EzRental Auto Rental POS Management System

Darwhin Gomez

NYCCT Software Solutions

300 Jay St

Brooklyn New York

Contents

EZRental Auto Rental POS Management System Database Design and Implementation

Executive Summary

The EZRental Applications and Database are designed to be flexible, well-functioning, and user-friendly for both front-end users and data maintenance personnel. The design and implementation process is diligently presented in the following sections.

- Project Statement & Objectives
- Project Management Methodology
- Application Business Requirements
- Application Development & Technical Requirements
- Application Physical & Software Technical Architecture
- Application Physical Architecture Overview
- Application Development Feature & Functionality (Agile Backlog)
- Database Management System Development Environment & Physical Architecture
- Project Roles & Responsibilities
- Entity Relational Conceptual Model
- Normalized Logical Model Diagram
- Physical Model Data Diagram
- Physical Model Schema Design Diagram
- Development & Implementation
- Implemented Physical Schema Diagram
- Database Validation Testing

Project Statement & Objectives

EZRental has requested the design and implementation of the following:

1) **EZRental Point-of-Sales (POS) system** intended for ***Customer Service Representative*** and other ***employees*** in the ***rental agencies***, such as Maintenance Personnel, Vehicle Inventory Team, Transport Drivers etc.

2) A **Corporate INTRANET Website** named **EZRentalCorp.com** intendedfor business employees in the corporate offices, and Rental Agencies,

3) an **e-commerce INTERNET Website** name **EZRental.com** intended for customers to make and manage reservations via the public internet.

The EZRental Auto Rental Management System features are designed to:

- Allow customers, both retail and corporate customers, to reserve vehicles for renting, like other in-person or online car rental systems such as **Avis, Hertz, Budget,** etc.
- The application needs to provide the required functionalities for our Customer Service representatives and other front-line workers in our rental agencies to service in-person customers for renting and reservation processes.
- Provide the features for our business users in our corporate offices who need to create reports, perform analytics and other business functionalities related to the management of the reservations and rental of our vehicles, via our INTRENET PORTAL.
- Finally, features that allow customers to make & manage vehicle reservations, profile, account etc., via the public internet.

- The application is designed to support dozens of major cities around the world. In addition, it provides a great user experience both in the rental agencies as well as the online systems with the best competitive pricing available in the market.
- The company currently has rental agency branches in US, Canada, Mexico, United Kingdom, Japan & Australia and looking to expand further globally into other markets in Asia, Africa, and the Mediterranean.

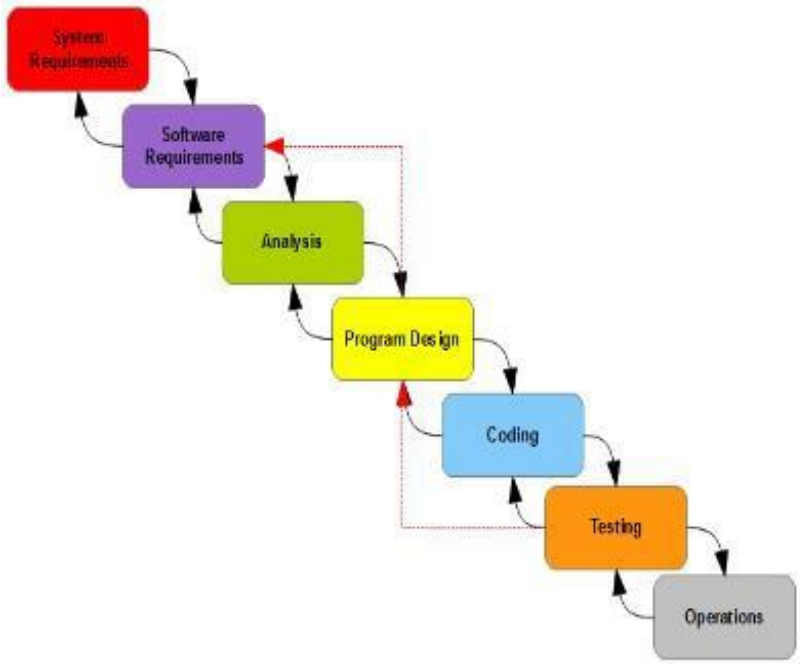
Project Management Methodology

In pursuit of the best outcome for the project build and delivering the best possible application developers implemented a combination of proven project management methodologies. Developers used both Agile and Waterfall concepts, as a result developers were able to create this project with the benefits of Agile while maintaining the proven methods within the waterfall methodology.

Waterfall

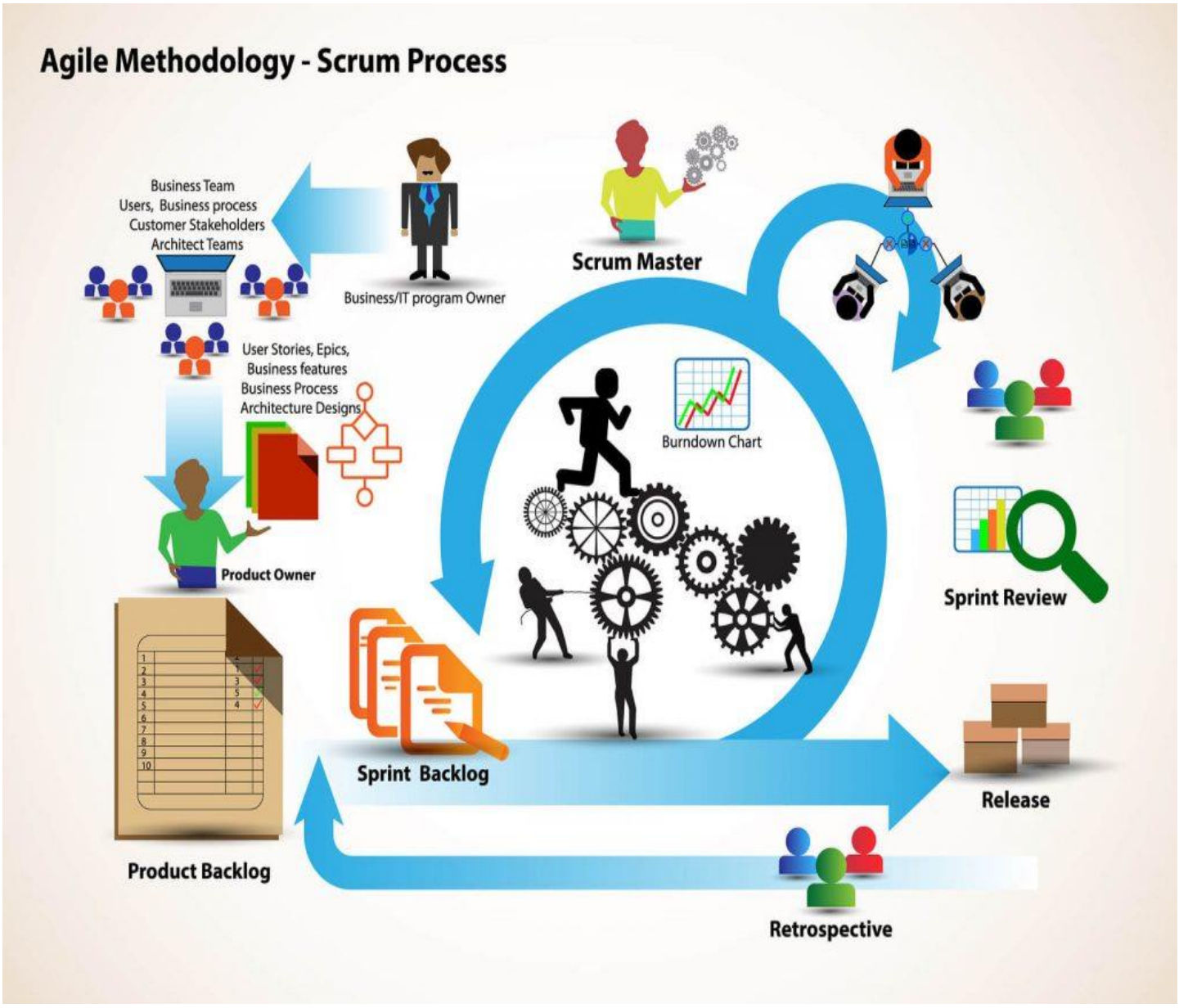
- A sequential project management design process where a project is divided into several phases.
- Each stage is executed in sequence.
- A developer cannot move on to the next phase until the current phase is completed.
- Only when all stages or phases are completed is the project considered done.

- It is named waterfall since it moves from stage to stage, and it cannot go back like a waterfall.



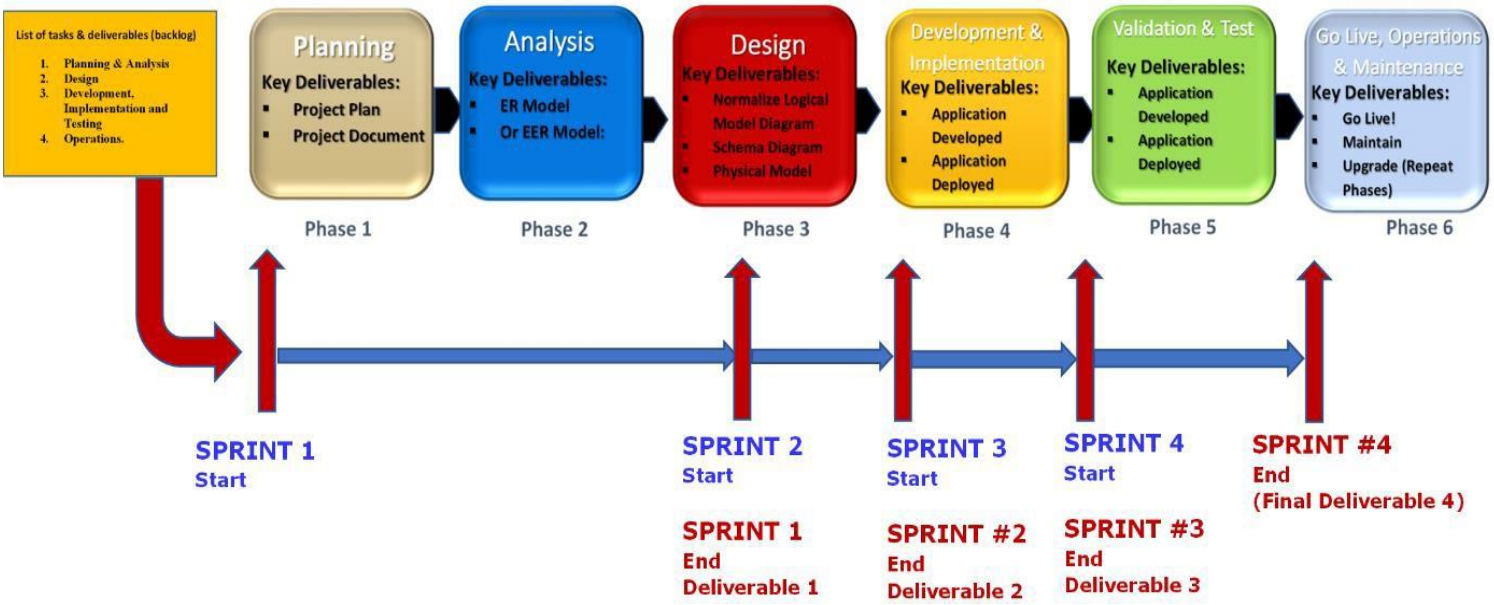
Agile

- An ***incremental*** approach to implementing a Project/Application by implementing subsets of FEATURES in pieces!
- The entire Project/Application is ***DIVIDED*** and ***EXECUTED*** in increments or subset called a SPRINT.
- The Agile methodology allows for changes to be made after the initial planning. Re-writes to the program, as the business decides to make changes, are expected. Because the Agile methodology allows you to make changes, it's easier to add features that will keep you up to date with the latest developments in your industry
    - Agile is the next phase in the evolution of waterfall and includes many subsets of methodologies.
    - Scrum – is what developers used to develop the application.



Approach

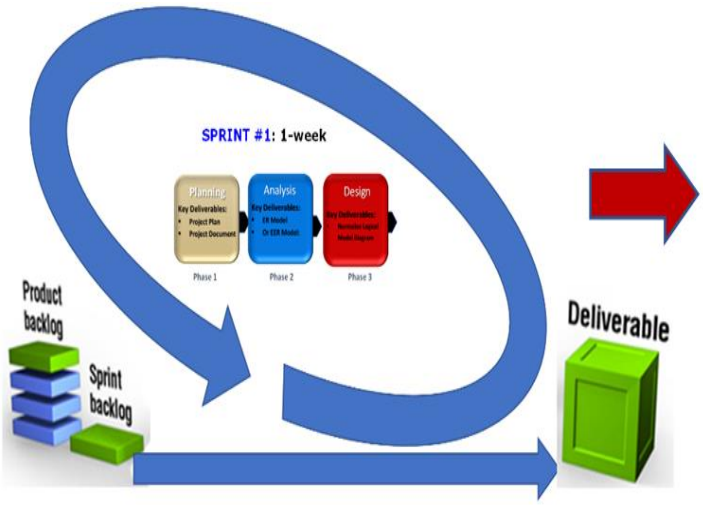
The development team embedded the Agile methodology within a Waterfall Framework. Below is an image of this combination.

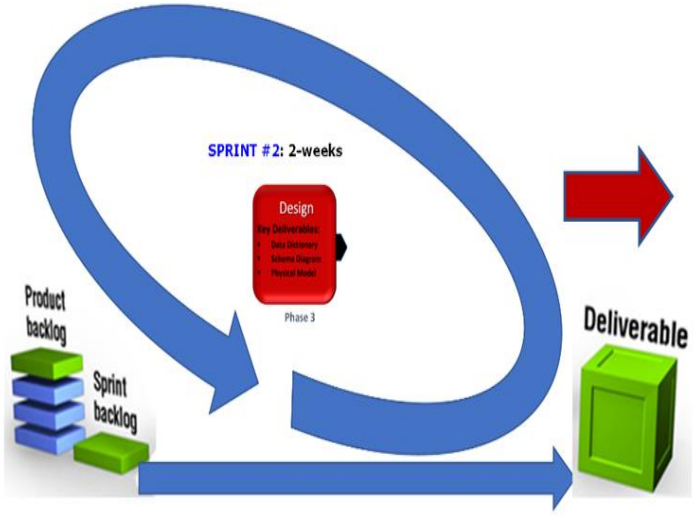


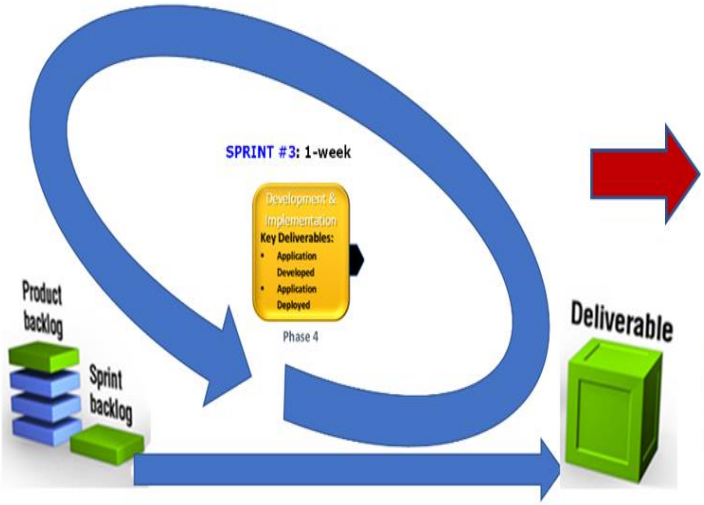
The following table describes the deliverables of each sprint and the corresponding phase during the waterfall methodology.

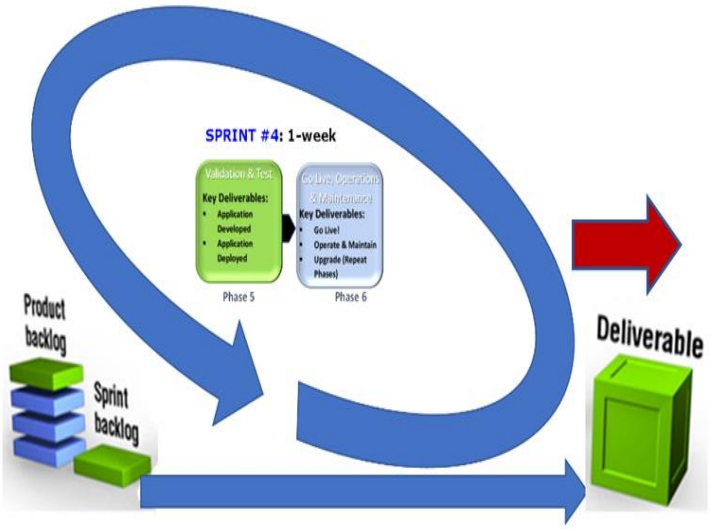
AGILE SPRINT #	WATERFALL PHASE	Output Deliverable
SPRINT #1	Planning	1. **Create Project Document** – Formatted and populated as per requirements. 2. **Business Requirements (Included in Project Document** – List of Business & Technical Requirements from customer.
	Analysis	3. **ER/EER Conceptual Model Diagram**
	Design Phase (Part 1)	4. **Normalized Logical Model Diagram**
SPRINT #2	Design Phase (Part 2)	5. **Data Dictionary matrix** 6. **Physical Schema Design Diagram** – from Normalize Logical Model + DataDictionary combination.
SPRINT #3	Development & Implementation	7. **Database application developed & implemented** – This includes the Database application installed, setup and configured 8. Generate the actual **Physical Schema Diagram** – from the Database & compared to the **Physical Schema Design Diagram** – to validate the design.
SPRINT #4	Validation & Testing	9. **Unit & Integration testing**.
	Operations	10. **Operations** – or keep database running. Keeping the lights on!

Here is a visual of the 4 sprints used in the embedded agile approach. The total time is 5 weeks.









END of Project

Application Business Requirements

The following pages contain the ***Business Requirements captured*** by the **Business Analyst**. For simplicity, note the following legend that describes the highlighting markings in the business requirements:

Business Requirements

About Us:

EZ-Car Rental is an auto rental company that rents vehicles such as cars, SUVs, minivans & cargo vans to customers. In addition, other specialized vehicles such as trucks, motorcycles, boats, mobile homes, etc. We operate in several countries with rental agency locations in the US, Canada, Mexico, UK, Japan & Australia. Within each country we own and operate rental agencies located in cities, regions and state. For example, New York City has 2 rental agencies in Manhattan, one in Brooklyn and two in Queens located at each airport. With multiple rental agencies in cities, states etc., a customer can pick up a vehicle in one location and drop it off at another.

Current Challenges:

Our current rental system is outdated, with a poor user-experience, inefficient (breaks often thus expensive to operate), does not meet our business requirements, and is not scalable (cannot be easily updated with new features). Another very important shortcoming of the current system, is the lack of elasticity since it does not give us the flexibility to scale-up or scale-down resources during business trends and seasonal changes in the market.

We want to invest in modernizing our business with a new vehicle management system that can meet these challenges and delivers a great user-experience, meet our new business requirements, scalable, and elastic to adopt to business trends and seasonal market changes. Elasticity is very important since recently we have been faced with a new type of competition; small rental companies that are nimble and can quickly adopt to market changes thus able to provide new offerings that are appealing to customers thus affecting our profits. These smaller competitors are using new technologies that enable them to be nimble and elastic. Figurative speaking "*they are eating our lunch*".

We look forward to your proposed architecture & implementation of this new system. Below are our business requirements.

Our Rental Agencies:

A **rental agency** is the location where customers visit to pick up and drop off rental vehicles. Each **rental agency** is identified by a unique ***rental agency ID*** number, ***agency name***, ***address*** that is composed of the following elements: ***address line1, address line 2 (***which is optional and used for apartment number, suite or any additional address information required), **city**, ***state code*** (which is the two-character code for a state in the US), ***zip code*** & ***country***. In addition, we also need to capture the agency's ***phone number***, and ***email*** which is unique for all agencies as all emails are.

Our Customers:

EZ-Car Rental offer their services to two types of **Customers**: **Corporate Customers** & **Retail Customers**. Corporate Customers are individuals whose corporation have a contract with us to use our services with special corporate rate for their employee's rental services. On the other hand, **Retail Customers** are consumers not associated with a company and engaging in personal rental.

Requirements for All Customers (Retail & Corporate Customers)

To run our business, the application must store the following customer information for both types of **customers** (***retail*** & ***corporate***) so this data is common to both types of customers:

- A ***Customer ID*** number which uniquely identifies the customer, ***customer name*** which is composed of: ***first name***, ***last name***.
- ***Birth date***, ***Age***, ***Address*** which includes the elements: ***address line1***, ***address line 2 (***which is optional and used for apartment number, suite or any additional address information required), ***city***, ***state code*** (which is the two-character code for a state in the US), ***zip code*** & ***country***.
- Customer ***phone number*** & ***email*** (unique like all emails and required to rent).
- In addition, a driver license is required to reserve and rent a vehicle. Therefore, we need to capture the unique ***driver license number (****an alpha numeric character string containing numbers & characters****.*** *Note that in the USA the format for the driver license number can be under 15 characters, but in other countries it could go up to 22 to 25 characters*) ***driver license expiration Date*** and ***driver license state***. In addition, note the following ***business rule*** policy regarding the business importance of the ***driver license number***:

1. ***The driver license number is used throughout the business to identify a customer for searching, reporting etc.***

2. ***Therefore, the driver license number is the main unique ID for a customer to be identified and managed from a business perspective.***

Business Requirements

Our Customers (Cont.):

- A very important attribute we need to capture for every customer is the **credit card**. For our credit card processing and transactions, we need to capture the following *credit card* components: *credit card number* that uniquely identifies the credit card and is a 16-character number digits. We also need to capture the *credit card owner name*, in addition, credit card processing attributes such as *credit card issuing bank code*, *credit card issuing bank name*, *credit card network company code*, *credit card network company name*, *credit card processing merchant service company code*, *credit card merchant service company name*, *credit card corporate merchant bank code*, and *credit card corporate merchant bank name*. Important – further details on these credit card processing attributes will be provided in sections to follow. Is important that these attributes are clearly understood to correctly design the system.
- Other attributes of credit card are *expiration date*, *billing address* composed of *address line1*, *address line 2* (which is optional and used for apartment number, suite or any additional address information required), *city*, *state code* (which is the two-character code for a state in the US), *zip code & country*, *credit card limit* (which is the maximum amount of money a customer can charge on their credit card), *credit card available credit* (which is how much you have left to spend with your credit card or unused amount within your limit). Note that we will capture the credit card limit to a maximum of $999,999.99, since we don't expect our customers to have a credit limit of $1 Million dollars. Finally, *activation status* (which a is true if the credit card is active and can be used, or false when the credit card is not active or disabled).
- During the interview with business stakeholders provided the following ***Business Rules*** related to a credit card:

1. *You cannot reserve or rent one of our vehicles without a credit card.*
2. *A customer can have many credit cards they can use to pay for rental transactions.*
3. *A credit card can be owned by the one customer or co-owned by other individuals such a family member or corporate entity the customer works for. Therefore, many customers can own the same credit card and a credit card can be owned by many customers.*

The Credit Card processing Workflow

Processing of the credit card transactions is a key part of this business and is important that we store data for each step of the credit card processing process. Therefore, the credit card processing attributes discussed in previous section need to be further analyzed and clearly understood. We will now provide the definition and detailed information on each of these credit card processing attributes: *credit card issuing bank code*, *credit card issuing bank name*, *credit card network company code*, *credit card network company name*, *credit card processing merchant service company code*, *credit card merchant service company name*, *credit card corporate merchant bank code*, and *credit card corporate merchant bank name*:

- ***Credit Card Processing Merchant Service Company*** – In credit card processing the merchant is the retailer where a customer purchased the goods and services and pay using a credit card. **EZRental Inc.**, is the merchant in this scenario. The *Credit Card Processing Merchant Service Company* is the institution which works directly with the merchant (**EZRental Inc.**) to provide handle the credit card processing services and handles all the complexity of credit card processing and interactions with the other financial entities involved in the credit card processing process on behalf of the merchant (**EZRental Inc.**). The *Credit Card Processing Merchant Service Company* provides the Merchant or Business with the hardware which the customer swipes or inserts to pay for goods and services with their credit card. As part of the credit card processing cycle, the first financial institution which the *Credit Card Processing Merchant Service Company* interacts with is the *Credit Card Network Company* (which we will cover next). The *Credit Card Processing Merchant Service Company* ensure the merchant (**EZRental Inc.**) is connected to the right *Credit Card Network Company*. We will describe the *Credit Card Network Company* next, nevertheless, we need to capture the following information for the *Credit Card Processing Merchant Service Company*:

