Card_Number, ExpiryDate, CVV)

Business_Roles(**Role_Id**, Business_Id, Business_Role, EmployeeName, Email_Id, Contact)

Business_Features(**Plan_Id**, Business_Id, Contact_Info, Count_of_Connections, Access_to_Connect_Pay, Access_of_reviews, Cost)

Services(**Service_Id**, Service_Name, Business_Id, Category, Reviews_Count, Duration_Of_Service)

Customer_Clicks(**Click_Id**, Number_Of_Clicks, User_Id, Count_Favourites, Location_Of_User, Advertisement_Views)

Questions(**Qsn_Id**, Service_Id, User_Id, Question, Answer, Status)

Who_Does_That_Employees(**Super_User_Id**, User_Id, Username, Contact, Role, Manager_Id)

Customer(**User_Id**, Name, Email_Id, Contact, Password, Super_User_Id, StartDate, EndDate, Plan, user_Feature_Id, CardNumber, ExpiryDate, CVV, Address_Id)

Customer_Features(**User_Feature_Id**, Real_Time_Review, Phone_No, Connections, Connect_Pay, Appointments, Cost)

Cart(**Cart_Id**, User_Id, Service_Id, Service_Name, Rating)

Recent_Access(**Recent_View_Id**, Service_Name, Number_of_Searches, Count_Of_Business_Selling, User_Id)

Reviews(**Review_Id**, Service_Id, Service_Name, User_Id, Review_Description, Rating)

Address(**Address_Id**, StreetName, City, State, PostalCode)

Verifies(**Business**, **Who_Does_That_Employees**)

Clicks(**Customer**, **Customer_Clicks**)

Helps(**Customer**, **Who_Does_That_Employees**)

Choose(**Customer**, **Services**)

Domains Of all attributes

Business Attribute Types:

int Business Id	
String Business Name	//Represents name of the business <25 chars
String Description	// About the business <500 chars
int Address_id	//foreign key for Address entity
int Plan_Id	//foreign key for Business-Features Entity
String StartDate	//Represent Start Date of the Plan selected by the Business
String EndDate	//Represent EndDate of the Plan selected by the Business
String Email_Id	//Represent the common business email id <40 chars
String Category	//Represent the Business category in the Application <50 chars
String Approval	//Represents the approval Status of the business <20 chars
Long int Card_Number	//Represents the payment detail of the business
String Expiry Date	//Represents Card expiration <30chars
Int CVV	//Payment Details
String PWD	// Password Of the business

Business_Roles Types:

int Role_id	
int Business_id	//foreign Key for Business Entity
String Role	//Represents type of the role <50 chars
String Employee Name	//Represents name of the employee <70 chars
String Email_Id	//Represents the email id
String ContactNumber	//Represents the number of the employee

Business_Features Types:

int Plan_Id	
int Business_Id	//foreignKey for Business Entity
Boolean Contact Info	//Represents the User Contact Info
Boolean Count_Of_Connections	//Represents weather business can connect or not
Boolean Access_to_Connect_Pay	//Represent the access to connect pay.
Boolean Access_to_reviews	//Weather to provide review access or not
int Cost	// Represents the cost of the plan choose by business.

Services Types:

```
int Service_id
int Business_id           //foreign Key of the Business entity
String Service_name       //represents the name of the service <50 chars
String Category           //represents the sub-categories of the services <100chars
int Reviews_Count         //represents the count of the reviews
String Duration_of_Service //represents the time of the service <50chars
```

Reviews Types:

```
int Review_Id
int Service_Id           //Foreign Key of the Services Entity
String Service_Name      // Name of the Service <50Chars
int User_Id              // Storing the user Id to let business know who gave the review
```

Customer_Clicks

```
int Click_Id
int Number_Of_Clicks     //Count of clicks done by the customer
int User_Id              //Foreign Key of the Customer Entity
int Count_Favourites     //count of the Favourites by the customer
String Location_Of_user   //represents the location of user <50chars
int Advertisements_Views //represents customer surfed from advertisement.
```