- ***Credit Card Processing Merchant Service Company Code*** – In our business, we use and store a number code used to identify the *Credit Card Processing Merchant Service Company*. This code has business meaning and appears in reports and discussed by users. Therefore, a *Credit Card Processing Merchant Service Company* can be identified by this code.
- ***Credit Card Processing Merchant Service Company Name*** – In our business, we also use and store the name of the *Credit Card Processing Merchant Service Company*. This name also has business meaning and appears in reports and discussed by users. Therefore, a *Credit Card Processing Merchant Service Company* can be identified by its name.
- Below is a listing of the values/instances of the *Credit Card Processing Merchant Service Company Code* and *Credit Card Processing*

Credit Card Processing Merchant Service Company Code	*Credit Card Processing Merchant Service Company Name*
1	Stax by Fattmerchant
2	Helcim
3	Dharma Merchant Services
4	Payment Depot
5	National Processing
6	Block
7	Intuit Quickbooks
8	PayPal
9	Stripe
10	Flagship Merchant Services
11	Clover

Business Requirements

Our Customers & Credit Card Processing (Cont.):

- ***Credit Card Network Company*** – In credit card processing the objectives of the ***Credit Card Network Company*** is to process transactions between the ***Credit Card Issuing Bank*** (which we will cover next) and the ***Credit Card Processing Merchant Service Company***. Covered previously. The ***Credit Card Network Company*** act like bridges between the ***Credit Card Issuing Bank*** that issue credit card and the ***Credit Card Processing Merchant Service Company*** that handles the transaction from the merchant (**EZRental Inc.**). The ***Credit Card Network Company*** that interacts with the ***Credit Card Issuing Bank*** to determine whether to approve or deny the transaction. And then the ***Credit Card Network Company*** notifies the merchant (**EZRental Inc.**) if the purchase was approved or denied. The ***Credit Card Network Company*** is a digital infrastructure that facilitates credit card transactions and prepares the transaction for the ***Credit Card Issuing Bank***. We will describe the ***Credit Card Issuing Bank*** next, nevertheless, we need to capture the following information for the ***Credit Card Issuing Bank***:
  - ***Credit Card Network Company Code*** – We use and store a number code to identify the ***Credit Card Network Company***. This code has business meaning and appears in reports and discussed by users. Therefore, a ***Credit Card Network Company*** can be identified by this code.
  - ***Credit Card Network Company Name*** – In our business, we also use and store the name of the ***Credit Card Network Company***. This name also has business meaning and appears in reports and discussed by users. Therefore, a ***Credit Card Network Company*** can be identified by its name.
  - Below is a listing of the values/instances of the ***Credit Card Network Company Code*** and ***Credit Card Network Company Name*** we use at **EZRental Inc.**:

Credit Card Network Company Code	*Credit Card Network Company Name*
1	**American Express**
2	**Visa**
3	**Mastercard**
4	**Discover**
5	**Diners Club**
6	**Interlink**
7	**Star**
8	**Accel**
9	**Interac**
10	**Visa ReadyLink**
11	**Pulse**
12	**JCB (Japan Credit Bureau)**
12	**Rupay**

- ***Credit Card Issuing Bank*** – The ***Credit Card Issuing Bank*** is the financial or lending institution that offers the Credit Card and pays for the goods and services until the customer pays back the credit/loan. These are Banks, Lending Institutions, Credit Unions, Fintech companies, etc. These institutions issues/provides the credit for the customer. The cardholder borrows money from the credit card issuing bank each time they make a purchase, and when they pay their credit card bill, they're paying the ***Credit Card Issuing Bank***. We need to capture the following information for the ***Credit Card Issuing Bank***:
  - ***Credit Card Issuing Bank Code*** – In our business, we use and store a number code used to identify the ***Credit Card Issuing Bank***. This code has business meaning and appears in reports and discussed by users. Therefore, a ***Credit Card Issuing Bank*** can be identified by this code.
  - ***Credit Card Issuing Bank Name*** – In our business, we also use and store the name of the ***Credit Card Issuing Bank***. This name also has business meaning and appears in reports and discussed by users. Therefore, a ***Credit Card Issuing Bank*** can be identified by its name.
  - Below is a listing of the values/instances of the ***Credit Card Issuing Bank Code*** and ***Credit Card Issuing Bank Name*** we use at **EZRental Inc.**:

Credit Card Issuing Bank Code	*Credit Card Issuing Bank Name*
1	**American Express**
2	**Bank of America**
3	**Barclays**
4	**Capital One**
5	**Chase**
6	**Citi**
7	**Discover**
8	**Synchrony Bank**
9	**U.S. Bank**
10	**Wells Fargo**

Business Requirements

Our Customers & Credit Card Processing (Cont.):

- ***Credit Card Corporate Merchant Bank*** – In credit card processing, the ***Credit Card Corporate Merchant Bank*** is the bank used by the merchant **EZRental Inc.**, to handle credit card processing money transactions, payments, etc., between **EZRental Inc.,** and the ***Credit Card Processing Merchant Service Company*** that handles the Credit Card Processing on behalf of **EZRental Inc**. In short, it is the bank that has the bank account used by **EZRental Inc.,** to handler the accounting for Credit Card Processing Transactions. We need to capture the following information for the ***Credit Card Corporate Merchant Bank***:

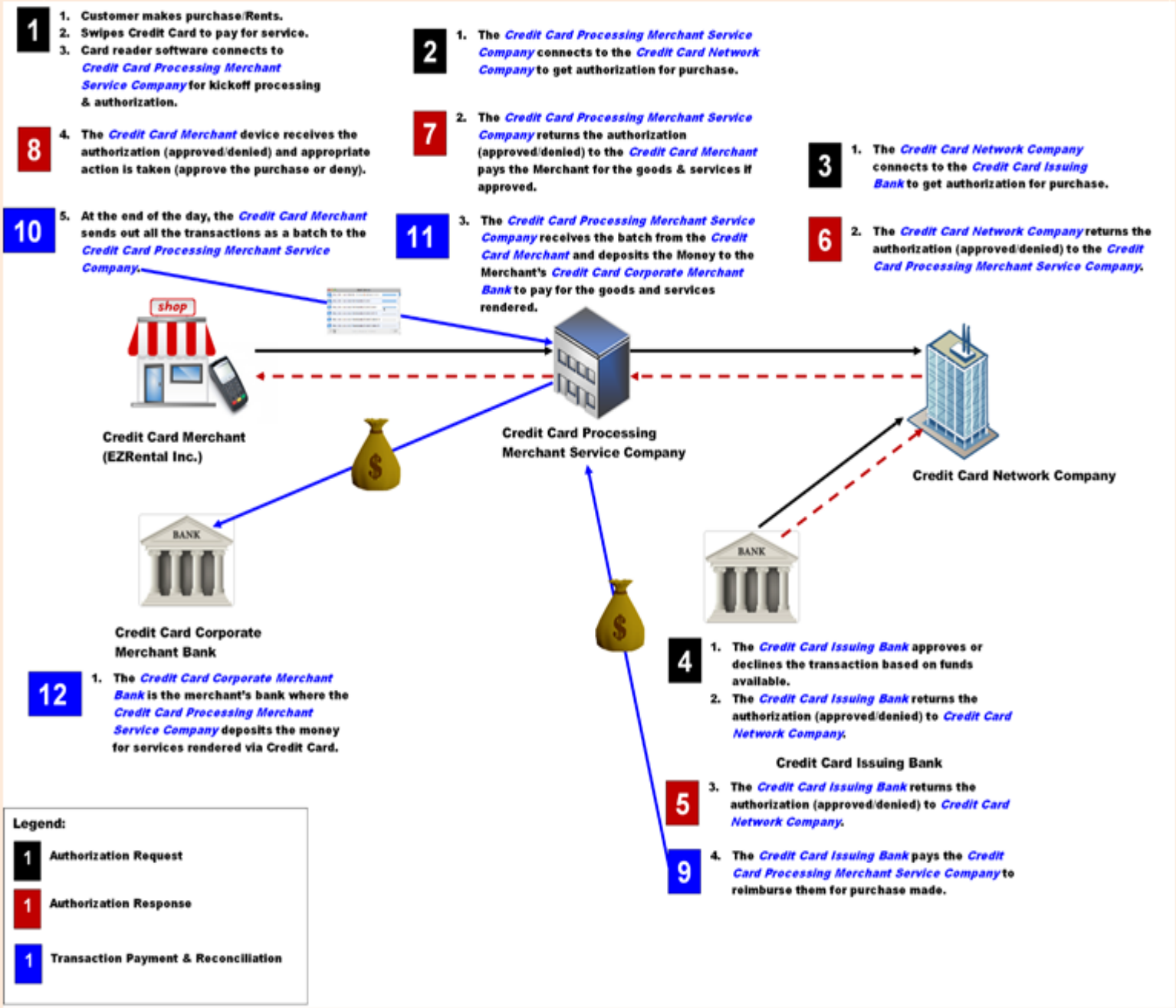
  - ***Credit Card Corporate Merchant Bank Code*** – In our business, we use and store a number code used to identify the ***Credit Card Corporate Merchant Bank***. This code has business meaning and appears in reports and discussed by users. Therefore, a ***Credit Card Corporate Merchant Bank*** can be identified by this code.
  - ***Credit Card Corporate Merchant Bank Name*** – In our business, we also use and store the name of the ***Credit Card Corporate Merchant Bank***. This name also has business meaning and appears in reports and discussed by users. Therefore, a ***Credit Card Corporate Merchant Bank*** can be identified by its name.
  - Below is a listing of the values/instances of the ***Credit Card Corporate Merchant Bank Code*** and ***Credit Card Corporate Merchant Bank Name*** we use at **EZRental Inc.**:

Credit Card Corporate Merchant Bank Code	*Credit Card Corporate Merchant Bank Name*
1	**Chase**
2	**Citi**
3	**Capital One**

Business Requirements

Our Customers & Credit Card Processing (Cont.):

Below, is a pictorial representation of the interaction between the credit card processing entities *Merchant (EZRental Inc.), credit card processing merchant service company, credit card network company* and *credit card issuing bank*:



In summary, we need to capture the data for the credit card processing attributes: *credit card issuing bank code, credit card issuing bank name, credit card network company code, credit card network company name, credit card processing merchant service company code, credit card merchant service company name, credit card corporate merchant bank code*, and *credit card corporate merchant bank name*.

Business Requirements

Our Customers (Cont.):

Corporate Customers

Corporate Customers are customers who are renting vehicle during business travel and their company have a contract with **EZRental Inc**. These companies get special corporate rate for their employee's rental services. Therefore, for our **corporate customers only**, we must store the following attributes/properties: unique *company ID* (we have a unique ID number for each company doing business with us), *company name*, *company address* which contains the elements: *address line1, address line 2* (which is optional and used for apartment number, suite or any additional address information required), *city*, *state code*, *zip code* (which is the two-character code for a state in the US) & *country*, in addition, *company contact* which is composed of *company representative name*, *contact phone number* & *contact email* (unique as all email addresses). And finally, we need to store the *company discount percentage rate* which is the discounted percentage applied to a corporate customers rental. The company Discount percentage rate is store in the database as a decimal percentage value, for example 20% is stored as 0.20, 30% as 0.30. 50% as 0.50 etc. This discount percentage (0.0x) is applied to the **Vehicle Rental Categories** which determines the price of each category to determine the total discount. Therefore, when a corporate customer rents a vehicle from a vehicle category (such as economic, compact, standard etc.), this discount percentage is applied to each of the categories during the rental/reservation process. Note that every company has a different percentage rating depending on their contract with **EZ-Rentals Inc**. For example, some companies have 20% discount towards their rentals, which would be stored as 0.20 in the database, some have 30% (0.30) etc. Vehicle Rental Categories are discussed in more details later in these requirements.

Retail Customers

Retail Customer Discounts

Retail Customers can (but don't have to) leverage promotional **discounts** or multiple coupons obtain from other businesses, internet, magazine, organizations, etc., to save money on their rentals. Therefore, we need to capture specific data for the promotional discounts used by a retail customer. A *Promotional Discount* is composed of the following attributes: *discount ID*, a unique random number which uniquely identifies a discount, another unique *discount code* or the coupon code itself used to redeem the coupon, which is an alphanumeric code **10-characters** long. This code is generated by our marketing team and published to magazines, newspapers, internet e-commerce sites, etc. Finally, the last attribute is *discount code description* or description of the discount. Examples of currently used *discount ID*, *discount code*, *discount code description* are shown in table below:

Discount ID	*Discount Code*	*Discount Code Description*
1234..	AAA9970054	**AAA Membership Discount - 25% off base rate plus 10% donated for breast cancer research.**
5678..	GOV8756921	**Government Employee Discount - 30% off base rate**
9101..	STA3415632	**State Employee Discount for 25% off base rate**
1213..	VET2055179	**Veteran Discount 35% off base rate Plus 10% donation to veteran's family fund.**
Etc..	Etc..	**Etc..**

The following business rules were identified regarding the retail customer discount program:

1. *Only a Retail Customer can use Discounts, no other type of customer, e.g., Corporate Customer, can apply a discount.*
2. *Discounts are applied during reservation of a vehicle or during the actual rental process only.*
3. *A Retail Customer can apply multiple Discounts throughout their lifetime as an EZRental Inc., customer, NEVERTHELESS, ONLY ONE Discount can be applied for a reservation or rental instance. You cannot apply multiple discounts to a reservation or rental.*
4. *A Discount can be used by many Retail Customers and many Retail Customers can use a Discount.*
5. *When a Discount is used by a Retail Customer, we need to capture the Discount Submitted Date which is the date the customer provided the discount and the Discount Redeemed Date which is the date the customer used the discount for a rental. In this business a customer can submit a discount during registration on one date but use it in a future date when they are renting.*

Business Requirements

Our Customers (Cont.):

Retail Customer EZPlus Rewards Program

Retail customers can opt-in to enrolled in the **EZPlus Rewards Program** where they earn points for every rental of a vehicle. These rewards points can be redeemed for future rentals. Note that the **EZPlus Rewards Program** is **optional** for retail customers & points are earned only when they rent vehicles. For the **EZPlus Rewards Program** we need to store unique random number *EZPlus ID*, the unique *Ezplus rewards code* which is the code used in the business when managing the **EZPlus Rewards Program**. This random code is generated and assigned to a **Retail Customer** by the client application. The number starts with the 3-characters EZP and a 10-digit number e.g., **EZP9999999999**, and the final attribute is the *EZPlus rewards earned points*, which is an integer that indicates the number of rewards points earned that accumulated after all the rentals and can be used to save on future rentals. **Examples of currently used** *EZPlus ID*, *EZPlus rewards Code* **and** *EZPlus earned points* **that we currently use are**:

EZPlus ID	*EZPlus Rewards Code*	*EZPlus Rewards Earned Points*
1234..	EZP9009854637	**10000**
5678..	EZP1000192461	**500**
9101..	EZP6493238865	**159000**
1213..	EZP2005135627	**23000**
	Etc..	**Etc..**

The following business rules apply to the EZPlus Rewards Program:

1. *Only a Retail Customer can leverage the EZPlus Rewards Program, no other type of customer such as a Corporate Customer can join the EZPlus program.*
2. *The EZPlus Rewards Program is OPTIONAL. A Retail Customer can join the EZPlus Rewards Program during registration or any other time after or not join at all.*
3. *A Retail Customer can drop out of the EZPlus Rewards program at any time.*
4. *Every time a Retail Customer that is member of the EZPlus Rewards Program rents a vehicle, they earn* ***1000 EZPlus Rewards Points****. When a Retail Customer member of the EZPlus Rewards Program earns* ***10,000 EZPlus Rewards Points*** *they earn a FREE RENTAL.*

As an incentive for our retail customers to join the **EZPlus Rewards Program** during registration to become a customer, we offer a *EZPLus sign-up rewards points* of **1000** *EZPlus Rewards Earned Points*. Also note that the maximum number of *EZPlus Rewards Earned Points* is capped at **50,000** points. A Retail Customer cannot accumulate more than **50,000** points in the *EZPlus Rewards* program.

Business Requirements (Cont.)

Our Customers (Cont.):

In this business, we have the following business rules for our customers (*Retail* or *Corporate*):

1. *We only have two types of customers retail customer or corporate customers. No other type of customer exists.*
2. *The minimum age to be a customer and rental our vehicles is 21 years old. This rule must be enforced.*
3. *A customer cannot be a retail & corporate customer at the same time. A customer can only rent as a retail customer or as a corporate and these transactions must be separate. We don't want our customers to be able to combine both retail customer discounts, rewards program and corporate rates at the same time.*

Our Vehicles:

EZ-Car Rental needs a system to manage their vehicles for renting, maintenance, selling, etc. Vehicles are classified into 4 main types: **CAR**, **SUV**, **MINIVAN**, and **CARGO VAN**. These are the vehicles most rented and available at every rental agency. Nevertheless, there are other categories of vehicles available only certain rental agency locations such as **RECREATIONAL VEHICLES**, **MOTORCYCLES**, **MOBILE HOMES**, etc. No matter what type of vehicle being rented, all vehicle types share the following common characteristics:

- Each vehicle is identified by the random number *vehicle ID*. In addition, each vehicle is also identified by the alpha-numeric *vehicle VIN number*. Note the following business rule on a *vehicle VIN number*:

  1. *The vehicle VIN number is used throughout the business to identify a vehicle for searching, reporting etc.*
  2. *Therefore, the vehicle VIN number is the unique ID for a vehicle to be identified and managed from a business perspective.*

- Other attributes include the *vehicle name* composed of *make*, *model* & **year**. Additional attributes are *color*, also the *license plate* composed of the following components: *license plate number*, *license plate state*.
- More attributes are *mileage*, *transmission type* of the vehicle. The Transmission Type attribute has business value thus used in reports and in the business processes. The values used for *transmission type* and a *transmission type description* as follows:

Transmission Type	*Transmission Type Description*
1	**Manual Transmission**
2	**Automatic Transmission**
3	**Continuously Variable Transmission (e.g., CVT).**
4	**Semi-automatic Transmission**
5	**Dual-clutch Transmission**
6	**Transaxle Transmission**

- *seat capacity* attribute, which is the number of seats in the vehicle. Vehicles such as *cars* have a seat capacity of 5 passengers (2 in front and 3 in the back), *SUVs* have 7 or 8 passengers. Cargo Vans have only 2 passenger seat capacity, Minivan have 8 to 9 passengers, special vehicles such as passenger van hold 12 passenger seat capacity, a shuttles bus can hold 16 to 20 passengers, mini-buses 30 to 40 passengers and large busses can hold 70 passengers.

- All vehicles also have a special code and description that we use to track the vehicle status named *vehicle status ID*. This is a unique number that identifies the status of a vehicle, which works in conjunction with *vehicle status description* which describes the status represented by the *Vehicle Status ID*, such as **reserved**, **rented**, **available**, **maintenance**, **not available**, **transferred**, etc. Below Is the list of vehicle status IDs we are currently using and their descriptions:

Vehicle Status ID	*Vehicle Status Description*
1	**Available**
2	**Reserved**
3	**Rented**
4	**Not available**
5	**Maintenance (Not available)**
6	**Dropped off and located at another agency**
7	**In Transport to Owning Agency**
8	**No Longer available for rental**

Business Requirements (Cont.)

Our Vehicles (Cont.):

In addition to these attributes shared by all vehicles, there are 4 main categories of vehicle which share unique characteristics than the other types of vehicles found in our agencies. These 4 types are as follows:

- A **Car** is a vehicle whose ***trunk capacity*** (measured in cubic feet volume) is advertised to our customers. Customers can decide which vehicles better fits their needs based on the trunk capacity and number of luggage they are carrying etc. For example, a *luxury Mercedes E class* car has a trunk capacity of 18.5 cubic ft., which has a large trunk capacity.
- An **SUV** is a vehicle with a ***towing capacity*** attribute in pounds. Towing capacity is a single number in pound or could also be a decimal number in pounds. For example, some of our SUV have a maximum towing capacity of 3,000 pounds etc. Another attribute of SUV is an attribute classification if the SUV is ***All-Wheel-Drive***, which stores a Boolean value of **YES/NO** or **TRUE/FALSE**.
- A **Minivan** has the option of ***having a disability package***, which is also a Boolean value of **YES/NO** or **TRUE/FALSE**.
- Finally, a **Cargo Van**, has a ***cargo capacity*** in cubic feet volume. For example, the typical volume of our Vans is 245 cubic feet (cu.ft.). Cargo Vans also have a ***maximum payload*** attribute that determines how much weight in pound it can hold. Our cargo vans have typically a maximum payload of 3,880 lbs.

- As stated previously, there are other types of vehicles of interest that in some location we may want to store data on other than car, SUV minivans and cargo van.
- Note that the following Business Rules were identified by the business stakeholders on the vehicles:

- ***A reservation/rental can only be for one of these four categories of Vehicles or other vehicle types, not a combination***.

- ***This means, you can only rent either a car, SUV minivans, cargo van or other for a reservation or rental, not a combination such as a car & SUV at the same time. Each reservation is unique to one vehicle***.

Below are additional business rules for our vehicles and agency ownership:

- ***Every vehicle is owned by one agency. The vehicle can be pick-up and dropped-off at any agency, but only one agency is the vehicle's owning agency. An agency can own many vehicles, but a vehicle can only be owned by one agency***.

- ***A vehicle can currently be located at any agency depending on where it was dropped-off after a rental. We need to track the current agency where the vehicle is located, to arrange a transfer or a rental that will ultimately direct the vehicle to the owning agency***.

Reservation Process:

A vehicle must be reserved if a customer wants to guarantee the vehicle will be available for rental. There is a distinction between a reservation and a rental. A reservation guarantees a vehicle will be ready for you to be pick-up and rented. A rental means a customer complied with the reservation and rented the vehicle. On the other hand, a customer can walk into an agency and rent without reservation but only vehicles that are available at the time and not reserved.

We have the following business rules for reserving a vehicle reservation:

- ***A reservation is NOT made for a specific vehicle, but to a vehicle rental category. Rental category examples are economy, intermediate, full size, luxury***.

- ***Thus, a customer makes a reservation of a vehicle rental category at a rental agency. Therefore, the reservation process involves a customer a vehicle rental category and the rental agency*** where the vehicle will be picked up.

Business Requirements (Cont.)

Reservation Process (Cont.):

A **Vehicle Rental Category** contains a list of vehicles depending on the vehicle type: Car (economy, intermediate, full size, luxury), SUV (standard, full size etc.), or Cargo Van etc. Each of these categories have a different price range. Therefore, for a vehicle rental category we need to capture the unique *vehicle rental category ID* that identifies the category of the vehicle being reserved or rented, *category name* and finally *category daily rental rate* for the category. We used a specific code for our vehicle rental category ID, category name & daily rental rate. The table below shows the ID, category names and rate we currently using in our business:

Vehicle Rental Category ID	*Vehicle Rental Category Name*	*Category Daily Rental Rate*
1	Car-Economic	$113.99
2	Car-Compact	$115.99
3	Car-Intermediate	$116.67
4	Car-Standard	$119.99
5	Car-Full Size	$121.99
6	Car-Premium	$127.79
7	Car-Luxury	$139.99
8	SUV-Intermediate	$127.99
9	SUV-Standard	$128.99
10	SUV-Standard Elite	$135.99
11	SUV-Full Size	$148.99
12	SUV-Premium	$157.99
13	Minivan-Standard	$152.99
14	Van-Cargo Van	$19.95
15	Pick Up-Mid Size	$69.95
16	Pick Up-Full Size	$105.99
17	Motorcycle-Touring	$19.95
18	Motorcycle-Cruiser	$199.99
19	Motorcycle-Scooter	$79.95
20	Passenger Van (12 passengers)	$161.00
21	Passenger Shuttle (16 passengers)	$180.00
22	Passenger Shuttle (20 passengers)	$220.00
23	Passenger Mini-Bus (30 passengers)	$250.00
24	Passenger Mini-Bus (40 passengers)	$280.00
25	Passenger Large-Bus (80 passengers)	$300.00

We have the following business rule relate to a vehicle and a vehicle rental category:

1. *A vehicle is a member of a vehicle rental category.*
2. *A vehicle rental category can have one, none or many vehicles belonging to that category at any given time, nevertheless, a vehicle can only belong to one vehicle rental category.*

As stated previously, **a customer makes a reservation of a vehicle rental category at a rental agency**. Therefore, the reservation process requires the **customer**, **vehicle rental category** & **rental agency** for a reservation to be made. The following business rules apply to a reservation:

1. *A vehicle can be reserved to be picked up at the **INDICATED** rental agency and dropped off at the **SAME** rental agency.*
2. *A vehicle can be reserved to be picked up at the **INDICATED** rental agency and dropped off at a **DIFFERENT** rental agency.*
3. *A reservation is made only for one pick-up rental agency, but a rental agency can have many reservations for pick-ups taking place.*
4. *A reservation can only be for one drop-off rental agency, but a rental agency can have many reservations drop-offs taking place.*
5. *For reporting and analytics, we need to capture all changes to the reservation's pick-up and drop-off agency, such as all changes by the customer for pick-up agency & drop-off agency later after the original reservation. This means we need to store all history on all reservation pick-and drop-off agency changes.*

When a customer reserves a vehicle rental category for a specific rental agency, we wish to capture the following:

- A unique *reservation ID* which is used by the business to manage and track reservations, the *rental agency ID* where the vehicle will be picked up, and the target *reservation drop-off rental agency*.
- In addition, we need to store the *reservation schedule* composed of *reservation pick up date*, *reservation pick up time*, *reservation drop off date* and *reservation drop off time*, also the *reservation estimated rental cost*. A reservation can have multiple schedule changes during the lifetime of the reservation since customers can make changes to the reservation and we need to track this history of changes for analytical purposes.

Business Requirements (Cont.)

Reservation Process (Cont.):

We have the following business rule relate to the *reservation schedule*:

1. *For reporting and analytics, we need to capture all changes to the reservation schedule, such as all changes by the customer for pick-up date & time and drop-off date & time. This means we need to store all history on all reservation schedule changes.*
2. *A customer can have MANY reservation schedules based on changes to the reservation, but a schedule can only belong to ONE customer.*

- Finally, we need to store the unique *reservation status ID* which is a unique number we use to indicate the status of a reservation and *reservation status description* which describe each of the status such as: **confirmed**, **cancelled**, **completed** etc. Below is an example of the *reservation status ID* and *status description* we currently use in our business.

Reservation Status ID	*Reservation Status Description*
1	**Confirmed**
2	**Modified & reconfirmed**
3	**Cancelled**
4	**Fulfilled & closed**
Etc..	**Etc..**

For a reservation we must adhere to the following business rules:

1. *A customer can make none, one or many reservations for a vehicle rental category at a rental agency.*
2. *A rental category can be reserved by none, one or many customers at a rental agency.*
3. *A rental agency can get many or no reservations for a vehicle rental category by a customer.*
4. *A reservation can only have one pick-up rental agency location, but a rental agency can have many reservation pick-ups happening.*
5. *Each reservation has a drop-off rental agency (may be different than pick-up rental agency). A reservation can only have one drop-off rental agency location, but a rental agency can have many reservation drop-offs taking place.*

The Rental Process:

Once a vehicle has been reserved, the vehicle can be rented (picked up/dropped off) as per the scheduled of the reservation agreement. **A rental means a customer complied and fulfilled the reservation and rented the vehicle**.

For the rental process, the following business rules apply:

1. *A customer rents a vehicle Rental Category at a rental agency. This means the rental process requires the **customer**, **vehicle rental category**, and & **rental agency** for a rental to be complete.*
2. *A Rental includes a specific Vehicle of the vehicle rental category. A vehicle can be rented many times, but a rental is only for one vehicle only. You cannot rent multiple vehicles in one rental contract.*
3. *During the rental process we may have any of the following business rules/scenarios:*

    1) *A vehicle can be picked up at the **SAME** rental agency as indicated by the reservation and dropped off at the **SAME** rental agency.*
    2) *Or a vehicle can be picked up at the **SAME** rental agency as indicated by the reservation and dropped off at **ANOTHER** rental agency.*
    3) *Or a vehicle can be picked up at **ANOTHER** rental agency other than what was indicated by the reservation and dropped off at **SAME** rental agency of the reservation.*
    4) *A vehicle can be picked up at **ANOTHER** rental agency other than what was indicated by the reservation and dropped off at **ANOTHER** rental agency of the reservation.*
    5) *For reporting and analytics, we need to capture all changes to the rental pick-up and drop-off agency, such as all changes by the customer for pick-up agency & drop-off agency later after the original reservation. This means we need to store all history on all rental pick-and drop-off agency changes.*

    - Note that for scenarios 3 & 4, we cannot guarantee that the vehicle rental category of the reservation will be available at the agency other than what was agreed in the reservation. We will do our best to accommodate the change during these scenarios or find another vehicle that will be closed to the original reservation.

For the rental process, the following business rules also apply:

1. *A rental can only be for one pick-up rental agency, but a rental agency can have many rental pick-ups taking place.*
2. *A rental can only be to one drop-off rental agency, but a rental agency can have many rental drop-offs taking place.*

When a customer rents a vehicle at the rental agency, we need to capture the following information about the rental:

- The *rental agreement ID* that uniquely identifies the rental transaction, *rental pick up date*, *rental pick up time*, *rental drop off date* and *rental drop off time*, *rental pick up odometer value* and *rental drop off odometer value*.

Business Requirements (Cont.)

The Rental Process (Cont.):

- In addition, a customers receive a vehicle with a full tank of gas and customers are expected to return the car on a full tank of gas otherwise they must pay a penalty upon return. Since we understand our customers are busy and may forget to return the car with a full tank of gas, we offer our customers with the option to pay in advance for a full tank of gas at our rates and don't have to worry about returning the vehicle with a full tank of gas. Therefore, we need to capture the unique *rental fuel option ID* or option chosen by the customer, *rental fuel option description* and *rental fuel option additional cost*. We currently use the following fuel option IDs, descriptions, and example of each of the additional cost for the fuel option:

Rental Fuel Option ID	*Rental Fuel Option Description*	*Rental Fuel Option Additional Cost*
1	Return with a full tank or on return, pay for gas that is missing.	$13.97 (Important, this Decimal value of $13.97 is just an example, since the value is calculated during car return process and is based on the current price for a gallon of gas etc. therefore price will vary.)
2	Pay for full tank in advanced at time of rental, return car empty. No refund for unused gas.	$45.99 (Important, this Decimal value of $45.99 is just an example, since the value is calculated during car return process and is based on the current price for a gallon of gas etc. therefore price will vary.)

- Also, we give customer options for car insurance & protection, therefore we need to capture the unique *insurance option ID*, *insurance option description* and *insurance option additional cost*. We currently use the following insurance option IDs, descriptions, and cost:

Rental Insurance Option ID	*Rental Insurance Option Description*	*Rental Insurance Option Additional Cost per Day*
1	No insurance. Opt-out	$0.00
2	Collision Damage Waiver Max - Agency will pay for damage, lost or stolen vehicle.	$49.99
3	Collision Damage Waiver 3000 - Agency will pay for first $3,000 of loss or damage, renter pays all loss & damage after $3,000.	$39.99
4	Lability Extended Protection – Agency provides renter with third party liability protection up to $1 Million per accident for bodily injury or death or property damage to others.	$89.99
5	Roadside Assistance Plus – 24/7 roadside assistance, replacement for lost keys, flat tire service, fuel delivery, etc.	$15.99

- Other attributes required for the rental that we need to capture are the unique *rental status ID & rental status description*. We currently use the following rental status IDs & descriptions:

Rental Status ID	*Rental Status Description*
1	Picked up as scheduled.
2	Dropped off as scheduled.
3	Returned late
4	In progress.
5	Roadside assistance in progress.
7	Unknown

Business Requirements (Cont.)

The Rental Process (Cont.):

- Other attribute we need to capture the ***rental deposit*** for a rental. The rental deposit value is calculated based on the **rental period + 25% of the rental period** and for any damage or other charges that were incurred during the rental period. This deposit is refunded to the customer's credit card when the vehicle is returned in the condition in which it was rented.
- Finally another attribute we need to capture is the ***rental total cost*** or total cost that needs to be paid by the customer. This value is calculated based on selected *fuel option, insurance option, vehicle rental category* price and other factor such as such as duration of the rental etc.

We need to be able to associate a reservation to a rental and vice versa, therefore we maintain the following additional business rules for our rental & reservation:

1. ***A reservation is made for a rental and the opposite holds true; a rental is based on a reservation.***
2. ***But NOT all rentals are based on a reservation. We allow a customer to walk into a rental agency and rent a vehicle without a reservation.***
3. ***When a reservation is made for a rental, then it must be for only one rental, and a rental can be for a reservation but not mandatory since a customer can walk into an agency and rent a vehicle without a reservation.***

Our Employees:

EZ-Car Rental currently has 5,500 employees across the world. We do expect to grow as we move into new markets such as Asia, Africa, and the Mediterranean. But our business does not require a large workforce, therefore, we don't expect to grow more than 12,000 in the next 10+ years. Our employees consist of *customer service agents* in the Rental Agencies & online support who interact with our customer to reserve and rent vehicles. In addition, *back-office inventory personnel*, *auto specialists* who work in our services centers servicing our vehicles, *drivers* to transport our vehicles from one agency to another and *maintenance personnel* who maintain our agencies and finally our *business team* that handles the day-to-day business activities in our agencies and other roles. For now, we are only interested in storing the following data for all these types of employees:

- An ***Employee ID*** which uniquely identifies the employee, ***employee name*** which is composed of: ***first name***, ***last name***, also ***employee address*** which includes the components: ***address line1***, ***address line 2***, ***city***, ***state code***, ***zip code*** & ***country***. Also, ***employee phone***, ***employee job title*** and ***employee email***. In addition, we need to capture the employee ***social security number***. Below are some business rules and usage for the ***EmployeeID*** and the ***social security number***.

The following business rules related to employees must be followed:

1. ***The employee social security number needs to be protected and secured as per federal regulations. All security measures such as encryption, etc., need to be taken to protect the social security number; therefore, the full social security number cannot be seen by employees, reports, and other business processes.***
2. ***In special cases where the social security number needs to be displayed, only the last 4 digits will be shown using the following format ****_**_1234. Nevertheless, the goal is NOT to display the social security number as much as possible, and it should only be used internally within the application for processing but not displaying.***
3. ***The EmployeeID number is what is used throughout the business to identify an employee for searching, reporting. business processing, etc., therefore, the EmployeeID is the unique ID for an employee to be identified and managed from a business perspective.***
4. ***The minimum age to be an employee of our company 18 years old. This rule must be enforced.***

Security & Application Access:

To access our systems proper security and authentication is required. Only authorized users can have access our agencies Point-0f-Sales & Back-End Management systems. In addition to our **EZRental.com** portal by our customers. Therefore, due to security and regulatory compliance purpose, we want to separate the employee access data from the customer access data by using two separate user accounts:

- Employee user accounts
- Customer user accounts

Security Access for Employees to Computer Systems in our Agencies (Employee User Accounts):
For our authorized employees & customer service employees to access the agencies Point-0f-Sales & Back-End Management systems they need to log in by entering a username & password for access to the application. This means every employee owns an employee user account.

An employee user account should store the user ***employee user account ID*** a unique identifier alpha-numeric string that identifies the employee user account, ***employee username*** another unique alpha-numeric that identifies each individual user, the ***employee password*** alpha-numeric that is known only to the user, and finally the employee ***email*** to map the user-account to an Employee. Note the following business rule:

1. An employee can own one employee user account only, and an employee user account can only be owned by one employee only since the user account represents the identify of that one employee.

Business Requirements (Cont.)

Security Access for our Customers who register for our EZ-CarRental.com web site (Customer User Accounts):
Customer who accesses our online portal to reserve and rent our vehicles also need a username and password to access our system, therefore each customer owns a customer user account.

A customer user account should store the user *customer user account ID* a unique alpha-numeric string identifier that identifies the customer user account, *customer username* another unique alpha-numeric value that identifies each customer, the *customer password* that is an alpha-numeric known only to the customer, and finally, the customer *email* to map the customer user-account to a customer. Note the following business rule:

1. A customer can own one customer user account only, and a customer user account can only be owned by one customer.
2. For a period of time, we will need to register customers into our **EZRental.com** business, nevertheless the web portal may NOT be implemented or completed when new customers are registering at this time, therefore, for period of time, creating a customer user account when registering a new customer is optional until the Web Portal Application is created. But is important in the future, that we force the creation of customer user accounts when a new customer is registered once the Web Portal Application is ready. It is the responsibility of the database architect(s) and full-stack developers to update this feature when the appropriate time comes.

Vehicle Transportation:

We need to know where our vehicles are located at all times, such as at the Rental Agency that owns the vehicle, another Rental Agency that does not own the vehicle, being transported from one Rental Agency to another as a result of a vehicle transfer after a rental to the owning rental agency, being transported as a new delivery to a Rental Agency from our distribution center, being transported for maintenance, or currently being rented by a customer. Vehicles need to be tracked or location status known. At this time, we are only interested in tracking when a vehicle is transported from one Rental Agency to another Rental Agency under the following scenarios:

- Vehicle can be located at a Rental Agency that does not own the vehicle after a rental dropping off at a different location than the picked up owning Rental Agency, thus vehicle eventually needs to be transported and delivered to the owning agency.
- Another non-owning Rental Agency requests support from other Rental Agency(s) for loans of vehicle(s) to borrow due to an unexpected busy period and requesting agency is short on inventory. After the first agency is done with the loaner vehicles, these vehicles need to be returned to the borrowed owning Rental Agency(s).
- In our current process & systems we currently use the following reason IDs and reason descriptions:

Transport Reason ID	*Transport Reason Description*
1	**Rental Drop off at different location**
2	**Vehicle Loaned to another Agency**
3	**Pick up from Distribution Center**
4	**Drop off to Distribution Center**
5	**Vehicle sent for maintenance**
7	**Unknown**

Note that transportation to and from Rental Agency is executed by an employee who is part of a transportation team or drivers. Therefore, when an employee executes a transport request of a vehicle to and from Rental Agencies, we need to capture the following information:

- *Transport pickup agency ID, Transport drop-off agency ID, Driver departure date, driver departure time, vehicle pick up date, vehicle pick up time, transport completed arrival date, transport completed arrival time, estimated arrival date, estimated arrival time,* & actual *transport time to completion.*

- In addition, we need to know at any time the transport status and transport status description of the transfer, such as: transfer completed, on route to pick up location, on route from pick up location, etc. Therefore, we need to capture the *Transport Status ID* or unique number that identifies a status and the *Transport Status Description*, or description of each status ID. Currently we track a transportation event using the following ID and description:

Transport Status ID	*Transport Status Description*
1	**Transport completed**
2	**On route to pick up location.**
3	**On route from pick up location**
4	**At pickup location. In progress (Loading etc.)**
5	**Pickup location delay**
7	**Unknown**

The goal again is to be able to know where our vehicles are located at any time and their status.

Business Requirements (Cont.)

Conclusion:

The business data listed in this business requirements document is what we need to capture for our business to operate. As our business evolve, additional data will be required in the future. We will address these new requirements in future versions of the application. For example, invoice processing & employee management at our rental agencies are features on our roadmap. Therefore, our expectations are that the design is modular and scalable for future growth.

Application Development & Technical Requirements

This section expands on the technical requirements that were leveraged for the development of the applications. There are detailed tables and explanations for all the technical requirements.

Application Development & Technical Requirements (Cont.)

Rental Agency Application Features and Functionalities Requirements:

The list of features and functionalities that we have compiled for the rental agencies' application are listed in the table below:

No.	Feature	Functionalities
1	**EZRental** Rental Agency Point-of-Sales (POS) System	▪ Car Rental, Car Return, New Customer Registration & Search/Print Customer Information, Customer Update, Customer Deletion, Customer Listing operations etc.
2	**EZRental** Rental Agency Back-Office Vehicle Inventory Management System	▪ Back-office system meant for employees to perform bulk IN-MEMORY inventory processing or management tasks on vehicles such as adding vehicles to the system, searching for vehicles, updating vehicles etc. ▪ This system is NOT meant for Point-of-Sales, but for the inventory management employees who need to search, add, remove etc., a large/bulk number of vehicles or employees during a session. ▪ Back-office vehicle Management features – Allows inventory personnel and employees to bulk-manage Cars, SUVs, Mini-Vans, Cargo Vans to be searched, added, removed, printed, listed etc.
3	**EZRental** Rental Agency Back-Office Credit Card Management System	▪ The EZRental Credit Card Management System is a Back-office system meant for the Credit Card Department Employees to manage Credit Card Information. These uses can Search/Print, Add, Edit & Delete credit card information in the database
4	**EZRental** Rental Agency Back-Office Employee & Customer User Account Management System	▪ The EZRental Customer & Employee User Account Management System is a Back-end system meant for IT ADMINISTRATOR Employees to manage both Employee & Customer USER ACCOUNTS.
5	**EZRental** Rental Agency Desktop Application Security Authentication System	▪ Proper security and authentication must be implemented to make sure only authorized employees can access the Point-Of-Sales, Back-End Management system or any other access to the applications.

Rental Agency Application Graphical User Interface Requirements:

- Graphical User-Interface should be fast rendering and user-friendly workflow.
- Visual screens or forms should be rich in color and appearance and navigation flow should be flexible and easy.
- The following UI controls or data field need to be pre-populated in GUI Screens:

  - **Addresses**

    - Any forms/UI which contains addresses, the STATE & COUNTRY fields should be automatically populated with a list of STATES or COUNTRIES, so the user does not have to manually enter a state or a country and simply select from drop-down list etc.

  - **Discount Codes:**

    - UI screens with customer's DISCOUNT CODE fields should be prepopulated with discount codes. The idea is the user should be able to select the discount to apply to a customer entry from a drop-down list/Combo Box etc. Note that this may or may not include the Discount Code Description on the UI screen as well.
    - Also note that the DISCOUNT CODE VALUES are generated by our Marketing Team and need to be pre-populated in the database before a code can be used. Therefore, the discount codes are prepopulated in the database.
    - Currently, when the Marketing Team generates a new code, they make the request to the database administrator to manually enter an update any new Discount Codes.
    - In the future, we want the application to have the necessary features for the Marketing Team to be able to manage the discount codes. This is not an immediate requirement out of the gate but should be targeted as part of a future upgrade.

Application Development & Technical Requirements

Rental Agency Application Graphical User Interface Requirements (Cont.):

- **EZPlus Rewards Codes:**
  - The EZPlus Reward UI screens with customer's EZPLUS REWARDS CODE fields should be prepopulated with the EZPlus Rewards code for the customer is being applied to. The idea is the user should be able to select the EZPLUS REWARD CODE to apply to a customer entry from a drop-down list/Combo Box etc. or be handled by the back-end database.
  - **Important:** The EZPLUS REWARDS CODE VALUES are NOT generated by a business entity in our organization, but AUTOMATICALLY GENERATED by the application on the fly when registering a new customer. This is a different approach compared to the DISCOUNT CODE which are generated by Marketing Team. In this case, the EZPlus Rewards Code values are generated by the application and available via the UI screen to be used or some other method of generation.
  - To finalize this requirement, the idea is the EZPlus Rewards Code should be automatically generated and either appear in the UI Screen or automatically generated in the database.

- **Company Name:**
  - UI screens with corporate customer's COMPANY NAME fields should be prepopulated with the list of corporations that are members of our corporate program, which enables our users to avoid having to manually enter the company name. Note that this may or may not include the Company ID in the UI Screen which is a unique number with business value that we assign to each company.
  - Note that the company names. Company ids and other company data are managed by our Corporate Sales Team and need to be pre-populated in the database before any corporate customer processing can be made. Therefore, the company information is prepopulated in the database.
  - Currently, when the Corporate Sales Team adds a new corporation or company into the program, they make the request to the database administrator to manually enter and add the new company to the database.
  - In the future we want the application to have the necessary features for the Corporate Sales Team to have the functionality to manage the data of our corporate companies via the application. This is not an immediate requirement out of the gate but should be targeted as part of a future upgrade.

- **Vehicle Status:**
  - UI screens for vehicle inventory management, VEHICLE STATUS field should be prepopulated with the list of vehicle status. Based on the business requirements, the current list of vehicle status is listed in table below:

Vehicle Status ID	*Vehicle Status Description*
1	**Reserved.**
2	**Rented.**
3	**Available.**
4	**Not available**
5	**Maintenance**
6	**Transferred to another agency**

  - Currently populating the database with a vehicle status record is handled manually by the database administrator. In the future we would like the application to have the necessary features for our business to be able to manage the vehicle status data. This is not an immediate requirement out of the gate but should be targeted as part of a future upgrade.

- **Rental Agency:**
  - UI screens that required adding or managing a RENTAL AGENCY field should be prepopulated with the list of rental agencies in our company.
  - Currently populating the database with a rental agency record is handled manually by the database administrator. In the future we would like the application to have the necessary features for our business to be able to manage the rental agency data. This is not an immediate requirement out of the gate but should be targeted as part of a future upgrade.

Application Development & Technical Requirements

Rental Agency Application Graphical User Interface Requirements (Cont.):

- **Vehicle Rental Category:**
  - UI screens that require the use of the VEHICLE RENTAL CATEGORY fields, must be prepopulated with the list of vehicle rental categories. Based on the business requirements, the current list of vehicle rental categories is as follows:

Vehicle Rental Category ID	*Vehicle Rental Category Name*	*Category Daily Rental Rate*
1	Car-Economic	$113.99
2	Car-Compact	$115.99
3	Car-Intermediate	$116.67
4	Car-Standard	$119.99
5	Car-Full Size	$121.99
6	Car-Premium	$127.79
7	Car-Luxury	$139.99
8	SUV-Intermediate	$127.99
9	SUV-Standard	$128.99
10	SUV-Standard Elite	$135.99
11	SUV-Full Size	$148.99
12	SUV-Premium	$157.99
13	Minivan-Standard	$152.99
14	Van-Passenger Van (12 passengers)	$161.00
15	Van-Cargo Van	$19.95
16	Pick Up-Mid Size	$69.95
17	Pick Up-Full Size	$105.99
18	Motorcycle-Touring	$19.95
19	Motorcycle-Cruiser	$199.99
20	Motorcycle-Scooter	$79.95

  - Currently populating the database with vehicle rental category records is handled manually by the database administrator. In the future we would like the application to have the necessary features for our business to be able to manage the vehicle rental categories data. This is not an immediate requirement out of the gate but should be targeted as part of a future upgrade.

- **Reservation Status:**
  - UI screens that require the use of the RESERVATION STATUS field, must be prepopulated with the list of reservation status data. Based on the business requirements, the current list of reservation status is as follows:

Reservation Status ID	*Reservation Status Description*
1	Confirmed.
2	Modified & reconfirmed.
3	Cancelled & Closed.
4	Fulfilled & Closed.
Etc..	Etc..

  - Currently populating the database with a reservation status record is handled manually by the database administrator. In the future we would like the application to have the necessary features for our business to be able to manage the reservation status data. This is not an immediate requirement out of the gate but should be targeted as part of a future upgrade.

Application Development & Technical Requirements

Rental Agency Application Graphical User Interface Requirements (Cont.):

- **Rental Status:**
  - UI screens that require the use of the RENTAL STATUS field, must be prepopulated with the list of rental status data. Based on the business requirements, the current list of rental status is as follows:

Rental Status ID	*Rental Status Description*
1	Picked up as scheduled.
2	Dropped off as scheduled.
3	Returned late
4	In progress.
5	Roadside assistance in progress.
7	Unknown

  - Currently populating the database with a rental status record is handled manually by the database administrator. In the future we would like the application to have the necessary features for our business to be able to manage the rental status data. This is not an immediate requirement out of the gate but should be targeted as part of a future upgrade.

- **Rental Fuel Option:**
  - UI screens that require the use of the RENTAL FUEL OPTION field, must be prepopulated with the list of rental fuel options data. Based on the business requirements, the current list of rental fuel option is as follows:

Rental Fuel Option ID	*Rental Fuel Option Description*	*Rental Fuel Option Additional Cost*
1	Return with a full tank or on return, pay for gas that is missing.	$13.97 (Important, this Decimal value of $13.97 is just an example, since the value is calculated during car return process and is based on the current price for a gallon of gas etc. therefore price will vary.)
2	Pay for full tank in advanced at time of rental, return car empty. No refund for unused gas.	$45.99 (Important, this Decimal value of $45.99 is just an example, since the value is calculated during car return process and is based on the current price for a gallon of gas etc. therefore price will vary.)

  - Currently populating the database with a rental fuel option record is handled manually by the database administrator. In the future we would like the application to have the necessary features for our business to be able to manage the rental fuel option data. This is not an immediate requirement out of the gate but should be targeted as part of a future upgrade.