Questions

```
int Qsn_id
int Service_Id           //foreign key for the service entity
int User_Id              //storing the user id to let business know who asked he query
String Question          //represent the questions asked by customer <1000 chars
String Answer            // represent the answers for the questions
String Status            // represent the status of the question
```

Recent Access Types:

```
int Recent_View_id
String Service_Name      // Name of the Service Searched by the User.
int Number_Of_Searches   //Count of searches related to the service name
int Count_Of_Business    //Count Of Business providing the Services
int User_Id              //Foreign Key Of the Customer entity
```

Customer Types:

int User_Id	
String Name	//Represents Customer Name <50 chars
String Email_Id	// Represents email id of the customer <50 chars
String Contact	//Phone number of the customer<50 chars
String Password	// Password of the Customer <50chars
int Super_User_Id	// Foreign Key of the WDT Employee
String StartDate	//represent start date of the Customer
String EndDate	//represent enddate of the Customer
String Plan	//represent the Selected Plan <50 Chars
int User_Feature_Id	//foreign Entity of the User-Feature Entity
Long int Card_Number	//Represents the payment detail of the customer
String Expiry Date	//Represents Card expiration <30chars
int CVV	//Payment Details
int Address_Id	//foreign key of the Address entity

Who_Does_That_Employees Types:

int Super_User_Id	
int User_Id	//Foreign Key Of the Customer Entity
String UserName	//Username of the Who Does that Employee <50chars
String Contact	//Represent the Customer Contact
String Role	// Represent Type of the admin role
int Manager_Id	//Recurrence foreign key

Customer_Features Types:

int User_feature_id	
Boolean Real Time Review	//Customer will take the feature or not
Boolean Phone_No	//Customer will take the feature or not
Boolean Connecions	//Customer will take the feature or not
Boolean Connect_Pay	//Customer will take the feature or not
Boolean Appointments	//Customer will take the feature or not
double Cost	//Customer membership plan cost

Cart Types:

int Cart_Id	
int User_Id	//foreign Key of the Customer Entity
int Service_Id	// Storing the service id of the customer favourites
String Service_Name	//Name of the Service <50 chars
double Rating	// rating the service provided by the customer

Address Types:

int Address_Id
String Street_Name //represent the street <50 Chars
String City // represent City <50 chars
String State // represent State <50 chars
int Postal_Code //represents area code

Verifies Types:

int Business_Id //foreign Key of the Business Entity
Int Super_User_Id //foreign Key Of the Who_Does_that_employee entity

Selected Types:

int User_Id //foreign Key Of the Customer Entity
int Service_Id //foreign Key Of the Services Entity

Clicks Types:

int Click_Id // foreign Key of the Customer_Click's Entity
int User_Id // foreign Key Of the Customer Entity

Helps Types:

int User_Id //foreign Key Of the Customer Entity
int Super_User_Id //foreign Key Of the Who_Does_That_Employee entity

Design Decision and Domain Of all Attributes (For E-R Diagram)

Entity: Business	
Primary Key Attribute: Business_Id	
Attributes	Description
String Business Name [< 50 chars]	Represents business' name.
String Description [< 50 chars]	To define about services provided by a business. (For ex: automobile services, cleaning services etc.)
int Address_id	Foreign key of address entity.
Date Start Date	This attribute stores the subscription start date of businesses.
Date End Date	This attribute stores the subscription start date of businesses.
int plan_id	This attribute stores the Id of business subscribed plan.
String Email [< 150 chars. xyz@gmail.com]	This attribute stores the Email of the business.
String Category [< 50 chars]	Type of the category a business belongs too.
String Approval [< 10 chars]	Status of approval from the Employees of Who Does That Employees
String CardNumber [< 16 chars]	To store the card information of businesses. So that businesses can pay for subscriptions to who do
Date ExpiryDate	