Application Development & Technical Requirements

- **Rental Insurance Option:**

  - UI screens that require the use of the RENTAL INSURANCE OPTION field, must be prepopulated with the list of rental insurance options data. Based on the business requirements, the current list of rental insurance option is as follows:

Rental Insurance Option ID	*Rental Insurance Option Description*	*Rental Insurance Option Additional Cost per Day*
1	No insurance. Opt-out.	$0.00
2	Collision Damage Waiver Max - Agency will pay for damage, lost or stolen vehicle.	$49.99
3	Collision Damage Waiver 3000 - Agency will pay for first $3,000 of loss or damage, renter pays all loss & damage after $3,000.	$39.99
4	Lability Extended Protection – Agency provides renter with third party liability protection up to $1 Million per accident for bodily injury or death or property damage to others.	$89.99
5	Roadside Assistance Plus – 24/7 roadside assistance, replacement for lost keys, flat tire service, fuel delivery, etc.	$15.99

  - Currently populating the database with a rental insurance option record is handled manually by the database administrator. In the future we would like the application to have the necessary features for our business to be able to manage the rental insurance option data. This is not an immediate requirement out of the gate but should be targeted as part of a future upgrade.

Application Development & Technical Requirements

- **Transportation Reason Option:**
  - UI screens that require the user to populate the TRANSPORTATION OPTIONS field, must be prepopulated with the list of transportation reason options as shown in the table below:

Transport Reason ID	*Transport Reason Description*
1	Rental Drop off at different location
2	Vehicle Loaned to another Agency
3	Pick up from Distribution Center
4	Drop off to Distribution Center
5	Vehicle sent for maintenance
7	Unknown

  - Currently populating the database with a transportation reason option record is handled manually by the database administrator. In the future we would like the application to have the necessary features for our business to be able to manage the transportation reason option data. This is not an immediate requirement out of the gate but should be targeted as part of a future upgrade.

- **Transportation Reason Option:**
  - UI screens that require the user to populate the TRANSPORTATION STATUS field, must be prepopulated with the list of transportation status options as shown in the table below:

Transport Status ID	*Transport Status Description*
1	Transport completed
2	On route to pick up location.
3	On route from pick up location
4	At pickup location. In progress (Loading etc.)
5	Pickup location delay
7	Unknown

  - Currently populating the database with a transportation status option record is handled manually by the database administrator. In the future we would like the application to have the necessary features for our business to be able to manage the transportation status option data. This is not an immediate requirement out of the gate but should be targeted as part of a future upgrade

Application Development & Technical Requirements (Cont.)

Customer Facing Self-Service Web-Portal Application Architecture Requirements:

We now address architecture requirements for the application used in customers via the public internet to make reservations to rent a vehicle, modify their personal account, profile etc.:

1. Customer will use a secure and standard Web Application via a Browser to access our self-service portal in the internet. We need a website to support all customer self-service related transactions.
2. Web Application Architecture must be reusable and scalable to support future updates and new feature enhancements, without a long development lifecycle.
3. For this web development, we support *JavaScript*, *React*, *NodeJS* and other standard Web Technologies. In addition, the primary Application Development Platform we use is ***C# & .NET technologies***. We have aligned ***C# & .NET*** & **Web** developers that have been assigned to assist, support, operated and update the application once NYCTech consultants complete the project and development of this system.
4. Web Portal Security Authentication System – Proper security and authentication must be implemented to make sure only the customer can access the **EZRental.com** website for his or her profile home page.

Customer Facing Self-Service Web-Portal Features and Functionalities Requirements:

No.	Feature	Functionalities
1	**EZRental.com** Customer Web Portal	▪ Front-end WEB INTERFACE SCREENS & features used by customers via our web portal EZRentalCar.com to reserve a vehicle for rental and manage their account online. ▪ Features include search & reserve a car for rental, register as a new customer, search/view their account information, update their account etc.
2	**EZRental.com** Customer Web Portal Application Security Authentication System	▪ Proper security and authentication must be implemented to make sure only our customer can access the web portal to use the application.

Web Portal Application Web Pages User Interface Requirements:

The web pages graphical UI requirements are listed below:

- The GUI requirements for the web pages are like those functionalities of the Rental Agency Application that are found on the web site for example Search & reserve a car for rental, register as a new customer, search/view their account information, update their account etc.
- The design and graphics of the application should be appealing to customers and a smooth and fluent workflow.
- The following UI controls or data field need to be pre-populated in GUI Screens:

  - **Addresses**

    - Any web-page UI which contains addresses, the STATE & COUNTRY fields should be automatically populated with a list of STATES or COUNTRIES, so the user does not have to manually enter a state or a country and simply select from drop-down list etc.

  - **Discount Codes:**

    - Web pages with customer's DISCOUNT CODE fields should be a text box that allows the customer to ADD/APPLY the discount codes to redeem the coupon.

Application Development & Technical Requirements

Rental Agency Application Graphical User Interface Requirements (Cont.):

- **EZPlus Rewards Codes:**
  - The EZPlus Reward web page screens with customer's EZPLUS REWARDS CODE fields should be prepopulated with the EZPlus Rewards code for the customer is being applied to. The idea is the user should be able to select the EZPLUS REWARD CODE to apply to a customer entry from a drop-down list/Combo Box etc. or be handled by the back-end database.
  - **Important:** The EZPLUS REWARDS CODE VALUES are NOT generated by a business entity in our organization, but AUTOMATICALLY GENERATED by the application on the fly when registering a new customer. The EZPlus Rewards Code values are generated by the application and available via the UI screen to be used or some other method of generation.
  - To finalize this requirement, the idea is the EZPlus Rewards Code should be automatically generated and either appear in the UI Screen or automatically generated in the database.

- **Rental Agency:**
  - Web pages that required adding a RENTAL AGENCY field should be prepopulated with the list of rental agencies in our company.

- **Vehicle Rental Category:**
  - Web pages that require the use of the VEHICLE RENTAL CATEGORY fields, must be prepopulated with the list of vehicle rental categories. Based on the business requirements, the current list of vehicle rental categories is as follows:

Vehicle Rental Category ID	*Vehicle Rental Category Name*	*Category Daily Rental Rate*
1	Car-Economic	$113.99
2	Car-Compact	$115.99
3	Car-Intermediate	$116.67
4	Car-Standard	$119.99
5	Car-Full Size	$121.99
6	Car-Premium	$127.79
7	Car-Luxury	$139.99
8	SUV-Intermediate	$127.99
9	SUV-Standard	$128.99
10	SUV-Standard Elite	$135.99
11	SUV-Full Size	$148.99
12	SUV-Premium	$157.99
13	Minivan-Standard	$152.99
14	Van-Passenger Van (12 passengers)	$161.00
15	Van-Cargo Van	$19.95
16	Pick Up-Mid Size	$69.95
17	Pick Up-Full Size	$105.99
18	Motorcycle-Touring	$19.95
19	Motorcycle-Cruiser	$199.99
20	Motorcycle-Scooter	$79.95

Application Physical & Software Technical Architecture

Application Physical Architecture Overview

- During the design meetings with the application architects and full-stack developers a decision was made on the *physical application architecture* for the EZRental POS application.
- After a thorough review of both the ***business requirements*** and ***technical requirements*** by the project team, theresultant decisions on architecture (s) were based on the following:

Application Physical Architecture Overview

- During the design meetings with the application architects and full-stack developers a decision was made on the *physical application architecture* for the **EZRental POS** application.
- After a thorough review of both the ***business requirements*** and ***technical requirements*** by the project team, the resultant decisions on architecture (s) were based on the following:

  - **Rental Agency Employees:**
    - The system in our agencies used by the customer service representatives or front-line workers, must be able to quickly respond and execute the necessary requests such as

      - **POS Customer Management (Retail Customer & Corporate Customer) features** such as ***Customer Search & Print, New Customer Registration***, ***Customer Update***, ***Customer Deletion***, & ***Customer Listing functionalities***
      - POS Vehicle Reservation, Rental & Return Management Feature such as ***Vehicle Reservations, Vehicle Rental & Vehicle Return functionalities.***
      - POS Vehicle Inventory Management Feature allows inventory personnel and employees to bulk-manage vehicles such as Cars, SUVs, Mini-Vans, ***Cargo Vans,*** and other vehicles to be ***searched***, ***added***, ***updated***, ***deleted***, ***printed***, ***listed*** etc.
      - POS Credit Card Management Feature such as ***Credit Card Search & Print, New Credit Card Registration***, ***Credit Card Update***, ***Credit Card Deletion***, & ***Credit Card Listing functionalities.***

    - customer reservations, rentals, returns, customer management etc., therefore fast response and performance is required to quickly service a customer and minimize the wait. This is more important in Airports and other high-traffic locations.
    - We also want to provide our customer service agents with a rich user-interface experience.
    - The system in the agencies is also used by other back-end personnel such as vehicle inventory managers and administrators, service personnel, vehicle transport drivers, etc. Therefore, the system needs to also perform well.

- **Corporate Offices:**
  - The corporate offices are where our business operations are managed by our business employees & employees at the rental agencies via the INTRANET Web Portal.
  - These features include:

    - **Intranet Web Enterprise Resource Planning Systems (ERP) Portal Feature** such as providing access to **Enterprise Resource Planning Systems (ERP) Applications** such as: ***Customer Credit Card Management System***, ***Vehicle Inventory Management System***, ***Customer Relationship Management (CRM)***, ***Human Resource Management System***, & ***Finance & Operations System, Marketing System, Customer & Field Service System etc.***
    - **Web EZRental Point-of-Sales Corporate Management System** which allows employees to ***manage*** & ***execute*** **Point-of-Sales (POS)** transactions via the **Intranet Web Poral** such as: ***Search Customer Profile Information***, ***Customer Account Management***, ***Customer Registration***, ***Customer Update***, ***Customer Delete***, & ***Customer Listing functionalities, Manage & Make Reservations of a Vehicle***, ***Manage an existing Rental***, ***etc.***

  - The system should also perform well, but the performance requirements are not as stringent as our rental agencies for which the Corporate Web Intranet meets these requirements.

- **Customer self-service Web Portal:**
  - Customers who wish to make reservations and manage their reservations and rentals online via the internet, should be able to do so from anywhere in the world via an INTERNET Web Portal.
  - Features include:

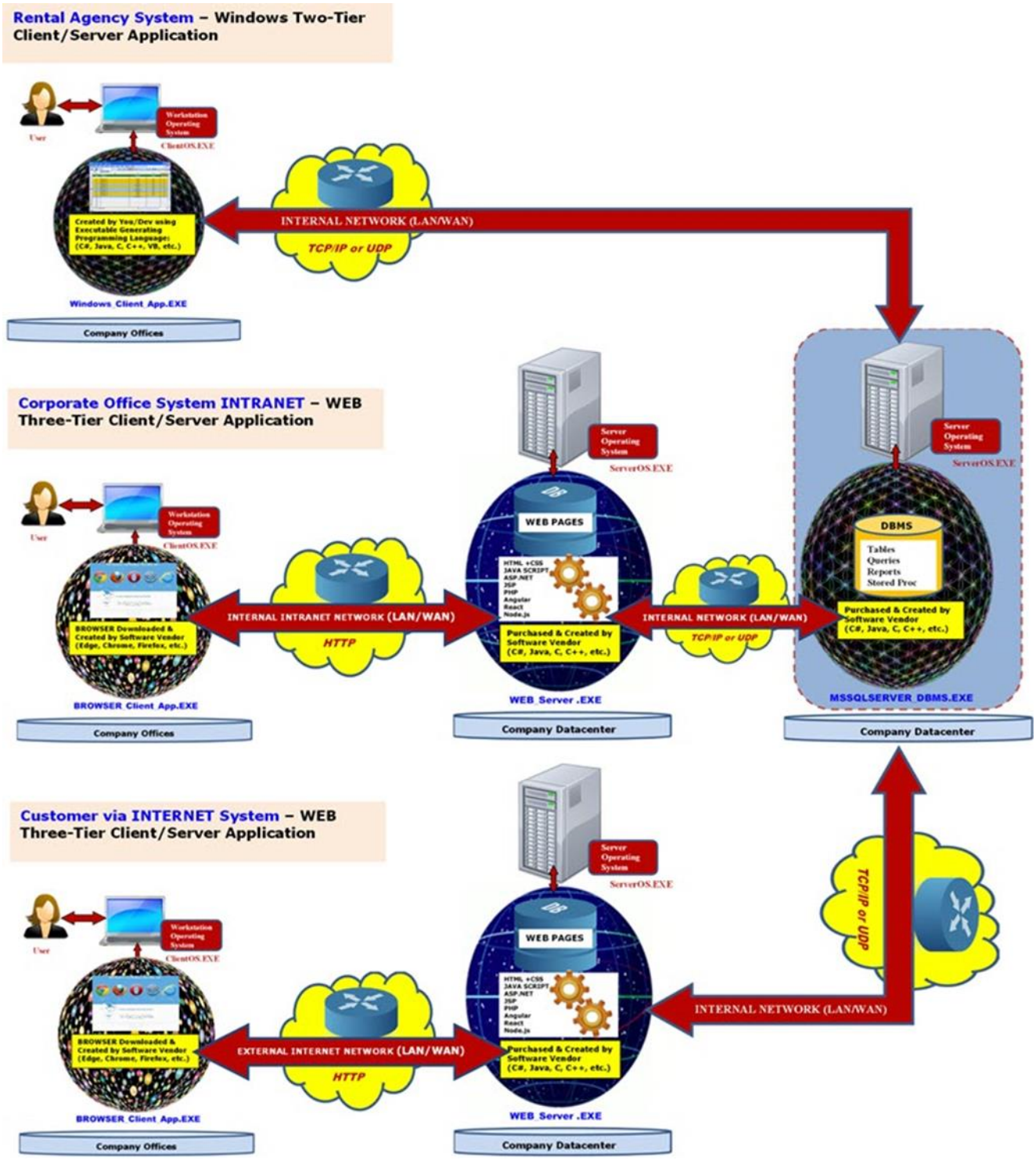
    - **Web Customer Facing EZRental Point-of-Sales System** which allows customers to ***manage*** & ***execute*** **Point-of-Sales (POS)** transactions online such as reservations & Profile online via a BROWSER using an **Internet Web Poral**. This includes functionality such as ***Search Customer Profile Information***, ***Customer Account Management***, ***Customer Registration***, ***Customer Update***, ***Customer Delete***, & ***Customer Listing functionalities, Manage & Make Reservations of a Vehicle***, ***Manage an existing Rental***, ***etc.***

  - The system should also have good user-experience and performance.

- Based on the above requirements and ideation, the derived target applications architecture and components are as follows:

  - Rental Agency Two-Tiered Windows Desktop Client/Server Application – Front-line workers such as customer service desk in store branches, airports etc., in addition to other support personnel such as service centers employees, inventory etc., are to use this Windows-based client application for speed and performance.
  - Corporate Office Three-Tiered Web-based Client/Server – This Web Application named EZRentalCorp.com, targeted for corporate business users in the corporate offices to manage the day-to-day business activities of our business and office workers personnel via a Browser Application.
  - Customer Internet Three-Tiered Web-based Client/Server – This Web Application named EZRental.com, targeted for customers who will reserve vehicles online via a Browser Application.
  - Database Tier supporting all Three Applications (Rental Agency, Corporate Office & Customer Internet) (***Two-Tier Window for agencies***, ***Three-tiered Web for Corporate Offices,*** and ***Three-Tier Web for Customers Internet application***) will SHARE the same **DATABASE TIER**.

Below is a pictorial diagram of this multi-component client/server architecture.



Application Development Feature & Functionality (Agile Backlog)

The application backlog represents a collection of pending tasks and feature requests, all were addressed by the development team. It served as a valuable resource for user needs and system enhancements and was carefully managed and executed in a systematic manner throughout the agile scrum process.

Feature #	Feature Description
FEATURE #1A	**FEATURE #1A – EZRental Rental Agency Point-of-Sales (POS) System CUSTOMER SERVICE – CUSTOMER MANAGEMENT SYSTEM:** 1. **WINDOWS CLIENT POINT-OF SALES SYSTEM** – ***WINDOWS UI FORM(S) FRONT-END APPLICATION & OOP PROGRAMMING*** features used by ***customer service representative employees*** via the ***Point-of-Sales computer*** machine in the ***Rental Agencies*** to service customer's **CUSTOMER MANAGEMENT** requests or transactions. 2. The following are features and functionality that are required for this application feature: 1. **POS Customer Management Feature:** **POS Customer Management (Retail Customer & Corporate Customer) features** such as ***Customer Search & Print, New Customer Registration***, ***Customer Update***, ***Customer Deletion***, & ***Customer Listing functionalities***. 2. **Note that each transaction is saved to database immediately after execution:** 1. **Feature UI Form Requirements:** ***Design & programming*** of required ***User-Interface Forms & GUI Controls*** to support this feature. 2. **Feature Processing Requirements:** ***Design & programming*** of required ***Object-Oriented (OOP) Processing & Logic*** to support this feature. 1. This feature is designed only to be used by customer service agents and other employees using the **Windows Two-Tiered Client/Server Application** in the **Rental Agencies**.
FEATURE #1B	**FEATURE #1B – EZRental Rental Agency Point-of-Sales (POS) Customer Management System Back-end Database Design & Implementation to support this feature:** 1. **DATABASE SERVER BACK-END SYSTEM – *BACK-END DATABASE DESIGN & FEATURES* (Create the Tables & Relationships, pre-populated tables, stored procedures, views, indexes etc.,) to support this feature.**

Feature #	Feature Description
FEATURE #2A	**FEATURE #2A – EZRental Rental Agency Point-of-Sales (POS) System CUSTOMER SERVICE VEHICLE RESERVATION, RENTAL & RETURN FEATURE MANAGEMENT:** 2. **WINDOWS CLIENT POINT-OF SALES SYSTEM** – ***WINDOWS UI FORM(S) FRONT-END APPLICATION & OOP PROGRAMMING*** features used by ***customer service representative employees*** via the ***Point-of-Sales computer*** machine in the ***Rental Agencies*** to service customer's **CUSTOMER VEHICLE RESERVATION, RENTAL & RETURN MANAGEMENT,** or transactions. 3. The following are features and functionality are required for this application feature: 4. **POS Vehicle Reservation, Rental & Return Management Feature: POS Customer Vehicle Reservation, Rental & Return Management (Retail Customer & Corporate Customer) features** such as ***Vehicle Reservations, Vehicle Rental & Vehicle Return functionalities:*** 5. **Feature UI Form Requirements:** ***Design & programming*** of required ***User-Interface Forms & GUI Controls*** to support this feature. 6. **Feature Processing Requirements:** ***Design & programming*** of required ***Object-Oriented (OOP) Processing & Logic*** to support this feature. 7. This feature is designed only to be used by customer service agents and other employees using the **Windows Two-Tiered Client/Server Application** in the **Rental Agencies**.
FEATURE #2B	**FEATURE #2B – EZRental Rental Agency Point-of-Sales (POS) Customer Vehicle Reservation, Rental & Return Management System Back-end Database Design & Implementation to support this feature:** 8. **DATABASE SERVER BACK-END SYSTEM – *BACK-END DATABASE DESIGN & FEATURES* (Create the Tables & Relationships, pre-populated tables, stored procedures, views, indexes etc.,) to support this feature.**

Feature #	Feature Description
FEATURE #3A	**FEATURE #3A – EZRental Internal Back-Office Agency BACK-OFFICE VEHICLE INVENTORY MANAGEMENT SYSTEM (NOT A CUSTOMER FACING APPLICATION):** 9. **WINDOWS CLIENT POINT-OF SALES SYSTEM** – ***WINDOWS UI FORM(S) FRONT-END APPLICATION & OOP PROGRAMMING*** features used by **Back-Office Inventory Team employees** via a **computer machine** in the ***Rental Agencies*** to service inventory needs for **VEHICLE INVENTORY MANAGEMENT,** or transactions. *10.* This is a unique Back-end system meant for inventory team employees to perform bulk **IN-MEMORY** inventory processing or management tasks on vehicles such as: ***adding*** vehicles to the system, ***searching*** for vehicles, ***updating*** vehicles, ***deleting*** vehicles, etc. The idea is that the employee can perform all these features on their computer **in-memory** repeatedly for several vehicles in one session saving to database after each transaction but managed in-memory using a **collection** or **other data structure** to manage it locally. When user is done with all inventory transactions, all transactions have been saved to database but, is still locally in the collection or other data structure and can be updated as needed. By keeping it locally in memory, the operations are faster. 11. The following are features and functionality are required for this application feature: 12. **POS Vehicle Inventory Management Feature:** **POS Vehicle Inventory Management features** allows inventory personnel and employees to bulk-manage vehicles such as **Cars**, **SUVs, Mini-Vans**, ***Cargo Vans,*** and other vehicles to be ***searched***, ***added***, ***updated***, ***deleted***, ***printed***, ***listed*** etc. 13. **Feature UI Form Requirements:** ***Design & programming*** of required ***User-Interface Forms & GUI Controls*** to support this feature. 14. **Feature Processing Requirements:** ***Design & programming*** of required ***Object-Oriented (OOP) Processing & Logic*** to in the to support this feature. 15. This back-office features are not designed to be used by customers and not available via the Web and implemented using the **Windows Two-Tiered Client/Server Application** in the **Rental Agencies**.
FEATURE #3B	**FEATURE #3B – EZRental Rental Agency Vehicle Inventory Management System Back-end Database Design & Implementation to support this feature:** 16. **DATABASE SERVER BACK-END SYSTEM – *BACK-END DATABASE DESIGN & FEATURES* (Create the Tables & Relationships, pre-populated tables, stored procedures, views, indexes etc.,) to support this feature.**

Feature #	Feature Description
FEATURE #4A	**FEATURE #4A – EZRental Rental Agency Point-of-Sales (POS) BACK-OFFICE CREDIT CARD MANAGEMENT SYSTEM:** 17. **WINDOWS CLIENT POINT-OF SALES SYSTEM** – ***WINDOWS UI FORM(S) FRONT-END APPLICATION & OOP PROGRAMMING*** features is a **back-end system** used by ***customer service representative & other employees*** via the ***Point-of-Sales computer*** machine in the ***Rental Agencies*** to service customer's **CREDIT CARD MANAGEMENT,** or transactions required when servicing customers. 18. The following are features and functionality are required for this application with features such as: 19. **POS Credit Card Management Feature:** **POS Customer Credit Card Management features** such as ***Credit Card Search & Print, New Credit Card Registration***, ***Credit Card Update***, ***Credit Card Deletion***, & ***Credit Card Listing functionalities:*** 20. **Feature UI Form Requirements:** ***Design & programming*** of required ***User-Interface Forms & GUI Controls*** to support this feature. 21. **Feature Processing Requirements:** ***Design & programming*** of required ***Object-Oriented (OOP) Processing & Logic*** to support this feature. 22. This feature is designed only to be used by customer service agents and other employees using the **Windows Two-Tiered Client/Server Application** in the **Rental Agencies**.
FEATURE #4B	**FEATURE #4B – EZRental Rental Agency Point-of-Sales (POS) Credit Card Management System Back-end Database Design & Implementation to support this feature:** 23. **DATABASE SERVER BACK-END SYSTEM – *BACK-END DATABASE DESIGN & FEATURES* (Create the Tables & Relationships, pre-populated tables, stored procedures, views, indexes etc.,) to support this feature.**

Feature #	Feature Description
FEATURE #5A	**FEATURE #5A – EZRental Rental Agency Point-of-Sales (POS) System BACK-OFFICE EMPLOYEE & CUSTOMER USER-ACCOUNT MANAGEMENT SYSTEM:** 24. **WINDOWS CLIENT POINT-OF SALES SYSTEM** – ***WINDOWS UI FORM(S) BACK-OFFICE APPLICATION & OOP PROGRAMMING*** **User-Accounts Management Features** used by ***customer service representative & IT Administrator employees*** via the ***Point-of-Sales computer*** machine in the ***Rental Agencies*** to service customer's **CUSTOMER & EMPLOYEE USER ACCOUNT MANAGEMENT** requests or transactions. 25. **Employee User Accounts –** These are the user accounts used by ***IT Administrators***, ***Customer Service Employees***, ***back-office employees***, and ***any employee who qualifies*** for access to the system. 26. **Customer User Accounts –** These are the user accounts used by ***IT Administrators***, ***Customer Service Employees***, ***back-office employees*** and any ***employee*** who has access to the system to ***manage*** the **Customer User Accounts** for login into the **Customer Web Portal**. 27. The following are features and functionality that are required for this application feature: 28. **POS User Account (Employee & Customer) Management Feature:** **POS User Account Management (Employee & Customer) features** such as: 1. **Employee User Account Feature 5A-1 –** Allows ***IT Administrators***, ***Customer Service Employees***, ***back-office employees***, and ***any employee who qualifies*** to ***manage*** employee user accounts that allow employees to login into the POS System. And perform the following tasks: ***Employee User Account Search by username, New Employee User Account Registration***, ***Employee User Account Update by username***, ***Employee User Account Deletion by username***, & ***Employee User Account Listing functionalities***. **IMPORTANT!** Note that the **password** is ***NEVER DISPLAYED or LISTED***, only the **username**! 2. **Customer User Account Feature 5A-2 –** Allows ***IT Administrators***, ***Customer Service Employees***, ***back-office employees***, and ***any employee who qualifies*** to ***manage*** customer user accounts that allow customers to login into the **Customer Web Portal System**. And perform the following tasks: ***Customer User Account Search by username, New Customer User Account Registration***, ***Customer User Account Update by username***, ***Customer, User, Account Deletion, deletion by username***, & ***Customer User Account Listing functionalities***. **IMPORTANT!** Note that the **password** is ***NEVER DISPLAYED or LISTED***, only the **username**! 3. ***Note that each transaction is saved to database immediately after execution:*** 4. **Feature UI Form Requirements:** ***Design & programming*** of required ***User-Interface Forms & GUI Controls*** to support this feature. 5. **Feature Processing Requirements:** ***Design & programming*** of required ***Object-Oriented (OOP) Processing & Logic*** to support this feature. 6. This feature is designed only to be used by ***IT Administrations*** and other employees who qualify to use this system to manage both employees & customers user accounts using the **Windows Two-Tiered Client/Server Application** in the **Rental Agencies**.

FEATURE #5B	**FEATURE #5B – EZRental Rental Agency Point-of-Sales (POS) User EMPLOYEE User Account Management System Back-end Database Design & Implementation to support this feature:** 7. **DATABASE SERVER BACK-END SYSTEM – *BACK-END DATABASE DESIGN & FEATURES* (Create the Tables & Relationships, pre-populated tables, stored procedures, views, indexes etc.,) to support this feature to store and manage EMPLOYEE USER ACCOUNTS.**
FEATURE #5C	**FEATURE #5C – EZRental Rental Agency Point-of-Sales (POS) User CUSTOMER User Account Management System Back-end Database Design & Implementation to support this feature:** 8. **DATABASE SERVER BACK-END SYSTEM – *BACK-END DATABASE DESIGN & FEATURES* (Create the Tables & Relationships, pre-populated tables, stored procedures, views, indexes etc.,) to support this feature to store and manage CUSTOMER USER ACCOUNTS.**

Feature #	Feature Description
FEATURE #6A	**FEATURE #6A – EZRental Rental Agency Point-of-Sales (POS) System EMPLOYEES BACK-OFFICE SECURITY LOGIN AUTHENTICATION SYSTEM:** 9. Proper ***security and authentication*** must be implemented to make sure only authorized **employees** can access the **Point-0f- Sales & Back-End Management systems** when they **login** into the **Windows Two-Tiered Client/Server Application** & **Web Three-Tiered Corporate Client/Server Application**. 10. **WINDOWS CLIENT POINT-OF SALES SYSTEM** – ***WINDOWS UI FORM(S) BACK-OFFICE APPLICATION & OOP PROGRAMMING*** **Login Authentication features** used by ***customer service representative & IT Administrator employees*** via the ***Point-of-Sales computer*** machine in the ***Rental Agencies*** to service employee **LOGIN AUTHENTICATION SYSTEM**. 11. The following are features and functionality that are required for this application such as: 12. **POS Employee Back-Office Security Login Authentication System:** **POS Login Authentication Access for Employees features** such as: 13. **Employee Authentication Feature 6A-1 –** To have access to the application, an employee (Customer Service Reps, Back-office employee etc.) must provide a **username & password**. This feature is required to be ***designed*** & ***programmed*** into the application. 14. **Employee Authentication Feature 6A-2 –** ***Design & programming*** of required **User-Interface Forms** & **GUI Controls** to support the **Authentication System feature**! 15. Programming includes: 16. **Feature UI Form Requirements:** ***Design & programming*** of required ***User-Interface Forms & GUI Controls*** to support this feature. 17. **Feature Processing Requirements:** ***Design & programming*** of required ***Object-Oriented (OOP) Processing & Logic*** to support this feature. 18. This feature is designed only to be used by all employees wishing access to the **Windows Two-Tiered Client/Server Application POS System** in the **Rental Agencies**.
FEATURE #6B	**FEATURE #6B – EZRental Rental Agency Point-of-Sales (POS) Employee Back-Office Security Login Authentication System Back-end Database Design & Implementation to support this feature:** 19. **DATABASE SERVER BACK-END SYSTEM – *BACK-END DATABASE DESIGN & FEATURES* (Create the Tables & Relationships, pre-populated tables, stored procedures, views, indexes etc.,) to support this feature.**

Feature #	Feature Description
FEATURE #7A	**FEATURE #7A – EZRental INTERNAL CORPORATE EMPLOYEE & RENTAL AGENCIES EMPLOYEES INTRANET CORPORATE BUSINESS APPLICATIONS WEB PORTAL:** 20. This **INTRANET (NOT THE PUBLIC INTERNET) Web Portal** EZRentalHub.com, is a **Web-based Three-Tiered Client/Server physical & Software Development Architecture** used by **CORPORATE & OTHER EMPLOYEES** to ***execute*** **Enterprise Resource Planning Systems (ERP) Applications & other Corporate Applications** online intranet via a **BROWSER**. 21. **BROWSER/INTRANET WEB ERP & CORPORATE APPLICATION SYSTEM** – ***WEB UI FORM(S) FRONT-END APPLICATION & OOP PROGRAMMING*** **Enterprise Resource Planning Systems (ERP) Applications & other Corporate Applications Features** used by ***Corporate Employees*** via the **CORPORATE INTRANET** ***PORTAL***. 22. The following are features and functionality that are required for this **INTRANET WEB APPLICATION**: 23. **Corporate Business Application Intranet Web Portal Features:** 24. **Intranet Web Enterprise Resource Planning Systems (ERP) Portal Feature 7A-1 –** Provides access to **Enterprise Resource Planning Systems (ERP) Applications** such as: ***Customer Credit Card Management System***, ***Vehicle Inventory Management System***, ***Customer Relationship Management (CRM)***, ***Human Resource Management System***, & ***Finance & Operations System, Marketing System, Customer & Field Service System etc***. 25. **Web EZRental Point-of-Sales Corporate Management Feature 7A-2 –** Allows Employees to ***manage*** & ***execute*** **Point-of-Sales (POS)** transaction via the **Intranet Web Poral** such as: ***Search Customer Profile Information***, ***Customer Account Management***, ***Customer Registration***, ***Customer Update***, ***Customer Delete***, & ***Customer Listing functionalities, Manage & Make Reservations of a Vehicle***, ***Manage an existing Rental***, ***etc.*** 26. Programming includes: 27. **Feature UI Form Requirements:** ***Design & programming*** of required ***INTRANET WEB*** *User-Interface Forms & GUI Controls* to support this feature. 28. **Feature Processing Requirements:** ***Design & programming*** of required ***INTRANET WEB TECHONLOGY***, ***Object-Oriented (OOP) Processing & Logic PROCESSING*** to support this feature. 29. This feature is designed only to be used by ***CORPORATE EMPLOYEES*** in the **Corporate Offices** & ***RENTAL AGENCIES EMPLOYEES*** via the **INTRANET** using the **Web Three-Tiered Client/Server Application**.
FEATURE #7B	**FEATURE #7B – EZRental Corporate Employee & Rental Agencies Employees INTRANET WEB PORTAL System Back-end Database Design & Implementation to support this feature:** 30. **DATABASE SERVER BACK-END SYSTEM – *BACK-END DATABASE DESIGN & FEATURES*** **(Create the Tables & Relationships, pre-populated tables, stored procedures, views, indexes etc.,) to support this feature.**

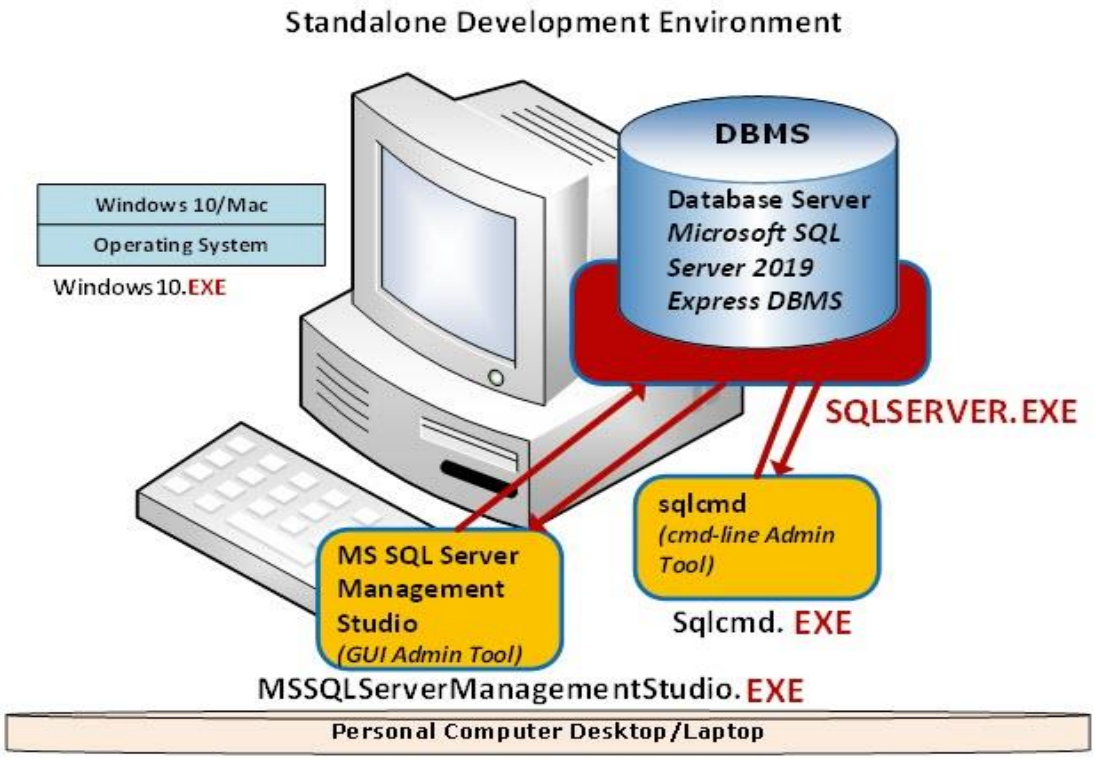
Feature #	Feature Description
FEATURE #8A	**FEATURE #8A – EZRental EXTERNAL CUSTOMER SELF-SERVICE INTERNET CUSTOMER FACING POINT-OF-SALES (POS) WEB PORTAL:** 31. This **Web Portal** EZRental.com, is a **Web-based Three-Tiered Client/Server physical & Software Development Architecture** used by **CUSTOMERS** to ***manage*** & ***make*** **reservations** online via a **BROWSER**. 32. **BROWSER/WEB CUSTOMER POINT-OF SALES SYSTEM** – ***WEB UI FORM(S) FRONT-END APPLICATION & OOP PROGRAMMING*** **Point-of-Sales (POS) Management Features** used by ***customers*** via the **INTERNET** ***Point-of-Sales PORTAL*** via their computers/laptop/tables/Mobile to **MAKE RESERVATIONS ONLINE & MANAGE THEIR RENTAL**. 33. The following are features and functionality that are required for this **WEB APPLICATION** feature: 34. **POS Reservation & Management Features:** 35. **Web POS Authentication System Feature 8A-1 –** Proper **security and authentication** must be implemented to make sure **only the authorized customer can access to its Point-0f-Sales portal and login and out of their profile website**. 36. **Web POS Customer Self-Service Management Feature 8A-2 –** Allows POS Customer (Retail Customer & Corporate Customer) Self-Service Management of their account such as: ***Customer Profile Information***, ***Customer Account & Login Registration***, ***Customer Update Profile***, ***Customer Delete Profile***, & ***Customer Listing functionalities such as listing of Reservations & Rental History etc***. 37. **Web POS Customer Self-Service Point-of-Sales Management Feature 8A-3 –** Allows POS Customer (Retail Customer & Corporate Customer) Self-Service features to make reservations and manage rentals such as: ***Make Reservations of a Vehicle***, ***Manage an existing Rental***, ***etc***. 38. **Web POS User Account Management Feature 8A-4 –** Allows POS Customer (Retail Customer & Corporate Customer) Self-Service features to enable customer to manage its **User Account** and perform the following operations: ***Reset Username*** & ***Reset Password.*** 39. Programming includes: 40. **Feature UI Form Requirements:** ***Design & programming*** of required ***WEB*** *User-Interface Forms & GUI Controls* to support this feature. 41. **Feature Processing Requirements:** ***Design & programming*** of required ***WEB TECHONLOGY***, ***Object-Oriented (OOP) Processing & Logic PROCESSING*** to support this feature. 42. This feature is designed only to be used by ***CUSTOMERS*** via the **INTERNET** using the **Web Three-Tiered Client/Server Application** from their **Personal Computer/Mobile Devices via the INTERNET**.
FEATURE #8B	**FEATURE #8B – EZRental Customer Serf-Service internet Customer facing Point-of-Sales (POS) WEB PORTAL System Back-end Database Design & Implementation to support this feature:** 43. **DATABASE SERVER BACK-END SYSTEM – *BACK-END DATABASE DESIGN & FEATURES* (Create the Tables & Relationships, pre-populated tables, stored procedures, views, indexes etc.,) to support this feature.**

Feature #	Feature Description
FEATURE #9A	**FEATURE #9A – EZRental Customer Point-of-Sales (POS) System CUSTOMER FACING WEB PORTAL SECURITY LOGIN AUTHENTICATION SYSTEM:** 44. Proper ***security and authentication*** must be implemented to make sure only authorized **customers** can access their **Self-Service-Point-Of-Sales Web Portal systems** when they **login** into the **Web Three-Tiered Customer Client/Server Application** via the **INTERNET**. 45. **CUSTOMER SELF-SERVICE POINT-OF SALES SYSTEM** – ***WEB UI FORM(S) FRONT-END APPLICATION & OOP PROGRAMMING*** for **Customer Login Authentication features** used by ***customer service representative & IT Administrator employees*** to service **CUSTOMER LOGIN AUTHENTICATION SYSTEM**. 46. The following are features and functionality that are required for this application such as: 47. **Customer Self-Service Web Portal Security Login Authentication System.** **Self-Service Web Portal Login Authentication Access for Customer features** such as: 48. **Customer Authentication Feature 9A-1 –** To have access to their Self-Service Web Portal Application, a customer must provide a **username & password**. This feature is required to be ***designed*** & ***programmed*** into the application. 49. **Customer Authentication Feature 9A-2 –** ***Design & programming*** of required **Web User-Interface Forms** & **GUI Controls** to support the **Web Portal Authentication System feature**! 50. Programming includes: 51. **Feature Web UI Form Requirements:** ***Design & programming*** of required ***User-Interface Forms & GUI Controls*** to support this feature. 52. **Feature Web Processing Requirements:** ***Design & programming*** of required ***Object-Oriented (OOP) Processing & Logic*** to support this feature. 53. This feature is designed only to be used by customers wishing access to their **Self-Service Web Portal Three-Tiered Client/Server Application** via the **INTERNET**.
FEATURE #9B	**FEATURE #9B – EZRental Customer Point-of-Sales (POS) System CUSTOMER FACING WEB PORTAL SECURITY LOGIN AUTHENTICATION SYSTEM Back-end Database Design & Implementation to support this feature:** 54. **DATABASE SERVER BACK-END SYSTEM – *BACK-END DATABASE DESIGN & FEATURES*** **(Create the Tables & Relationships, pre-populated tables, stored procedures, views, indexes etc.,) to support the CUSTOMER facing authentication features.**

Feature #	Feature Description
FEATURE #10A	**FEATURE #10A – EZRental INTERNAL CORPORATE EMPLOYEE & RENTAL AGENCIES EMPLOYEES INTRANET BACK-OFFICE VEHICLE TANSPORT MANAGEMENT SYSTEM WEB PORTAL:** 55. This **INTRANET Web Portal** EZRentalHub.com, is a **Web-based Three-Tiered Client/Server physical & Software Development Architecture** used by **CORPORATE & AGENCY EMPLOYEES** to ***manage*** **Transportation of Vehicles by Employee Drivers to and from Rental Agencies, Vehicle Distribution Centers, and other Locations** via a **BROWSER**. 56. **WEB TRANSOPORT MANAGEMENT SYSTEM APPLICATION** – ***WEB UI FORM(S) FRONT-END APPLICATION & OOP PROGRAMMING*** **Transport Management Features** used by ***Vehicle Transportation Managers & Drivers Employees*** to handle the day-to-day vehicle transportation process via the **CORPORATE INTRANET** *PORTAL*. 57. The following are features and functionality that are required for this **INTRANET Transport Management WEB APPLICATION**: 58. **Corporate Vehicle Transport Application Intranet Web Portal Features:** 59. **Transport Scheduling Feature –** handle the day-to-day creating & scheduling of a pic-up & delivery (Any vehicle type) such as: ***Creation of NEW Vehicle Transport Request***, ***Vehicle Pick-up***, ***Vehicle Drop-off*** & ***Vehicle Transport Status etc***. 60. Programming includes: 61. **Feature UI Form Requirements:** ***Design & programming*** of required ***INTRANET WEB*** *User-Interface Forms & GUI Controls* to support this feature. 62. **Feature Processing Requirements:** ***Design & programming*** of required ***INTRANET WEB TECHONLOGY***, ***Object-Oriented (OOP) Processing & Logic PROCESSING*** to support this feature. 63. This feature is designed only to be used by ***CORPORATE EMPLOYEES*** in the **Corporate Offices** & ***RENTAL AGENCIES EMPLOYEES*** via the **INTRANET** using the **Web Three-Tiered Client/Server Application**.
FEATURE #10B	**FEATURE #10B – EZRental Corporate Vehicle Transport Application Intranet Web Portal Features System Back-end Database Design & Implementation to support this feature:** 64. **DATABASE SERVER BACK-END SYSTEM – *BACK-END DATABASE DESIGN & FEATURES*** **(Create the Tables & Relationships, pre-populated tables, stored procedures, views, indexes etc.,) to support this feature.**

Database Management System Development Environment & Physical Architecture

The Database Tier will continue to use MS SQL SERVER COMMUNITY EDITTION standard as per the requirements gathered during the planning phase, the database development environment is visually represented in the image below:



Project Roles & Responsibilities

From the ***business requirements*** captured during discovery & analysis, the **Business Analyst/Architect** following database design principles and the ***Project Management Methodology*** phases and deliverable determined the following skills/roles required to successfully complete the project at the required timelines:

1. Consultant #1 – Business/Database Analyst/Architect – *Created* the Conceptual Model (ER/EER), Logical Model & Normalized Logical Model from the *Business Requirements*.

2. Consultant #2 – Database Developer – *Designed & Created* a **Data Dictionary** with the required **METADATA** for all the Logical Tables & their Columns/Attributes from the *Normalized Logical Model Diagram* targeting **Microsoft SQL Server DBMS DATA TYPES**
3. Consultant #3 – Database Developer – *Designed & Created* the **Physical Schema Design Diagram** for the **DATABASE** you will implement,
4. Consultant #4 – Database Developer – *Implement* the **DBMS Application** from the **Physical Schema Design Diagram**
5. Consultant #5 – Database Developer – *Test* the **DBMS Application**
6. Consultant #6 – Database Administrator (DBA) – *Manage & Operate* the **DBMS Application** for future upgrades.

Summary of the DBMS Server Application Development Roles and Responsibilities

The **Business/Database Analyst hired by Mr. Rodriguez** who assembled the required database development team, and the table below describes each of the roles and the individual (s) that will execute the role:

Person	Role	Description
Prof. Rodriguez	**Program Manager, AgileScrum Master & ProjectManager**	▪ Owner of the project and liaison to Manage the **EZRental Inc**., the customer. ▪ Activities include but not limited to. 1. Owner of project responsible for the success of the project. 2. Project Management 3. Scrum Master ensures the project stays on time and moves in the right direction. Clear any obstacles impeding the team's progress etc.
Consultant #1: **Prof. Rodriguez**	**Business & DatabaseAnalyst**	▪ A **Business/Database Analyst was hired by Prof. Rodriguez** to interview the stakeholders at **EZRental Inc**. And create the ***Business Requirements*** that will be the foundation to the database design & implementation. ▪ Activities included but not limited to: 1. Engage in discovery activities & interview the stakeholders at **EZRental Inc**. 2. From Resulting interviews and discoveries created: **1) ER/EER ConceptualData Model from the business requirements &** **2) Normalized Logical Model**.
Consultant #2, 3, 4 & 5 **Darwhin Gomez**	**Database Developers**	▪ This role uses the Normalized Logical Model created by consultant #2 to create the Data Dictionary, Physical Schema Diagram, and Implement the Database Application forthe Auto Rental System. ▪ Activities included but not limited to: 1. Use the ***Normalized Logical Model*** created by consultant #2 to do the **following:** 1) **Create Data Dictionary tables for eachlogical table targeting MS SQL Server** 2) **Create Physical Schema Diagram**. 2. From these two deliverables, 1) **implement the Database Application using Oracle 18** for the Auto Rental System.
Consultant #6 **Darwhin Gomez**	**Database Administrator**	▪ The DB Admin, install the DBMS, maintain, and operate the DBMS throughoutits lifetime. ▪ Activities included but not limited to: **1) Setup & install MS SQL Server** **2) Administrative tools for target DBMS**. 1. **3) Operate & Maintain the DBMS**.