int CVV	To store the card information of businesses. So that businesses can pay for subscriptions to who does that app team.es that app team.
<ul style="list-style-type: none"> Business Uses business features. [Many to One]. A business can use only one business feature plan at-a-time when approval comes from who does that app team, then only a business can be able to acquire business features. So business features is in Partial Participation and business is in Partial Participation. Business Works_with Business Roles. [One to Many] Every business role is associated with one business. So, it is in Total Participation. But businesses may or may not have business roles. (As even one person can run a small-scale business). So business is in Partial Participation. Business Provides Services. [One to Many] A single business can provide many services. So business is in Partial Participation and Services are in Total Participation. Business Has Address [One to One] Both are in Total Participation. Business gets Verified by Who does that app employees. [Many to Many] Every business should be verified by who does that employees to do business by obtaining business features. So, both are in Partial Participation with each other. 	

Entity: Business Role	
Primary Key Attribute: Role_id	
Attributes	Description
int Business_id	It is a Foreign key of Business
String Business_role [< 150 chars]	To describe the Business Role of employees working under a business.

String EmployeeName	To Store the information of the Employee Working in a business
[< 150 chars]	
String Email	
[< 150 chars. xyz@gmail.com]	
String Contact	
[< 15 chars (123)-456-7890]	
<ul style="list-style-type: none">Business_Roles entity has Many to One relationship with the Business Entity.	

Entity: Business_Features	
Primary Key Attribute: Plan_Id	
Attributes	Description
String Business_Name [< 50 chars]	This attribute stores the business names which are enrolled in subscription.
Boolean Contact_Info [True or False value]	<ul style="list-style-type: none"> This attribute provides whether a business has enabled contact info feature or not. Contact info feature enable businesses to view their customers information.
Boolean Access_to_Connect_Pay [True or False value]	<ul style="list-style-type: none"> This attribute provides whether a business has enabled contact info feature or not. Contact info feature enable businesses to view their customers information.
Boolean Access_to_Reviews [True or False value]	<ul style="list-style-type: none"> This attribute provides whether a business has enabled Access to reviews info feature or not. Access to Reviews feature enable businesses to view reviews of customers to their services.
Double Cost [Units are in Dollars and Cents.]	This attribute stores the total price of the features

- Business Features are used by business. [One to One]

Businesses may or may not use Business features. So, both are in **Partial Participation**.

Entity: Services

Primary key Attribute: Service_Id

Attributes	Description
String Service_Name [< 50 chars]	Represents the type of service provided by a business.
int Business_Id	Foreign key of the business entity
String Category [< 50 chars]	It defines under which category a service belongs to.
int Reviews_Count	Represents total count of the reviews given by customers for a particular service.
String Duration_Of_service [< 50 chars]	Represents the time period of the service provided by business to its customers
<ul style="list-style-type: none"> • Services About Questions. [One to Many]. Services enable businesses to answer the questions asked by the customers. • Services Collect Reviews posted by customers. 	

Entity: Questions	
Primary key Attribute: Qsn_id	
Attributes	Description
int Service_id	This attribute is a foreign key of service entity.
int user_id	This attribute stores the id of customers
String Question [< 1000 chars]	This attribute stores the questions of customers to various services.
String Answer [< 1000 chars]	This attribute stores the answers provided by businesses to customers.
String Status [< 50 chars]	This attribute stores the status of whether a business had answered or unanswered customer questions.
<ul style="list-style-type: none"> Questions entity has Many to One relationship with services. 	

Entity: Reviews	
Primary key Attribute: Review_Id	
Attributes	Description
int Service_Id	Foreign key of service entity
String Service_Name [<150 chars]	This attribute stores customer reviewed service names.
int User_Id	To fetch the data of customers who gave reviews.

String Review_Description [<1000 chars]	This attribute stores the reviews of customers given to various services.
double Rating	This attribute stores the rating given by the customer for a service.
<ul style="list-style-type: none"> Reviews entity have Many to One Relationship with the Services entity 	

Entity: Who_Does_That_Employees	
Primary key Attribute: Super_User_Id	
Attributes	Description
String Username [< 50 chars]	Represents the username of who does that app employees.
String Contact [< 15 chars (123)-456-7890]	Represents phone numbers of who does that app employees
String Role [< 50 chars]	This represents the role of employees who are working under who does that app
int Manager_Id	It is a foreign key of who does that app employees who's role is manager.
<ul style="list-style-type: none"> Who_Does_That_Employees Manages Who_Does_That_Employees (Recursive Relationship). [One to Many] Who does that app employees manages who does app employees at different hierarchy levels like manager, junior manager, entry level employee etc. So, Who does that app employees have Partial Participation. Who does that app employees verify and approves businesses and customers to be on their platform. 	

Entity: Address	
Primary key Attribute: Address_id	
Attributes	Description
String Street_Name [< 150 chars]	To store the personal information of customers and business on who does that app platform.
String City [<150 chars]	
String State [<150 chars]	
int Postal Code	
<ul style="list-style-type: none">Address has One to One Relationship with both the Customer entity and Business entity.	

Entity: Customer	
Primary key Attribute: User_Id	
Attributes	Description
String Name [< 150 chars]	To keep the record of personal information of each customer.
String Email_Id [< 150 chars. xyz@gmail.com]	
String Contact [< 15 chars (123)-456-7890]	
String Password [< 50 chars]	

Date StartDate	This attribute stores the subscription start date of customers.
Date EndDate	This attribute stores the subscription end date of customers.
String Plan [< 50 chars]	This attribute represents the plan name that the customer is on.
int User_Feature_Id	Foreign key of customer feature entity
Long int CardNumber	It stores the card information of customers for future purchases.
Date ExpiryDate	
int CVV	
int Address_Id	Foreign key of address entity
<ul style="list-style-type: none"> Customer Uses Customer Features. [Many to One]. A customer can use only one customer feature plan at-a-time when approval comes from who does that app team, then only a customer can be able to acquire customer features. So customer features are in Partial Participation and customer is in Total Participation. Customer Adds to Carts. [One to Many]. A customer may or may not add services to their cart. So, both are in Partial Participation. Customer Clicks Customer_Clicks. (Customer click count of services) [Many to Many] Every customer may or may not click the services provided by various businesses. So, both are in Partial Participation. Customer view Recent access. (Customer can view their recently searched services). [One to Many] A customer can view all their respective recent searches of various services while searching. So, customer is in Partial Participation and recent access is in Total Participation. 	

- Customer gets **Help** from who_does_that_app_employees. [Many to Many]
A customer may or may not request help from who does that app team. So, both are in **Partial Participation**.
- Customer **Has** an Address [One to One]
Both are in **Total Participation**.
- Customer **Chose** a Services. [Many to Many].
Every customer may or may not select a service provided by businesses. So, both are in **Partial Participation**.

Entity: Customer_Features	
Primary key Attribute: User_Feature_Id	
Attributes	Description
Boolean Real_Time_Reviews [True or False value]	<ul style="list-style-type: none"> • This attribute provides whether a business has enabled real time reviews feature or not. • Real time review feature provides latest reviews given by customers.
<ul style="list-style-type: none"> • Boolean Phone_no • [True or False value] 	<ul style="list-style-type: none"> • This attribute provides whether a business has enabled Phone number feature or not. • Phone_no feature provides the contact info of businesses to customers
Boolean Connections [True or False value]	<ul style="list-style-type: none"> • This attribute provides whether a business has enabled connection feature or not. • Connections feature enable customers to view/follow businesses via social media links
Boolean Connect_Pay [True or False value]	<ul style="list-style-type: none"> • This attribute provides whether a business has enabled connect pay feature or not. • Connect pay feature provides the ability to do payments by using connect pay app.