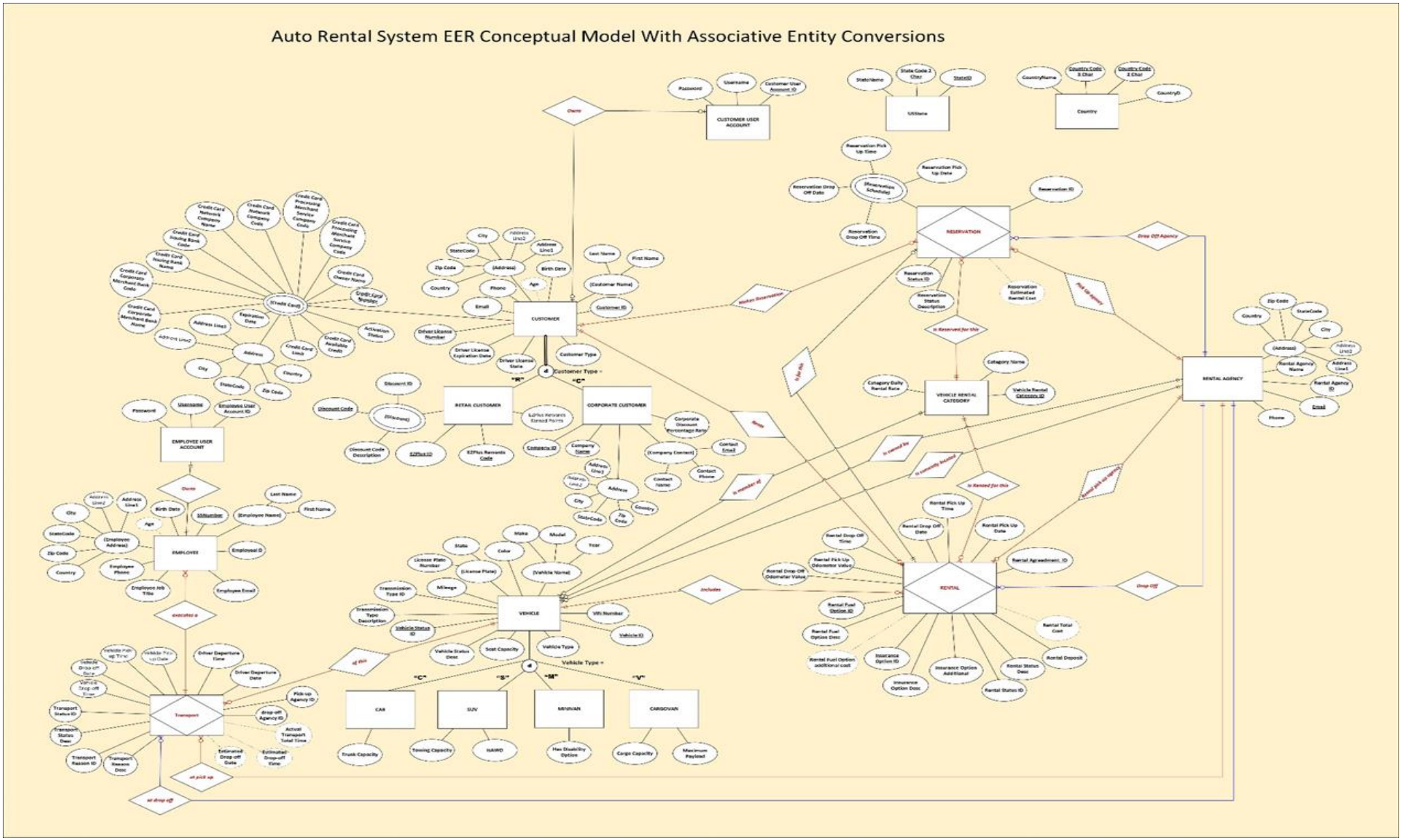
Summary of the Windows Client & Browser Client Applications Development Project Roles and Responsibilities

The **Application Full Stack OOP Architect/Analyst** hired by Mr. Rodriguez aligned the required application development team and the table below describes each of the roles and the individual (s) that will execute the roles:

Person	Role	Description
Consultant #7 & 13 **Mr. Rodriguez**	**Full Stack Object-Oriented-Programming Architect**	▪ An Object-Oriented-Programming Architect was hired by Prof. Rodriguez to interview the stakeholders at **EZRental Inc**. and derive the **Application Technical Requirements** in addition to designing the **Class/Object Model Architecture**. This also includes the planning and designing both **Windows Client Application** and the **Web Browser Application**. ▪ Activities include but not limited to: 1. Engage in discovery activities & interview the stakeholders at **EZRental Inc**. 2. From the interview and discovery **1) Design/Architect the Object-Oriented-Programming Class/Object Model for the Windows Client Application**. 3. **Design/Architect the Object-Oriented-Programming Class/Object Model for the Web Browser Application**.
Consultants #8, 9, 10, 11 & 12 **Darwhin Gomez**	**Full Stack Windows Application Developers & UI/UX Client Application Developer**	▪ Object-Oriented-Programming developer to implement the **Windows Client Application** using **C# & .NET technologies** & on the database side, implement stored procedures and support the databased team as needed. ▪ Activities included but not limited to: 1. As full stack developer, **Programming & implementation of theObject-Oriented-Programming of Class/Object Model designed by consultant #7 for the Windows Client Application using C# & .NET Technologies**. 2. In addition, **Development of Database Stored Procedures,and other development requirements in the Back-end DBMS**. 3. From the technical requirements, **design a high-level GraphicalUser-Interface (GUID) wireframe**, & **implement the front-end UI Programming, features & functionality**
Consultant #14, 15, 16, 17 & 18 **Darwhin Gomez**	**Full Stack Web Application Developer & UI/UX Web Application Developer**	▪ Object-Oriented-Programming developer to implement the **Web Browser Application** using **C# & ASP.NET technologies**. ▪ Activities included but not limited to: 1. As full stack developer, **Programming & implementation of theObject-Oriented-Programming of Class/Object Model designed by consultant #7 for the Web Browser Client Application using C# & ASP.NET Technologies**. 2. From the technical requirements, **design a high-level Graphical User-Interface (GUID) wireframe**, & **implement the Webfront-end UI Programming, features & functionality in the Web Server Application**

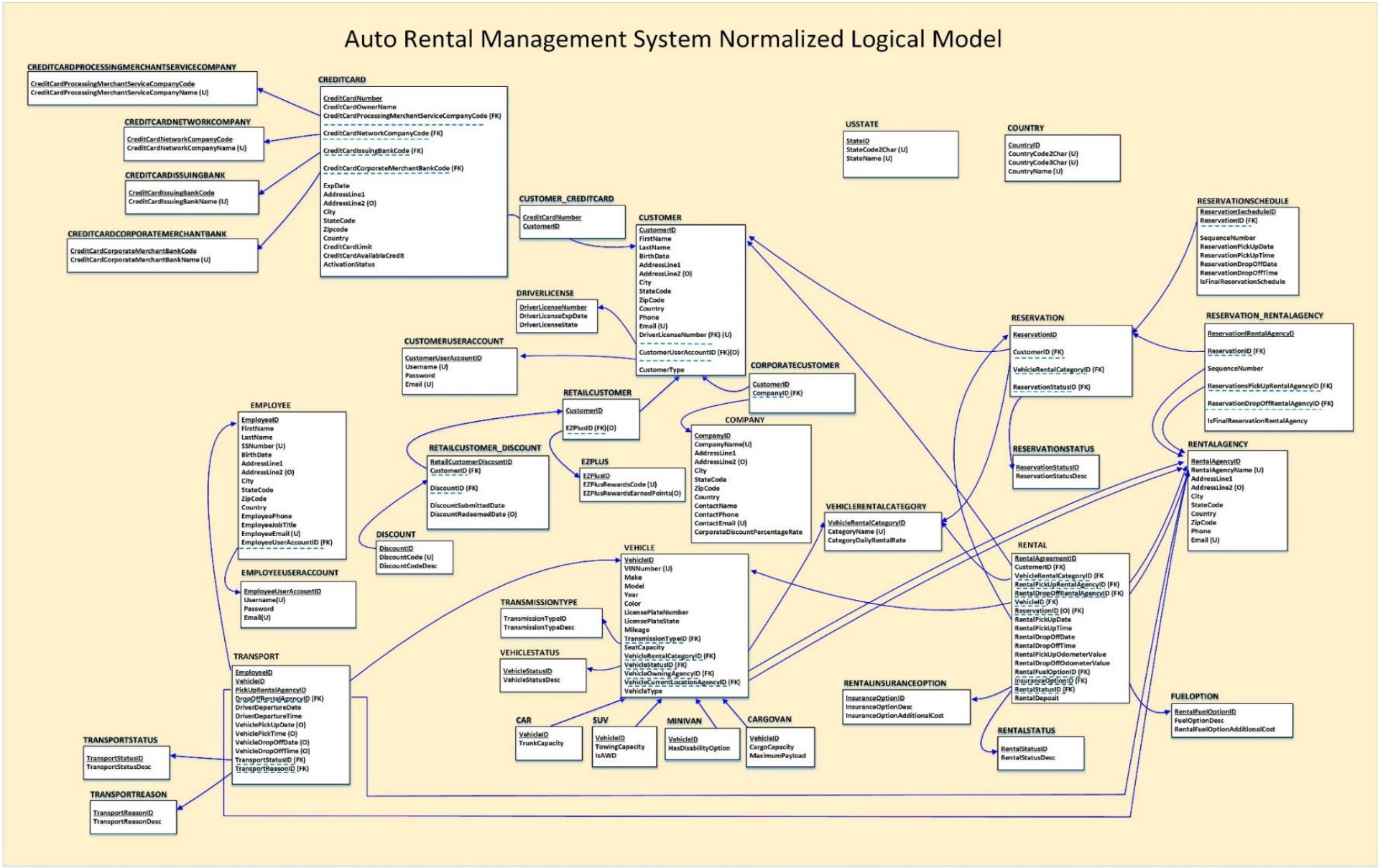
Entity Relational Conceptual Model

The entity relational Conceptual model illustrates the different entities in the database that were identified during the planning phase based on the application requirements. It also provides relationships between these entities.

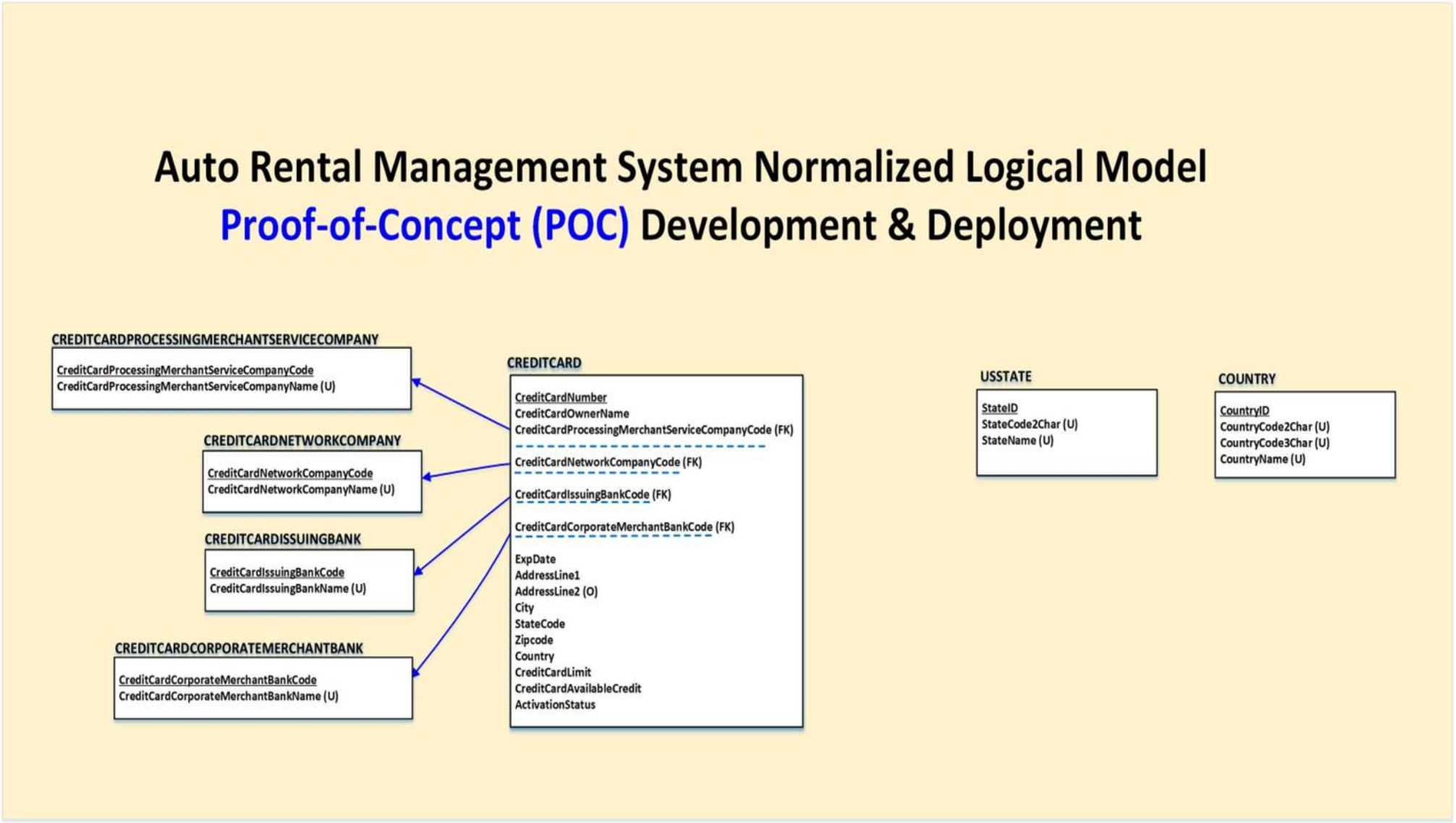


Normalized Logical Model Diagram

A normalized ERR (Entity-Relationship-Relationship) logical diagram is a graphical representation of a database model that has undergone normalization, which systematically reduces data redundancy and improves data integrity by organizing information into related tables and adhering to specific normal forms.



Below is the **PROOF-OF-CONCEPT (POC) Normalized Logical Model** that will be used to create a **PROTOTYPE** of the application to demo to the business. Because it has been optimized and normalized for efficiency and data integrity this diagram only contains the **7 logical tables** in scope of the **POC**:



Physical Model Data Diagram

Below is the physical model detailed using data dictionary tables that serve as reference tools. They provide detailed overviews of the database structure, including entities, columns, data types, constraints, and relationships. The data dictionary ensures data integrity, facilitates efficient data management, and promotes a shared understanding of the database schema.

CREDITCARD

Column Num.	Attribute/Column Name	Generic Data Type Name	MS SQL SERVER Data Type Name	Is it Required?	Length /Size /Format	Constraints	Description/ purpose
1.	CreditCardNumber	String	VARCHAR (16)	Yes	16	PIMARY KEY	***Unique Identifier*** or **Primary Key** for the CreditCard table.
2.	CreditCardOwnerName	String	VARCHAR (100)	Yes	100	NOT NULL	Credit Card Owner first, last and middle initial if included in name
3.	CreditCardProcessingMerchantServiceCompanyCode (FK)	Number	TINYINT	Yes	2	CHECK(CreditCardProcessingMerchantServiceCompanyCode between 1 and 20), NOT NULL	Column stores the Credit Card Processing Merchant Service Company unique code. **Foreign Key** to ***CreditCardProcessingMerchantServiceCompany*** Table.
4.	CreditCardNetworkCompanyCode (FK)	Number	TINYINT	Yes	2	CHECK(CreditCardNetworkCompanyCode between 1 and 20), NOT NULL	Stores the Credit Card Network Company unique code. **Foreign Key** to ***CreditCardNetworkCompany*** Table.

5.	**CreditCardIssuingBankCode (FK)**	**Number**	**TINYINT**	**Yes**	**2**	**CHECK(CreditCardIssuingBankCode between 1 and 20) NOT NULL**	Stores the Credit Card Issuing Bank unique code. **Foreign Key** to ***CreditCardIssuingBank*** Table.
6.	**CreditCardCorporateMerchantBankCode (FK)**	**Number**	**TINYINT**	**Yes**	**2**	**CHECK(CreditCardCorporateMerchantBankCode between 1 and 10) NOT NULL**	Stores the Credit Card Corporate Merchant Bank unique code. **Foreign Key** to ***CreditCardCorporateMerchantBank*** Table.
7.	**ExpDate**	**Date**	**DATE**	**Yes**	**YYYY/MM/DD**	**NOT NULL**	The date the credit card expires. MySQL Date type only supports YYYY/MM/DD format
8.	**AddressLine1**	**String**	**VARCHAR (150)**	**Yes**	**100**	**NOT NULL**	Stores house/building number & street (part 1 of address).
9.	**AddressLine2 (O)**	**String**	**VARCHAR (50)**	**No**	**50**	**NULL**	Optional value that stores remaining part of address such as apartment number, or other address information.
10.	**City**	**String**	**VARCHAR (50)**	**Yes**	**50**	**NOT NULL**	Stores the city name
11.	**StateCode**	**Characters**	**CHAR (2)**	**Yes**	**2**	**NOT NULL**	Stores the U.S. State 2-character code. E.g., NY, NJ, CT, etc.
12.	**Zipcode**	**String**	**VARCHAR (10)**	**Yes**	**10**	**NOT NULL**	Stores US Zip Code/Postal Code. Support format: xxxxx-xxxx.
13.	**Country**	**String**	**VARCHAR (100)**	**Yes**	**100**	**NOT NULL**	Stores the name of the country.

14.	**CreditCardLimit**	**Number**	**DECIMAL (8,2)**	**Yes**	**X = 8** **Y = 2**	**NOT NULL**	The maximum amount of dollars that can be charged to the credit card. Assumes that the credit card has the full limit available. The format is **NUMBER(X,Y)**,where X = The total number of digits and Y = total number of digits to the right of the decimal point. We assume the maximum number that can be stored is 999999.99 for a maximum limit amount $999,999.99, since we don't expect a customer to have a credit limit of $1 Million dollars, we cap it at $999,999.99.
15.	**CreditCardAvailableCredit**	**Number**	**DECIMAL (8,2)**	**Yes**	**X = 8** **Y = 2**	**NOT NULL**	Stores the current remaining credit available for charging.
16.	**ActivationStatus**	**Boolean**	**BIT**	**Yes**	**1**	**NOT NULL**	Stores a Boolean value indicating **True** if credit card is active or **False** otherwise. The **MS SQL SERVER DBMS** has a Data Type named **BIT** that will store **1** to represent **True** and **0** to represent **False**.

CREDITCARDPROCESSINGMERCHANTSERVICECOMPANY

Column Num.	Attribute/Column Name	Generic Data Type Name	MS SQL SERVER Data Type Name	Is it Required?	Length/Size /Format	Constraints	Description/ purpose
1.	CreditCardProcessingMerchantServiceCompanyCode	Number	TINYINT	Yes	2	PRIMARY KEY CHECK(CreditCard Processing MerchantService CompanyCode between 1 and 20)	***Unique Identifier*** or Primary Key for this table.
2.	CreditCardProcessingMerchantServiceCompanyName (U)	String	VARCHAR (30)	Yes	30	UNIQUE NOT NULL	Stores the ***unique*** name of the Credit Card Processing Merchant Service Company.

CREDITCARDNETWORKCOMPANY

Column Num.	Attribute/Column Name	Generic Data Type Name	MS SQL SERVER Data Type Name	Is it Required?	Length/Size /Format	Constraints	Description/ purpose
1.	CreditCardNetworkCompanyCode	Number	TINYINT	Yes	2	PRIMARY KEY CHECK(CreditCardNetworkCompanyCode between 1 and 20)	***Unique Identifier*** or **Primary Key** for this table.
2.	CreditCardNetworkCompanyName (U)	String	VARCHAR (30)	Yes	30	UNIQUE NOT NULL	Stores the ***unique*** name of the Credit Card Network Company.

CREDITCARDISSUINGBANK

Column Num.	Attribute/Column Name	Generic Data Type Name	MS SQL SERVER Data Type Name	Is it Required?	Length/Size /Format	Constraints	Description/ purpose
1.	CreditCardIssuingBankCode	Number	TINYINT	Yes	2	PRIMARY KEY CHECK(CreditCardIssuingBankCode between 1 and 20)	***Unique Identifier*** or **Primary Key** for this table.
2.	CreditCardIssuingBankName (U)	String	VARCHAR (30)	Yes	30	UNIQUE NOT NULL	Stores the ***unique*** name of the Credit Card Processing Merchant Service Company.

CREDITCARDCORPORATEMERCHANTBANK

Column Num.	Attribute/Column Name	Generic Data Type Name	MS SQL SERVER Data Type Name	Is it Required?	Length/ Size /Format	Constraints	Description/ purpose
1.	CreditCardCorporateMerchantBankCode	Number	TINYINT	Yes	2	PRIMARY KEY CHECK(CreditCardCorporateMerchantBankCode between 1 and 10)	***Unique Identifier*** or **Primary Key** for this table.
2.	CreditCardCorporateMerchantBankName (U)	String	VARCHAR (30)	Yes	30	UNIQUE NOT NULL	Stores the ***unique*** name of the Credit Card Corporate Merchant Bank.

USSTATE

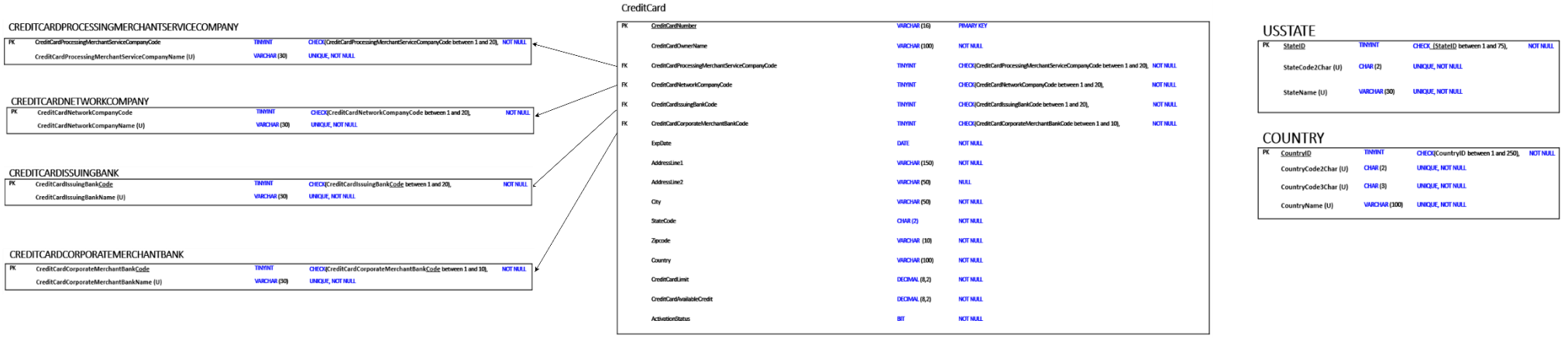
Column Num.	Attribute/Column Name	Generic Data Type Name	MS SQL SERVER Data Type Name	Is it Required?	Length/Size /Format	Constraints	Description/ purpose
1.	**StateID**	Number	**TINYINT**	**Yes**	**2**	**PRIMARY KEY CHECK(StateID between 1 and 75)**	***Unique Identifier*** or **Primary Key** for this table. For the **50** US states **+** Additional territories
2.	**StateCode2Char (U)**	Characters	**CHAR (2)**	Yes	**2**	**UNIQUE NOT NULL**	Stores the U.S. State 2-character code. E.g., NY, NJ, CT etc.
3.	**StateName (U)**	String	**VARCHAR (30)**	Yes	**30**	**UNIQUE NOT NULL**	Stores the unique full name of the U.S. State.

COUNTRY

Column Num.	Attribute/Column Name	Generic Data Type Name	MS SQL SERVER Data Type Name	Is it Required?	Length/Size /Format	Constraints	Description/ purpose
1.	**CountryID**	Number	**TINYINT**	**Yes**	**3**	**PRIMARY KEY CHECK(CountryID between 1 and 250)**	***Unique Identifier*** or **Primary Key** for this table. This **Primary Key HAS Business Meaning**.
2.	**CountryCode2Char (U)**	Characters	**CHAR (2)**	**Yes**	**2**	**NOT NULL UNIQUE**	Stores the country 2-character code. E.g., US, GB, etc.
3.	**CountryCode3Char (U)**	Characters	**CHAR (3)**	**Yes**	**3**	**NOT NULL UNIQUE**	Stores the country 3-character code. E.g., USA, GBR, etc.
4.	**CountryName (U)**	String	**VARCHAR (100)**	**Yes**	**100**	**NOT NULL UNIQUE**	Stores the unique full name of the country.

Physical Model Schema Design Diagram

Below is the physical schema for these tables, which outlines the database's structure and organization. This schema provides a comprehensive description of the tables, including their relationships, primary and foreign keys. It serves as a blueprint for implementing the database design in a real-world environment, ensuring efficient data storage and retrieval.