Boolean Appointments [True or False value]	<ul style="list-style-type: none"> This attribute provides whether a business has enabled appointments feature or not. Appointments feature enable the customers to interact and schedule appointments with businesses
Double Cost	This attribute provides the total price of the features selected by customers. (Basic, Standard Premium).
<ul style="list-style-type: none"> Customer features has One to One relationship with Customer entity. 	

Entity: Cart	
Primary key Attribute: Cart_id	
Attributes	Description
int User-id	Foreign key of the customer entity
int Service_id	To keep track of the service Id's that customer adds to their cart.
String Service-Name [<150 chars]	This attribute stores the name of the services in customers cart
Double Rating	This attribute provides the rating of services that are in customer's cart.
<ul style="list-style-type: none"> Cart entity is in Many to One relationship with customer entity. 	

Entity: Recent_Access	
Primary key Attribute: Recent_Review_Id	
Attributes	Description
int User_Id	Foreign key of customer entity.
String Service_Name [<150 chars]	This attribute stores recently accessed service names of customers.
int Number_Of_Searches	This attribute stores the count of customer searches for a particular service.
int Count_ of_Business_Selling	This attribute represents the count of businesses that are providing the recently searched services of customers.
<ul style="list-style-type: none"> •This entity is a business feature that keeps track of the recent searches of customers so that who does that app can sell this feature to businesses which opt for this feature. •This entity is beneficial to businesses to improve services provided by them. •This entity is in Many to One relationship with customer entity 	

Entity: Customer_Clicks	
Primary key Attribute: Click_Id	
Attributes	Description
int Number_of_clicks	It provides the click count of customer visited businesses on the platform.
int User_Id	To store the Id of customers.
int Count_Favourites	It provides count of how many services a customer had clicked favorite.
String Location_Of_User [< 50 chars]	This attribute provides the login location of the user

int Advertisement_Views	This attribute gives the count of customer visits to businesses via advertisements
<ul style="list-style-type: none"> • This entity is utilized as a premium feature for businesses to obtain customer app activity. So that businesses can analyze and increase their reachability to customers. • Customer_clicks is in Many to Many relationships with customer entity. 	

Description for mapping E-R Diagram to Relational Schema Diagram

1.How you handled n:m relationships?

- The “Verifies” Relationship between Who_Does_That_Employees entity and Business entity is handled by adding a relationship entity “Verfies”. In this entity, we have included primary key of business: Business_Id and primary key of Who_Does_That_Employees: Super_User_Id as foreign keys.
- The “Choose” Relationship between Services and Customer is a Many to Many Relation, it is handled by adding a relationship entity “Choose”. In this entity, we have included primary key of Services: service_Id and primary key of Customer entity: User_Id as foreign keys.
- The “Helps” Relationship between Who_Does_That_Employees entity and Customer entity is handled by adding a relationship entity “Helps”. In this entity, we have included the primary keys of the both entities, User_Id and Super_User_Id as foreign keys
- The “Clicks” Relationship between Customer entity and Customer_Click’s entity is handled by adding a relationship entity “Clicks”. In this entity, we have included primary key of Customer_Clicks: Click Id and primary keys of Customer: User_Id as foreign keys.

2.How you handled 1:m relationships?

- The “works with” relationship between Business_Roles entity and Business entity is a Many to One relationship, so we have used optimized mapping version, where we have included primary key of Business entity: Business_Id in the Business_Roles entity as an attribute, and used this attribute in Business_Roles as a foreign key.
- The “Provides” relationship between Business entity and Services entity is a One to Many relationship,so we have used optimized relational mapping version, where we have included primary key of Business entity: Business_Id in Services as an attribute, used this attribute in Services as a foreign key.
- The “Collects_Review” relationship between Reviews entity and Services entity is a Many to One relationship, so we have used optimized mapping version,where we have included the primary key of Services entity: Service_Id in the Reviews Entity as an attribute and used this as a foreign key.
- The “About” relationship between Questions entity and Services entity is a Many to One relationship, so we have used optimized mapping version, where we have included the primary key of Services entity: Service_Id in the Questions entity as an attribute and used this as a foreign key.
- The “Adds” relationship between Cart entity and Customer entity is a Many to One relationship, so we have used optimized relational mapping version, where we have included the primary key of the Customer entity: User_Id in the Cart entity as an attribute and used this as a foreign key.