Development & Implementation

To implement the database a script fille was created to create and define the tables and relationships in the proof of concept. Below is the script used for these 7 tables in Microsoft SQL Server Management Studio. In order for a successful implementation the physical model schema was followed exactly.

```
CREATE TABLE CREDITCARDPROCESSINGMERCHANTSERVICECOMPANY (
    CreditCardProcessingMerchantServiceCompanyCode TINYINT CHECK(
        CreditCardProcessingMerchantServiceCompanyCode between 1 and 20
    ) PRIMARY KEY,
    CreditCardProcessingMerchantServiceCompanyName VarChar(30) UNIQUE NOT NULL
);

CREATE TABLE CREDITCARDNETWORKCOMPANY (
    CreditCardNetworkCompanyCode TINYINT CHECK(
        CreditCardNetworkCompanyCode between 1 and 20) PRIMARY KEY,
    CreditCardNetworkCompanyName VarChar(30) UNIQUE NOT NULL
);

CREATE TABLE CREDITCARDISSUINGBANK (
    CreditCardIssuingBankCode TINYINT CHECK(
        CreditCardIssuingBankCode between 1 and 20) PRIMARY KEY,
    CreditCardIssuingBankName VarChar(30) UNIQUE NOT NULL
);

CREATE TABLE CREDITCARDCORPORATEMERCHANTBANK (
    CreditCardCorporateMerchantBankCode TINYINT CHECK(
        CreditCardCorporateMerchantBankCode between 1 and 10) PRIMARY KEY,
    CreditCardCorparateMerchantBankName VarChar(30) UNIQUE NOT NULL
);

Create TABLE USSTATE (
    StateID TINYINT CHECK(
        StateID between 1 and 75) PRIMARY KEY,
    StateCode2Char CHAR(2) UNIQUE NOT NULL,
    StateName VARCHAR(30) UNIQUE NOT NULL,
);
```

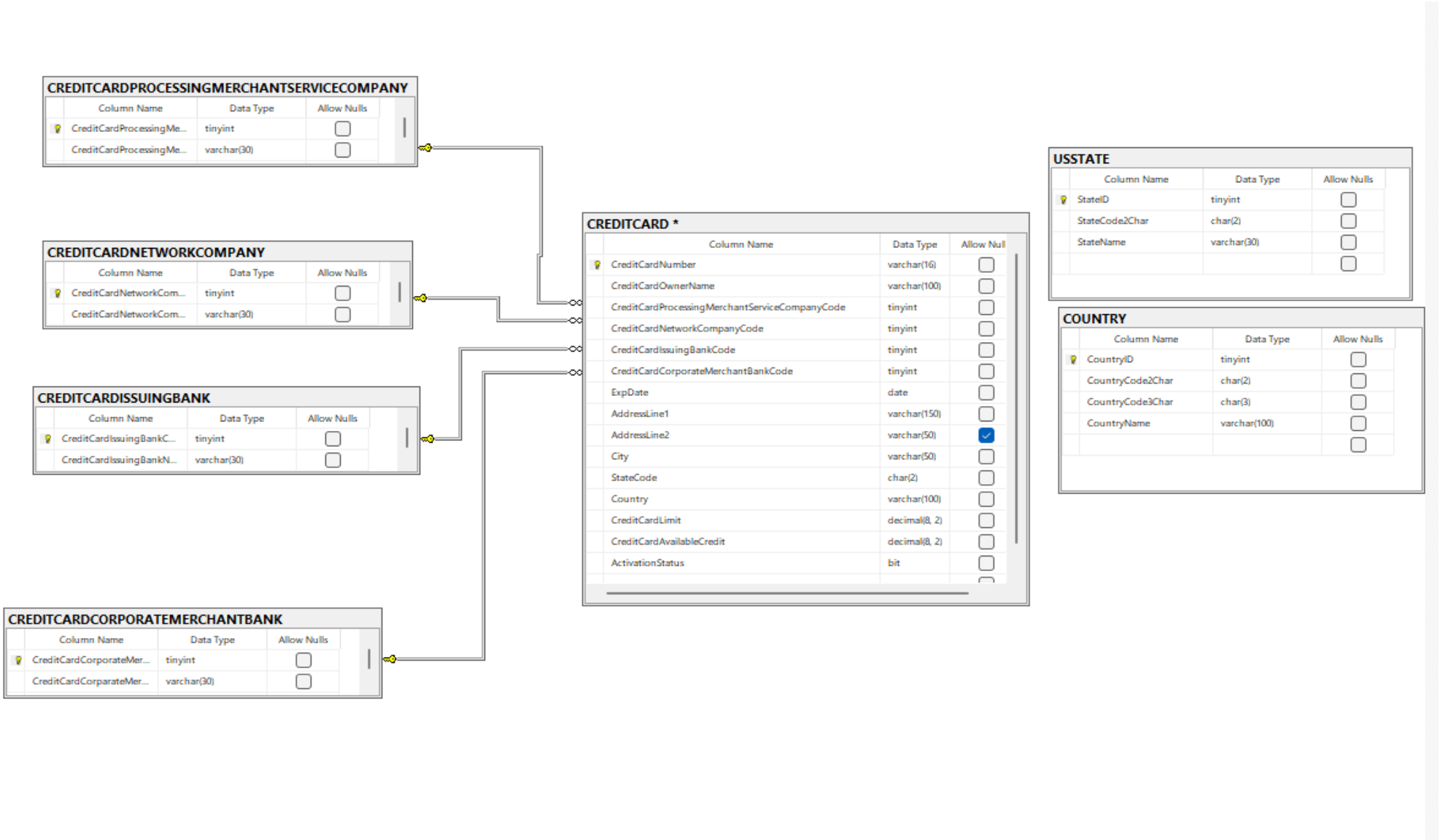
```
Create TABLE COUNTRY(
    CountryID TINYINT CHECK(
        CountryID between 1 and 250) PRIMARY KEY,
    CountryCode2Char CHAR(2) UNIQUE NOT NULL,
    CountryCode3Char CHAR(3) UNIQUE NOT NULL,
    CountryName VARCHAR(100) UNIQUE NOT NULL,
);


CREATE TABLE CREDITCARD (
    CreditCardNumber VARCHAR(16) PRIMARY KEY,
    CreditCardOwnerName VARCHAR(100) NOT NULL,
    CreditCardProcessingMerchantServiceCompanyCode TINYINT CHECK(
        CreditCardProcessingMerchantServiceCompanyCode between 1 and 20) NOT NULL,
    CreditCardNetworkCompanyCode TINYINT CHECK(
        CreditCardNetworkCompanyCode between 1 and 20
    )NOT NULL,
    CreditCardIssuingBankCode TINYINT  CHECK(
        CreditCardIssuingBankCode between 1 and 20) NOT NULL,
    CreditCardCorporateMerchantBankCode TINYINT CHECK(
        CreditCardCorporateMerchantBankCode between 1 and 10) NOT NULL,
    ExpDate DATE NOT NULL,
    AddressLine1 VARCHAR(150) NOT NULL,
    AddressLine2 VARCHAR(50)  NULL,
    City VARCHAR(50) NOT NULL,
    StateCode CHAR(2) NOT NULL,
    Country VARCHAR(100) NOT NULL,
    CreditCardLimit DECIMAL(8, 2) NOT NULL,
    CreditCardAvailableCredit DECIMAL(8, 2) NOT NULL,
    ActivationStatus BIT NOT NULL,
    CONSTRAINT fk_CreditCardInstance_CreditCardProcessingMerchantService FOREIGN KEY (CreditCardProcessingMerchantServiceCompanyCode)
REFERENCES CREDITCARDPROCESSINGMERCHANTSERVICECOMPANY (CreditCardProcessingMerchantServiceCompanyCode) ON DELETE CASCADE ON UPDATE
CASCADE,
    CONSTRAINT fk_CreditCardInstance_CreditCardNetworkCompany FOREIGN KEY (CreditCardNetworkCompanyCode) REFERENCES
CREDITCARDNETWORKCOMPANY (CreditCardNetworkCompanyCode) ON DELETE CASCADE ON UPDATE CASCADE,
    CONSTRAINT fk_CreditCardInstance_CreditCardIssuingBank FOREIGN KEY (CreditCardIssuingBankCode) REFERENCES CREDITCARDISSUINGBANK
(CreditCardIssuingBankCode) ON DELETE CASCADE ON UPDATE CASCADE,
    CONSTRAINT fk_CreditCardInstance_CreditCardCorporateMerchantBank FOREIGN KEY (CreditCardCorporateMerchantBankCode) REFERENCES
CREDITCARDCORPORATEMERCHANTBANK(CreditCardCorporateMerchantBankCode) ON DELETE CASCADE ON UPDATE CASCADE
);
```

Implemented Physical Schema Diagram

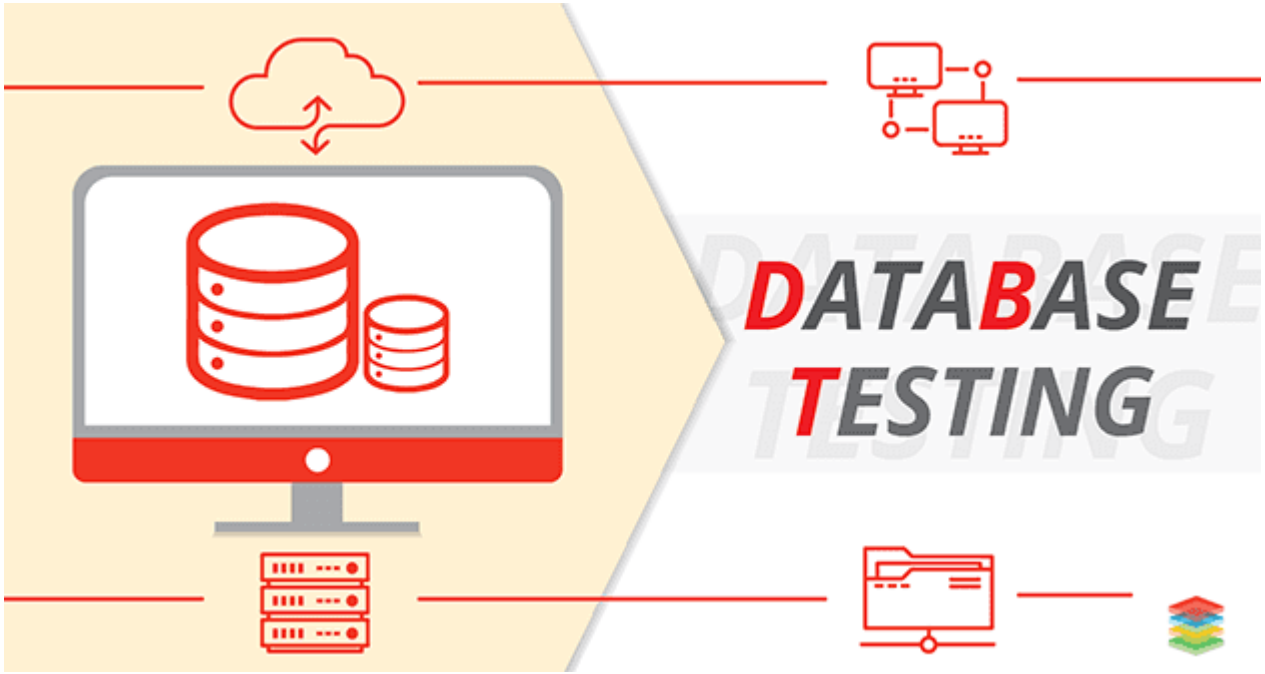
It is the goal to have an implemented physical schema design that matches our physical schema model. Being able to generate the exact schema within the development environment as the model shows that the schema was implemented successfully. Below is the implemented schema diagram.

Implemented Schema Diagram



Database Validation Testing

In this crucial phase, the team conducts thorough testing of the meticulously designed table groups within the database. A battery of SQL statements and methods are deployed to scrutinize specific business-related conditions, guaranteeing that the database performs with the precision intended during the design phase.



CREDITCARDPROCESSINGMERCHANTSERVICECOMPANY Table

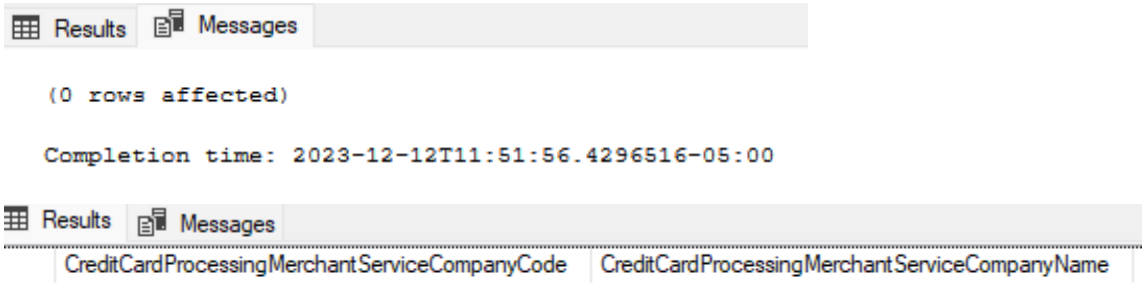
This Section shows the content of the CREDITCARDPROCESSINGMERCHANTSERVICECOMPANY Table before and after populating it using SQL INSERT STATEMENTS

CREDITCARDPROCESSINGMERCHANTSERVICECOMPANY Before execution of SQL INSERET STATEMENTS:

The following SQL SELECT STATEMENT was used to list the context of the CREDITCARDPROCESSINGMERCHANTSERVICECOMPANY Table:

```
SELECT * FROM CREDITCARDPROCESSINGMERCHANTSERVICECOMPANY;
```

The Current state of CREDITCARDPROCESSINGMERCHANTSERVICECOMPANY Table is:



List of SQL INSERT STATEMENTS:

The following SQL INSERT STATEMENTS were used to insert records into the CREDITCARDPROCESSINGMERCHANTSERVICECOMPANY Table:

```
Insert into CREDITCARDPROCESSINGMERCHANTSERVICECOMPANY
(CreditCardProcessingMerchantServiceCompanyCode,CreditCardProcessingMerchantServiceCompanyName)
values
(1,'Stax by Fattmerchant'),
(2,'Helcim'),
(3,'Dharma Merchant Services'),
(4,'Payment Depot'),
(5,'National Processing'),
(6,'Block'),
(7,'Intuit Quickbooks'),
(8,'PayPal'),
(9,'Stripe'),
(10,'Flagship Merchant Services'),
(11,'Clover');
```

CREDITCARDPROCESSINGMERCHANTSERVICECOMPANY

After Execution SQL INSERT STATEMENTS:

The following SQL SELECT STATEMENTS was used to list the content of the CREDITCARDPROCESSINGMERCHANTSERVICECOMPANY Table:

```
SELECT * FROM CREDITCARDPROCESSINGMERCHANTSERVICECOMPANY
ORDER BY CreditCardProcessingMerchantServiceCompanyCode;
```

The Final state of CREDITCARDPROCESSINGMERCHANTSERVICECOMPANY Table is:

Results

CreditCardProcessingMerchantServiceCompanyCode	CreditCardProcessingMerchantServiceCompanyName
1	Stax by Fattmerchant
2	Helcim
3	Dharma Merchant Services
4	Payment Depot
5	National Processing
6	Block
7	Intuit Quickbooks
8	PayPal
9	Stripe
10	Flagship Merchant Services
11	Clover

(11 rows affected)

Completion time: 2023-12-12T16:59:34.6144978-05:00

Populating the CREDITCARDNETWORKCOMPANY Table

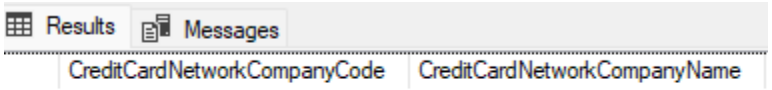
This Section shows the content of the CREDITCARDNETWORKCOMPANY Table before and after populating it using SQL INSERT STATEMENTS

CREDITCARDNETWORKCOMPANY Before execution of SQL INSERET STATEMENTS:

The following SQL SELECT STATEMENT was used to list the context of the CREDITCARDNETWORKCOMPANY Table:

```
Select * From  CREDITCARDNETWORKCOMPANY;
```

The Current state of CREDITCARDNETWORKCOMPANY Table is:




```
(0 rows affected)

Completion time: 2023-12-12T11:50:12.8748447-05:00
```

List of SQL INSERT STATEMENTS:

The following SQL INSERT STATEMENTS were used to insert records into the CREDITCARDNETWORKCOMPANY Table:

```
INSERT INTO
CREDITCARDNETWORKCOMPANY
(CreditCardNetworkCompanyCode,CreditCardNetworkCompanyName)
VALUES
(1,'American Express'),
(2,'Visa'),
(3,'Master Card'),
(4,'Discover'),
(5,'DinersClub'),
(6,'Interlink'),
(7,'Star'),
(8,'Accel'),
(9,'Interac'),
(10,'Visa Ready Link'),
(11,'Pulse'),
(12,'JCB (Japan Credit Bureau)'),
(13,'Rupay');
```

CREDITCARDNETWORKCOMPANY After Execution SQL INSERT STATEMENTS:

The following SQL SELECT STATEMENTS was used to list the content of the CREDITCARDNETWORKCOMPANY Table:

```
SELECT * FROM CREDITCARDNETWORKCOMPANY
ORDER BY CreditCardNetworkCompanyCode;
```

Results

CreditCardNetworkCompanyCode	CreditCardNetworkCompanyName
1	American Express
2	Visa
3	Master Card
4	Discover
5	Diners Club
6	Interlink
7	Star
8	Accel
9	Interac
10	Visa Ready Link
11	Pulse
12	JCB (Japan Credit Bureau)
13	Rupay

(13 rows affected)

Completion time: 2023-12-12T17:09:00.8902789-05:00

Populating the CREDITCARDISSUINGBANK Table

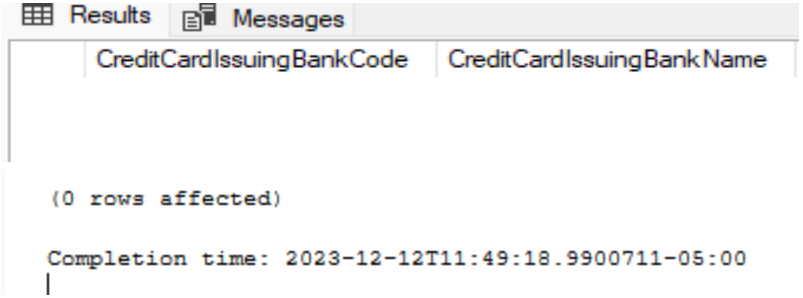
This Section shows the content of the CREDITCARDISSUINGBANK Table before and after populating it using SQL INSERT STATEMENTS

CREDITCARDISSUINGBANK Before execution of SQL INSERET STATEMENTS:

The following SQL SELECT STATEMENT was used to list the context of the CREDITCARDISSUINGBANK Table:

```
SELECT * FROM CREDITCARDISSUINGBANK;
```

The Current state of the CREDITCARDISSUINGBANK Table is:



List of SQL INSERT STATEMENTS:

The following SQL INSERT STATEMENTS were used to insert records into the CREDITCARDISSUINGBANK Table:

```
INSERT INTO
CREDITCARDISSUINGBANK
(CreditCardIssuingBankCode,CreditCardIssuingBankName)
VALUES
(1, 'American Express'),
(2, 'Bank of America'),
(3, 'Barclays'),
(4, 'Capital One'),
(5, 'Chase'),
(6, 'Citi'),
(7, 'Discover'),
(8, 'Synchrony Bank'),
(9, 'U.S. Bank'),
(10, 'Wells Fargo');
```

CREDITCARDISSUINGBANK After Execution SQL INSERT STATEMENTS:

The following SQL SELECT STATEMENTS was used to list the content of the CREDITCARDISSUINGBANK **Table:**

```
Select * From CREDITCARDISSUINGBANK;
```

The Final State of the CREDITCARDISSUINGBANK Table is:

Results

```
CreditCardIssuingBankCode CreditCardIssuingBankName
------------------------- ------------------------------
1                         American Express
2                         Bank of America
3                         Barclays
4                         Capital One
5                         Chase
6                         Citi
7                         Discover
8                         Synchrony Bank
9                         U.S. Bank
10                        Wells Fargo

(10 rows affected)


Completion time: 2023-12-12T17:46:45.8391575-05:00
```

Populating the CREDITCARDCORPORATEMERCHANTBANK Table

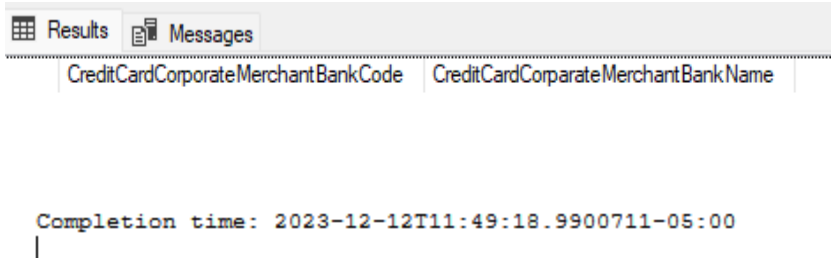
This Section shows the content of the CREDITCARDCORPORATEMERCHANTBANK Table before and after populating it using SQL INSERT STATEMENTS

CREDITCARDCORPORATEMERCHANTBANK Before execution of SQL INSERET STATEMENTS:

The following SQL SELECT STATEMENT was used to list the context of the CREDITCARDCORPORATEMERCHANTBANK Table:

```
SELECT * FROM CREDITCARDCORPORATEMERCHANTBANK;
```

The Current state of the CREDITCARDCORPORATEMERCHANTBANK Table is:



List of SQL INSERT STATEMENTS:

The following SQL INSERT STATEMENTS were used to insert records into the CREDITCARDCORPORATEMERCHANTBANK Table:

```
INSERT INTO
CREDITCARDCORPORATEMERCHANTBANK
(CreditCardCorporateMerchantBankCode,CreditCardCorparateMerchantBankName)
VALUES
(1,'Chase'),
(2, 'Citi'),
(3,'Capital One');
```

CREDITCARDCORPORATEMERCHANTBANK After Execution SQL INSERT STATEMENTS:

The following SQL SELECT STATEMENTS was used to list the content of the **CREDITCARDCORPORATEMERCHANTBANK** Table:

```
SELECT * FROM CREDITCARDCORPORATEMERCHANTBANK;
```

The Final state of CREDITCARDCORPORATEMERCHANTBANK Table is:

Results

```
CreditCardCorporateMerchantBankCode CreditCardCorparateMerchantBankName
----------------------------------- -----------------------------------
3                                   Capital One
1                                   Chase
2                                   Citi

(3 rows affected)


Completion time: 2023-12-12T17:41:57.2593291-05:00
```

Populating the CREDITCARD Table

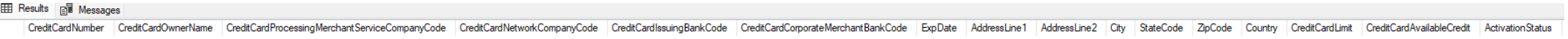
This Section shows the content of the CREDITCARD Table before and after populating it using SQL INSERT STATEMENTS

CREDITCARD Before execution of SQL INSERET STATEMENTS:

The following SQL SELECT STATEMENT was used to list the context of the CREDITCARD Table:

```
SELECT * FROM CREDITCARD;
```

The Current state of the CREDITCARD Table is:



List of SQL INSERT STATEMENTS:

The following SQL INSERT STATEMENTS were used to insert records into the CREDITCARD Table:

```
INSERT INTO CREDITCARD
(CreditCardNumber,CreditCardOwnerName,
CreditCardProcessingMerchantServiceCompanyCode,CreditCardNetworkCompanyCode,
CreditCardIssuingBankCode,CreditCardCorporateMerchantBankCode,ExpDate,
AddressLine1,AddressLine2,City,StateCode,ZipCode,Country,
CreditCardLimit,CreditCardAvailableCredit,ActivationStatus
)
VALUES
(1111111111111111,'Ryan Brown',8,2,2,1,'01/01/2025',
'111 Jay Street','Suite 101', 'Freehold','NJ','17711','USA',
3000.00,1000.00,1),
(2222222222222221,'ALex Rodriguez',7,6,4,2,'02/02/2027',
'222 Glenwood Rd', 'Apt 6H','Brooklyn','NY', '11222','USA',
10000.00,8000.00,1),
(2222222222222222,'ALex Rodriguez',8,2,1,2,'02/02/2027',
'222 Glenwood Rd', 'Apt 6H','Brooklyn','NY', '11222','USA',
10000.00,8000.00,1),
(2222222222222223,'ALex Rodriguez',11,4,2,3,'02/02/2027',
'222 Glenwood Rd', 'Apt 6H','Brooklyn','NY', '11222','USA',
10000.00,8000.00,1),
(3333333333333333,'Michelle Apicotta',1,3,5,1,'03/03/2024',
'333 5th Avenue', NULL, 'New York','NY', '10033', 'USA',
3000.00,3000.00,0),
(4444444444444444,'Mike Greene', 6, 11,8,3,'04/04/2029',
'444 Flatlands Ave','3rd Floor', 'Allentown','PA','14344','USA',
5000.00,3000.00,1),
(5555555555555555,'Sandra Lopez', 8, 7, 1, 1,'05/05/2030',
'6th Avenue', ' Apt 4f', 'Jersey City','NJ','07032','USA',
10000.00,500.00,1);
```

CREDITCARD After Execution SQL INSERT STATEMENTS:

The following SQL SELECT STATEMENT was used to list the content of the CREDITCARD Table:

```
SELECT * FROM CREDITCARD;
```

The Final state of the CREDITCARD Table is:

Results Messages

	CreditCardNumber	CreditCardOwnerName	CreditCardProcessingMerchantServiceCompanyCode	CreditCardNetworkCompanyCode	CreditCardIssuingBankCode	CreditCardCorporateMerchantBankCode	ExpDate	AddressLine1	AddressLine2	City	StateCode	ZipCode	Country	CreditCardLimit	CreditCardAvailableCredit	ActivationStatus
1	1111111111111111	Ryan Brown	8	2	2	1	2025-01-01	111 Jay Street	Suite 101	Freehold	NJ	17711	USA	3000.00	1000.00	1
2	2222222222222221	ALex Rodriguez	7	6	4	2	2027-02-02	222 Glenwood Rd	Apt 6H	Brooklyn	NY	11222	USA	10000.00	8000.00	1
3	2222222222222222	ALex Rodriguez	8	2	1	2	2027-02-02	222 Glenwood Rd	Apt 6H	Brooklyn	NY	11222	USA	10000.00	8000.00	1
4	2222222222222223	ALex Rodriguez	11	4	2	3	2027-02-02	222 Glenwood Rd	Apt 6H	Brooklyn	NY	11222	USA	10000.00	8000.00	1
5	3333333333333333	Michelle Apicotta	1	3	5	1	2024-03-03	333 5th Avenue	NULL	New York	NY	10033	USA	3000.00	3000.00	0
6	4444444444444444	Mike Greene	6	11	8	3	2029-04-04	444 Flatlands Ave	3rd Floor	Allentown	PA	14344	USA	5000.00	3000.00	1
7	5555555555555555	Sandra Lopez	8	7	1	1	2030-05-05	6th Avenue	Apt 4f	Jersey City	NJ	07032	USA	10000.00	500.00	1