- The “View” relationship between Recent_Access entity and Customer entity is a Many to One relationship, so we have used optimized relational mapping version, where we have included the primary key of the Customer entity: User_Id in the Recent_Access as an attribute and used this as a foreign key.

3. How you handled 1:1 relationships?

- The “Uses” relationship between Business entity and Business_Features is an One to One relationship, so we have included the primary key of Business_Features entity: Plan_Id in Business entity as an attribute and used this as a foreign key.
- The “Has” relationship between Business entity and Address is an One to One relationship, so we have included the primary key of Address entity: Address_Id in Business entity as an attribute and used this as a foreign key.
- The “Uses” relationship between Customer entity and Customer_Features is an One to One relationship, so we have included the primary key of Customer_Features entity: User_Feature_Id in Customer entity as an attribute and used this as a foreign key.
- The “Has” relationship between Customer entity and Address is an One to One relationship, so we have included the primary key of Address entity: Address_Id in Customer entity as an attribute and used this as a foreign key.

4.How you handled recursive relationship?

- The Manages relationship is recursive relation in Who_Does_That_Employees, where the employees of the who does that application are mapped to their managers. In this relationship, we have included a Manager_Id attribute in the Who_Does_That_Employees entity where it acts as a foreign key to that entity itself. So whenever an employee is added it also holds the Manager_Id, which is nothing but the Super_User_Id of the Manager in the Who_Does_That_Employees entity.

5.How you handled ternary relationships?

- We haven’t used any ternary relationships in ER diagram.

6.How you handled Multi Value Attributes in relational Schema?

- We haven’t used any Multi Value Attributes in our entities.

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

- For ER Diagram, defining the relationships between the entities was a difficult task for us.
- Also, the cardinality and participation of the entities was bit time consuming task for us, we have spent more than two days on ER diagram to final the diagram.
- In the initial stages of our ER Diagram, we were getting many ternary relationships ideas but after listening to the professor's lectures we have come to a point that it is an inefficient relationship. So, we have spent lot of time in understanding concepts and read many articles to reduce our ER Diagram from ternary to binary relationships and how defining an attribute changes processing time of the queries and many more.
- For Relational Schema Diagram, defining the foreign keys and connecting it to the correct entity was bit confusing and difficult task. And also maintaining the notations as per ER was bit challenging and we tried our best.

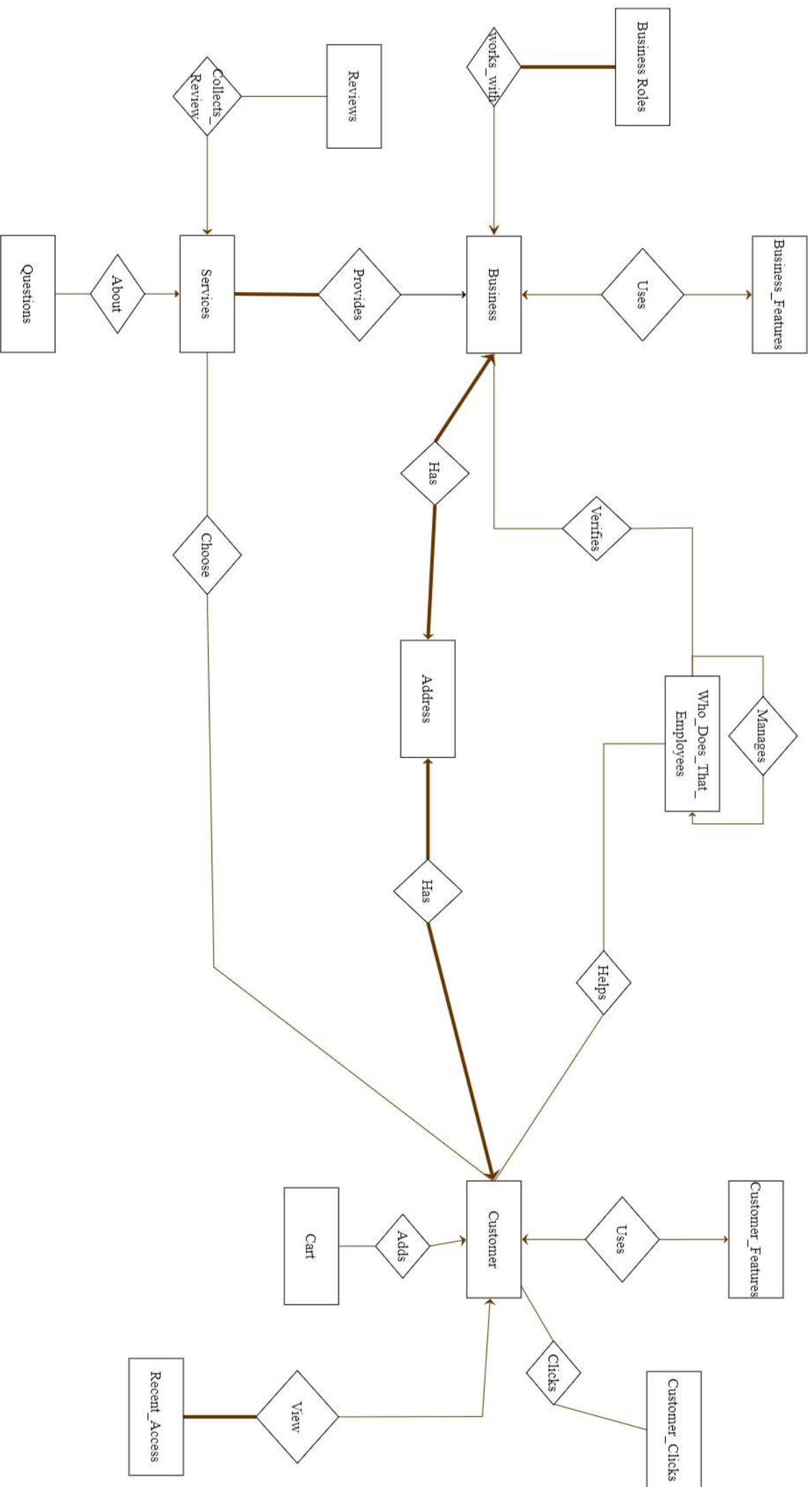
Group Meeting Log

Date	Time	Participants	Activity
08/31/2022	8-11pm	Sai Pranav Datrika Yeshwanth Gowd Sreerama Karthik Reddy Thimmadi	First project meeting. Discussed the overall Project Specification and tried to gather the ideas from each other. Identified the basic requirements for the projects.
09/05/2022	1:30-3:30pm	Sai Pranav Datrika Yeshwanth Gowd Sreerama Karthik Reddy Thimmadi	We have gathered the information from other apps like yelp and angel list. Then, we have discussed some vital points related to the business rules. And figured out few entities to include in our ER diagram.
09/09/2022	6:00-9:30pm	Sai Pranav Datrika Yeshwanth Gowd Sreerama Karthik Reddy Thimmadi	Each of us have gathered around 6 business rules for our project. Then, we have selected best amongst them and finalized the business rules. Then, we have listed 20 entity names for our ER diagram.
09/11/2022	9:00-2:00pm	Sai Pranav Datrika Yeshwanth Gowd Sreerama Karthik Reddy Thimmadi	We discussed and finalized 13 entities that needs to be included on ER diagram. After that, each of us spent on 4 entities to obtain attributes. Then, we discussed and corrected our mistakes in the attribute names and came to a conclusion on the attribute names.
09/16/2022	1:00-3:00pm	Sai Pranav Datrika Yeshwanth Gowd Sreerama Karthik Reddy Thimmadi	Sai Pranav started working on Business and Business_Features entities and relationships between them. Yeshwanth started working on Services and Who_Does_That-Employess entities and relationships between them.

			<p>Karthik started working on Customer and Customer_Features entities and relationships between them.</p> <p>At the end of the meeting, we all have discussed how these entities must be related with each other and concluded these entities position in the ER Diagram.</p>
09/18/2022	1:30-4:30pm	<p>Sai Pranav Datrika</p> <p>Yeshwanth Gowd Sreerama</p> <p>Karthik Reddy Thimmadi</p>	<p>Sai Pranav started working on the Business_Roles and Customer_Clicks entities and relationship among them in the ER Diagram.</p> <p>Yeshwanth Started working on the Reviews and Questions entities and relationship among them in the ER Diagram.</p> <p>Karthik started working on the Address and Cart entities and relationship among them in the ER Diagram.</p> <p>At the end we have finalized the ER diagram and documented all the entities that are required for our project.</p>
09/20/2022	2:30-4:30pm	<p>Sai Pranav Datrika</p> <p>Yeshwanth Gowd Sreerama</p> <p>Karthik Reddy Thimmadi</p>	<p>Sai Pranav has presented the ER Diagram in the zoom meeting, Yeshwanth and Karthik were analyzing the ER Diagram and made necessary changes to the ER diagram.</p> <p>Then, we started working on the ER to relational mapping for all the entities. And we went through the class notes given by the professor for relational mapping.</p> <p>Based on that we all have understood the concepts and started working on relational schema mapping.</p>

09/22/2022	9:00-1:00pm	<p>Sai Pranav Datrika</p> <p>Yeshwanth Gowd Sreerama</p> <p>Karthik Reddy Thimmadi</p>	<p>Yeshwanth Started mapping the Business and Business_Roles features with the foregin keys.</p> <p>Sai Pranav Started working on Who_Does_That_Employees and Services entities relational mapping with the foreign keys.</p> <p>Karthik Started working on the Customer and Customer_Features entities relational mapping with the foreign keys.</p> <p>After that, we have concluded the major entites relational mapping in the ER diagram. Then, we started working on small relational mapping together.</p>
09/24/2022	9:00-11:30pm	<p>Sai Pranav Datrika</p> <p>Yeshwanth Gowd Sreerama</p> <p>Karthik Reddy Thimmadi</p>	<p>Sai Pranav presented the flow of the final version of relational scchema and we discussed on any necessary changes that should be done. And Finalized the relational schema diagram and documented the relational schema</p>
09/26/2022	8:00-2:00pm	<p>Sai Pranav Datrika</p> <p>Yeshwanth Gowd Sreerama</p> <p>Karthik Reddy Thimmadi</p>	<p>We began the meeting by reviewing all the sections of the project such as ER diagram, Entities and relational schema, requirements gathering and business policies.</p> <p>Sai Pranav and Yeshwanth started working on documenting the entity names. Karthik assisted us in proof reading the names in the documents.</p>
09/28//2022	3:00-5:00pm	<p>Sai Pranav Datrika</p> <p>Yeshwanth Gowd Sreerama</p> <p>Karthik Reddy Thimmadi</p>	<p>Final Project meeting:</p> <p>We reviewed the final version of the document and made few required changes to the documents such as formatting and proof reading the whole document.</p>

			We edited the document and checked the documents for multiple times.
09/29/2022	7:30-9:30pm	Sai Pranav Datrika Yeshwanth Gowd Sreerama Karthik Reddy Thimmadi	We all got together and again reviewed the final document and Sai Pranav has submitted the project on Canvas.



Relational Schema Diagram