Populating the USSTATE Table

This Section shows the content of the USSTATE Table before and after populating it using SQL INSERT STATEMENTS

USSTATE Before execution of SQL INSERET STATEMENTS:

The following SQL SELECT STATEMENT was used to list the context of the USSTATE Table:

```
Select * From USSTATE;
```

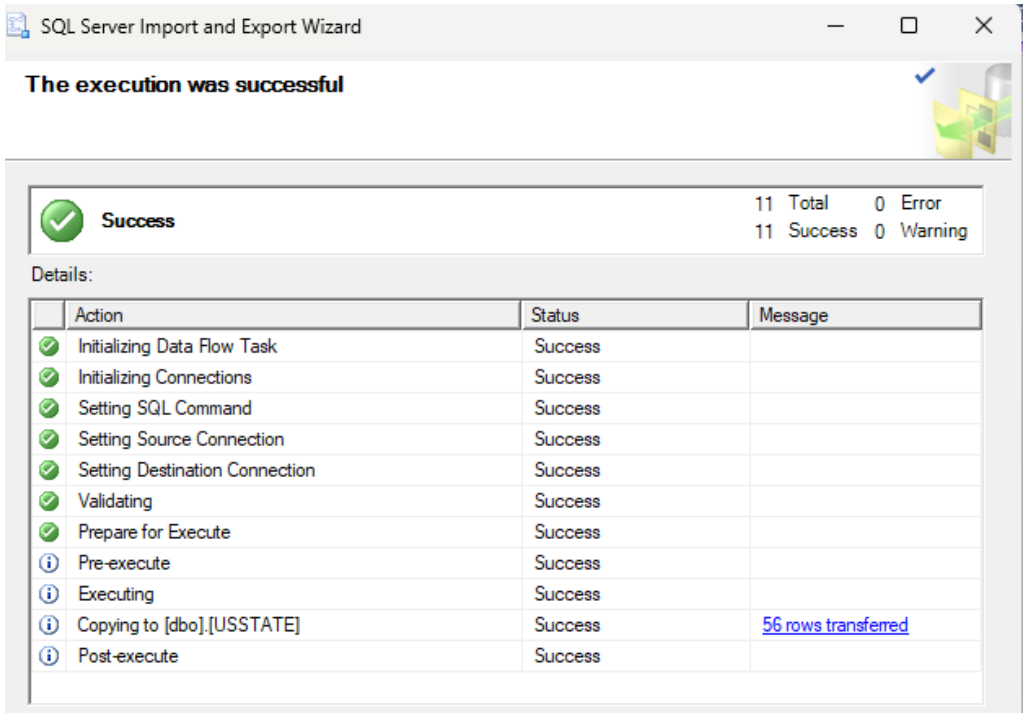
The Current state of USSTATE Table is:


```
(0 rows affected)

Completion time: 2023-12-12T11:42:51.4994249-05:00
```

SQL IMPORT WIZARD:

The SQL SERVER IMPORT EXPORT WIZARD TOOL was leveraged to populate the USSTATE Table:



USSTATE Table After SQL IMPORT WIZARD:

The following SQL SELECT STATEMENT was used to list the content of the **USSTATE** Table:

```
Select * From USSTATE;
```

The Current state of USSTATE Table is:

Results

38	ND	NORTH DAKOTA
39	OH	OHIO
40	OK	OKLAHOMA
41	OR	OREGON
42	PA	PENNSYLVANIA
43	PR	PUERTO RICO
44	RI	RHODE ISLAND
45	SC	SOUTH CAROLINA
46	SD	SOUTH DAKOTA
47	TN	TENNESSEE
48	TX	TEXAS
49	UT	UTAH
50	VT	VERMONT
51	VI	VIRGIN ISLANDS
52	VA	VIRGINIA
53	WA	WASHINGTON
54	WV	WEST VIRGINIA
55	WI	WISCONSIN
56	WY	WYOMING

(56 rows affected)

Completion time: 2023-12-12T17:30:04.9074150-05:00

Populating the COUNTRY Table

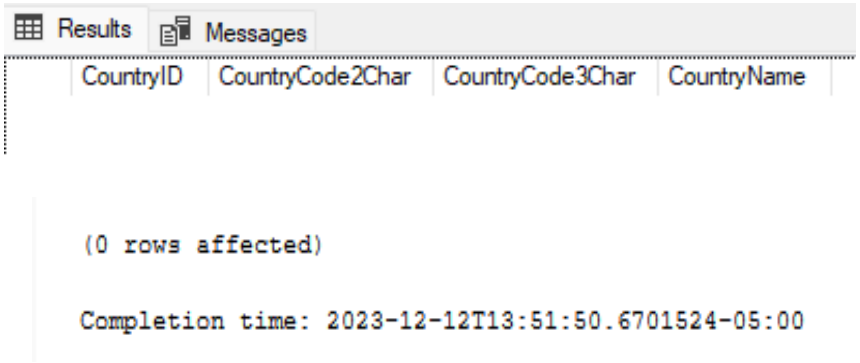
This Section shows the content of the COUNTRY Table before and after populating it using SQL INSERT STATEMENTS

COUNTRY Before execution of SQL INSERET STATEMENTS:

The following SQL SELECT STATEMENT was used to list the context of the COUNTRY Table:

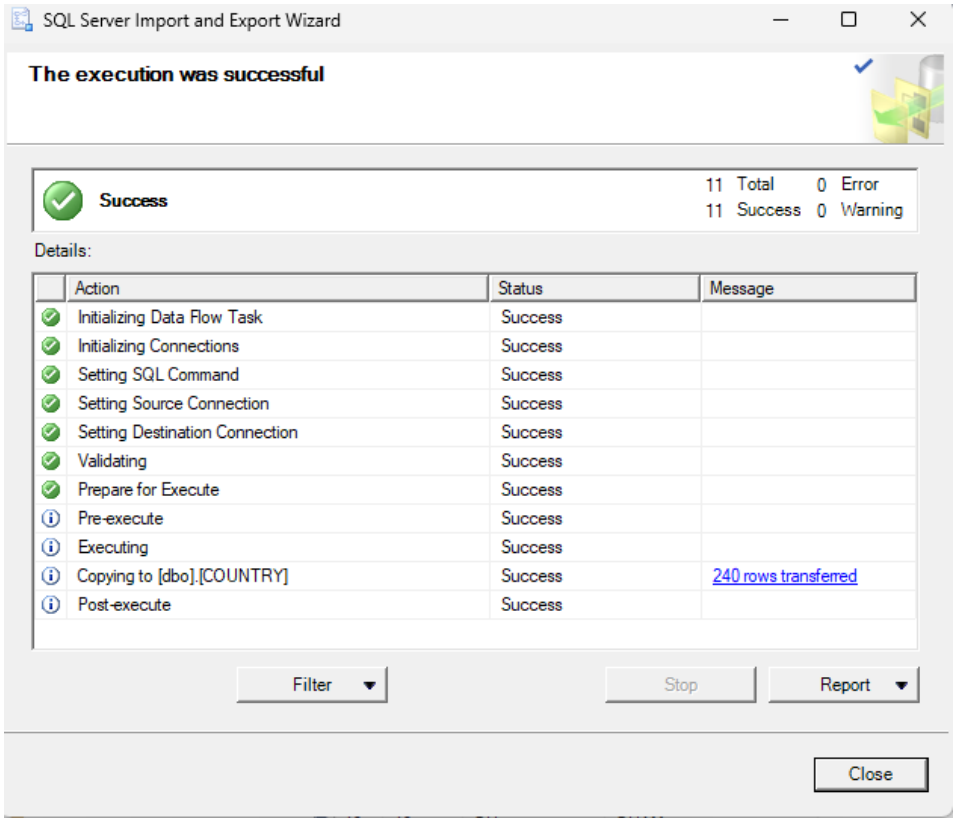
```
Select * From COUNTRY;
```

The Current state of the COUNTRY Table is:



SQL IMPORT WIZARD:

The SQL SERVER IMPORT EXPORT WIZARD TOOL was leveraged to populate the COUNTRY Table from a preconfigured CSV file:



COUNTRY Table After Execution SQL IMPORT WIZARD:

The following SQL SELECT STATEMENTS was used to list the content of the COUNTRY Table:

```
Select * From COUNTRY;
```

The Current state of COUNTRY Table is:

Results

222	UG	UGA	UGANDA
223	UA	UKR	UKRAINE
224	AE	ARE	UNITED ARAB EMIRATES
225	GB	GBR	UNITED KINGDOM
226	US	USA	UNITED STATES
227	UM	UMI	UNITED STATES MINOR OUTLYING ISLANDS
228	UY	URY	URUGUAY
229	UZ	UZB	UZBEKISTAN
230	VU	VUT	VANUATU
231	VA	VAT	VATICAN CITY STATE (HOLY SEE)
232	VE	VEN	VENEZUELA
233	VN	VNM	VIET NAM
234	VG	VGB	VIRGIN ISLANDS (BRITISH)
235	VI	VIR	VIRGIN ISLANDS (U.S.)
236	WF	WLF	WALLIS AND FUTUNA ISLANDS
237	EH	ESH	WESTERN SAHARA
238	YE	YEM	YEMEN
239	ZM	ZMB	ZAMBIA
240	ZW	ZWE	ZIMBABWE

(240 rows affected)

Completion time: 2023-12-12T17:32:12.7724584-05:00

Searching for records in the Database

This section provides testing and validation the querying of the **CREDITCARD Table** by different SQL Select Statements.

CreditCard Table Current State

The following SQL Select Statement was used to query the **CREDITCARD Table** to query its current contents:

```
SELECT * FROM CREDITCARD;
```

CreditCardNumber	CreditCardOwnerName	CreditCardProcessingMerchantServiceCompanyCode	CreditCardNetworkCompanyCode	CreditCardIssuingBankCode	CreditCardCorporateMerchantBankCode	ExpDate	AddressLine1	AddressLine2	City	StateCode	ZipCode	Country	CreditCardLimit	CreditCardAvailableCredit	ActivationStatus
1111111111111111	Ryan Brown	8	2	2	1	2025-01-01	111 Jay Street	Suite 101	Freehold	NJ	17711	USA	3000.00	1000.00	1
2222222222222221	ALex Rodriguez	7	6	4	2	2027-02-02	222 Glenwood Rd	Apt 6H	Brooklyn	NY	11222	USA	10000.00	8000.00	1
2222222222222222	ALex Rodriguez	8	2	1	2	2027-02-02	222 Glenwood Rd	Apt 6H	Brooklyn	NY	11222	USA	10000.00	8000.00	1
2222222222222223	ALex Rodriguez	11	4	2	3	2027-02-02	222 Glenwood Rd	Apt 6H	Brooklyn	NY	11222	USA	10000.00	8000.00	1
3333333333333333	Michelle Apicotta	1	3	5	1	2024-03-03	333 5th Avenue	NULL	New York	NY	10033	USA	3000.00	3000.00	0
4444444444444444	Mike Greene	6	11	8	3	2029-04-04	444 Flatlands Ave	3rd Floor	Allentown	PA	14344	USA	5000.00	3000.00	1
5555555555555555	Sandra Lopez	8	7	1	1	2030-05-05	6th Avenue	Apt 4f	Jersey City	NJ	07032	USA	10000.00	500.00	1

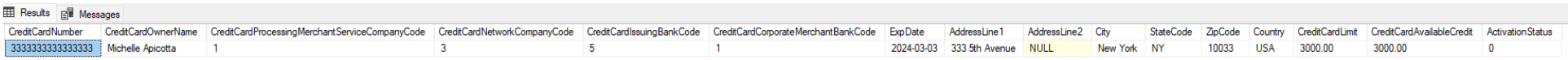
Selecting one record based on Credit Card Number

The following SQL Select Statement was used to query the **CREDITCARD Table** to query one record with all columns filtered by the tables primary key **CreditCardNumber**:

```
SELECT * FROM CREDITCARD
WHERE CreditCardNumber = '3333333333333333';
```

Result of the Query:

Below is the direct result for Select statement above proving that the **CREDITCARD Table** can be queried to pull records via **CreditCardNumber**.



Results | Messages

CreditCardNumber	CreditCardOwnerName	CreditCardProcessingMerchantServiceCompanyCode	CreditCardNetworkCompanyCode	CreditCardIssuingBankCode	CreditCardCorporateMerchantBankCode	ExpDate	AddressLine1	AddressLine2	City	StateCode	ZipCode	Country	CreditCardLimit	CreditCardAvailableCredit	ActivationStatus
3333333333333333	Michelle Apicotta	1	3	5	1	2024-03-03	333 5th Avenue	NULL	New York	NY	10033	USA	3000.00	3000.00	0

Searching for multiple records based on Credit Limit

The following SQL Select Statement was used to query the **CREDITCARD Table** to all the records with customers from **New Jersey**

And return the following columns**: CreditCardNumber, CreditCardOwnerName**.
ExpDate,Addressline1,AdressLine2,City,StateCode,ZipCode,Country,ActivationStatus:

```
SELECT CreditCardNumber, CreditCardOwnerName,
ExpDate,AddressLine1,AddressLine2,
City,StateCode,ZipCode,Country,ActivationStatus
FROM CREDITCARD
WHERE StateCode = 'NJ';
```

Result of the Query:

Below is the direct result for Select statement above proving that the CREDITCARD Table can be queried to pull records and filtered with a nonprimary key like **StateCode**.

Results Messages

CreditCardNumber	CreditCardOwnerName	ExpDate	AddressLine1	AddressLine2	City	StateCode	ZipCode	Country	ActivationStatus
1111111111111111	Ryan Brown	2025-01-01	111 Jay Street	Suite 101	Freehold	NJ	17711	USA	1
5555555555555555	Sandra Lopez	2030-05-05	6th Avenue	Apt 4f	Jersey City	NJ	07032	USA	1

Searching Multiple Tables

Often times it is necessary to access information from various tables to test and validate this we preformed a query from 5 different tables to provide information from each for any customers who live in the city of Brooklyn.
Below is the SQL Statement used to preform this query:

```
SELECT          CREDITCARD.CreditCardNumber, CREDITCARD.CreditCardOwnerName,
          CREDITCARD.CreditCardProcessingMerchantServiceCompanyCode,
          CREDITCARDPROCESSINGMERCHANTSERVICECOMPANY.CreditCardProcessingMerchantServiceCompanyName,
          CREDITCARD.CreditCardNetworkCompanyCode,CREDITCARDNETWORKCOMPANY.CreditCardNetworkCompanyName,
          CREDITCARD.CreditCardIssuingBankCode, CREDITCARDISSUINGBANK.CreditCardIssuingBankName,
          CREDITCARD.CreditCardCorporateMerchantBankCode,CREDITCARDCORPORATEMERCHANTBANK.CreditCardCorparateMerchantBankName,
          CREDITCARD.ExpDate, CREDITCARD.AddressLine1, CREDITCARD.AddressLine2, CREDITCARD.City, CREDITCARD.StateCode,
          CREDITCARD.ZipCode, CREDITCARD.Country, CREDITCARD.CreditCardLimit, CREDITCARD.CreditCardAvailableCredit,
          CREDITCARD.ActivationStatus


FROM   CREDITCARD
INNER JOIN
     CREDITCARDPROCESSINGMERCHANTSERVICECOMPANY ON
          CREDITCARD.CreditCardProcessingMerchantServiceCompanyCode =
          CREDITCARDPROCESSINGMERCHANTSERVICECOMPANY.CreditCardProcessingMerchantServiceCompanyCode
INNER JOIN
     CREDITCARDNETWORKCOMPANY ON
          CREDITCARD.CreditCardNetworkCompanyCode = CREDITCARDNETWORKCOMPANY.CreditCardNetworkCompanyCode
INNER JOIN
     CREDITCARDISSUINGBANK ON
          CREDITCARD.CreditCardIssuingBankCode = CREDITCARDISSUINGBANK.CreditCardIssuingBankCode
INNER JOIN
          CREDITCARDCORPORATEMERCHANTBANK ON
          CREDITCARD.CreditCardCorporateMerchantBankCode = CREDITCARDCORPORATEMERCHANTBANK.CreditCardCorporateMerchantBankCode

WHERE          CITY='Brooklyn';
```

Result of the Query

Below is the result of the selection query above proving selection across multiple tables filtered by the City column.

CreditCard Number	CreditCard Owner Name	CreditCard Processing Merchant Service Company Code	CreditCard Processing Merchant Service Company Name	CreditCard Network Company Code	CreditCard Network Company Name	CreditCard Issuing Bank Code	CreditCard Issuing Bank Name	CreditCard Corporate Merchant Bank Code	CreditCard Corparate Merchant Bank Name	ExpDate	Address Line1	Address Line2	City	StateCode	ZipCode	Country	CreditCard Limit	CreditCard Available Credit	Activation Status
2222222222222221	Alex Rodriguez	7	Intuit Quickbooks	6	Interlink	4	Capital One	2	Citi	2/2/2027	222 Glenwood Rd	Apt 6H	Brooklyn	NY	11222	USA	10000	8000	1
2222222222222222	Alex Rodriguez	8	PayPal	2	Visa	1	American Exp	2	Citi	2/2/2027	222 Glenwood Rd	Apt 6H	Brooklyn	NY	11222	USA	10000	8000	1
2222222222222223	Alex Rodriguez	11	Clover	4	Discover	2	Bank of Amer	3	Capital One	2/2/2027	222 Glenwood Rd	Apt 6H	Brooklyn	NY	11222	USA	10000	8000	1

Updating Records

It is important to that the database provide functionality to make updates or charges to records and related tables. Here developers will test and validate this ability of the database to respond to changes made from the Application.

- Application features that **UPDATE** A CREDIT CARD RECORD OF A CUSTOMER MODIFYING EVERY SINGLE COLUMN EXCEPT THE CREDIT CARD NUMBER.
- Application features that **UPDATE** A CREDIT CARD RECORDS BY CHANGING THE NAME & ADDRESS FOR A PARTICULAR CREDIT CARD NUMBER.
- Application features that **UPDATE** CREDIT CARD RECORDS OF CUSTOMERS BY CHANGING THE **CREDITCARDPROCESSINGMERCHANTSERVICECOMPANYCODE** COLUMN FOR A CREDIT CARD.

State Of CREDITCARD and CREDITCARDPROCESSINGMERCHANTSERVICECOMPANY Tables before updates:

Using the following statements developers query the states of each table used in the test:

```
Select * From CREDITCARD;
Select * From CREDITCARDPROCESSINGMERCHANTSERVICECOMPANY;
```

Result:

Results Messages

	CreditCardNumber	CreditCardOwnerName	CreditCardProcessingMerchantServiceCompanyCode	CreditCardNetworkCompanyCode	CreditCardIssuingBankCode	CreditCardCorporateMerchantBankCode	ExpDate	AddressLine1	AddressLine2	City	StateCode	ZipCode	Country	CreditCardLimit	CreditCardAvailableCredit	ActivationStatus
1	1111111111111111	Ryan Brown	8	2	2	1	2025-01-01	111 Jay Street	Suite 101	Freehold	NJ	17711	USA	3000.00	1000.00	1
2	2222222222222221	Alex Rodriguez	7	6	4	2	2027-02-02	222 Glenwood Rd	Apt 6H	Brooklyn	NY	11222	USA	10000.00	8000.00	1
3	2222222222222222	Alex Rodriguez	8	2	1	2	2027-02-02	222 Glenwood Rd	Apt 6H	Brooklyn	NY	11222	USA	10000.00	8000.00	1
4	2222222222222223	Alex Rodriguez	11	4	2	3	2027-02-02	222 Glenwood Rd	Apt 6H	Brooklyn	NY	11222	USA	10000.00	8000.00	1
5	3333333333333333	Michelle Apicotta	1	3	5	1	2024-03-03	333 5th Avenue	NULL	New York	NY	10033	USA	3000.00	3000.00	0
6	4444444444444444	Mike Greene	6	11	8	3	2029-04-04	444 Flatlands Ave	3rd Floor	Allentown	PA	14344	USA	5000.00	3000.00	1
7	5555555555555555	Sandra Lopez	8	7	1	1	2030-05-05	6th Avenue	Apt 4f	Jersey City	NJ	07032	USA	10000.00	500.00	1

	CreditCardProcessingMerchantServiceCompanyCode	CreditCardProcessingMerchantServiceCompanyName
1	1	Stax by Fattmerchant
2	2	Helcim
3	3	Dharma Merchant Services
4	4	Payment Depot
5	5	National Processing
6	6	Block
7	7	Intuit Quickbooks
8	8	PayPal
9	9	Stripe
10	10	Flagship Merchant Services
11	11	Clover

Updating one record on the CREDITCARD Table

Developers used the following update statement to make changes to a row for all columns (Even Foreign Keys) except the CreditCardNumber:

```
UPDATE CREDITCARD
SET CreditCardOwnerName='Sandra German-Lopez',
        CreditCardProcessingMerchantServiceCompanyCode = 4,
        CreditCardNetworkCompanyCode =2,
        CreditCardIssuingBankCode=3,
        CreditCardCorporateMerchantBankCode=2,
        ExpDate='05/05/2032',
        AddressLine1 = '123 Villa Lane',
        AddressLine2=null,
        City='Nueva Villa',
        StateCode= 'PA',
        ZipCode ='14132',
        Country= 'USA',
        CreditCardLimit=250000.00,
        CreditCardAvailableCredit=245000.00,
        ActivationStatus=1
        WHERE CreditCardNumber ='5555555555555555';
```

Result:

Developers are able to prove that they can make changes to all fields in a record on the **CREDITCARD** Table without alternating other records. Below is a screen shot of query of the **CREDITCARD** Table after the **UPDATE**.

CreditCardNumber	CreditCardOwnerName	CreditCardProcessingMerchantServiceCompanyCode	CreditCardNetworkCompanyCode	CreditCardIssuingBankCode	CreditCardCorporateMerchantBankCode	ExpDate	AddressLine1	AddressLine2	City	StateCode	ZipCode	Country	CreditCardLimit	CreditCardAvailableCredit	ActivationStatus
1111111111111111	Ryan Brown	8	2	2	1	2025-01-01	111 Jay Street	Suite 101	Freehold	NJ	17711	USA	3000.00	1000.00	1
2222222222222221	Alex Rodriguez	7	6	4	2	2027-02-02	222 Glenwood Rd	Apt 6H	Brooklyn	NY	11222	USA	10000.00	8000.00	1
2222222222222222	Alex Rodriguez	8	2	1	2	2027-02-02	222 Glenwood Rd	Apt 6H	Brooklyn	NY	11222	USA	10000.00	8000.00	1
2222222222222223	Alex Rodriguez	11	4	2	3	2027-02-02	222 Glenwood Rd	Apt 6H	Brooklyn	NY	11222	USA	10000.00	8000.00	1
3333333333333333	Michelle Apicotta	1	3	5	1	2024-03-03	333 5th Avenue	NULL	New York	NY	10033	USA	3000.00	3000.00	0
4444444444444444	Mike Greene	6	11	8	3	2029-04-04	444 Flatlands Ave	3rd Floor	Allentown	PA	14344	USA	5000.00	3000.00	1
5555555555555555	Sandra German-Lopez	4	2	3	2	2032-05-05	123 Villa Lane	NULL	Nueva Villa	PA	14132	USA	250000.00	245000.00	1

Updating one record on the CREDITCARD Table

State of table before update:

SQL statement used to query state of table:

```
SELECT * FROM CREDITCARD;
```

CreditCardNumber	CreditCardOwnerName	CreditCardProcessingMerchantServiceCompanyCode	CreditCardNetworkCompanyCode	CreditCardIssuingBankCode	CreditCardCorporateMerchantBankCode	ExpDate	AddressLine1	AddressLine2	City	StateCode	ZipCode	Country	CreditCardLimit	CreditCardAvailableCredit	ActivationStatus
1111111111111111	Ryan Brown	8	2	2	1	2025-01-01	111 Jay Street	Suite 101	Freehold	NJ	17711	USA	3000.00	1000.00	1
2222222222222221	Alex Rodriguez	7	6	4	2	2027-02-02	222 Glenwood Rd	Apt 6H	Brooklyn	NY	11222	USA	10000.00	8000.00	1
2222222222222222	Alex Rodriguez	8	2	1	2	2027-02-02	222 Glenwood Rd	Apt 6H	Brooklyn	NY	11222	USA	10000.00	8000.00	1
2222222222222223	Alex Rodriguez	11	4	2	3	2027-02-02	222 Glenwood Rd	Apt 6H	Brooklyn	NY	11222	USA	10000.00	8000.00	1
3333333333333333	Michelle Apicotta	1	3	5	1	2024-03-03	333 5th Avenue	NULL	New York	NY	10033	USA	3000.00	3000.00	0
4444444444444444	Mike Greene	6	11	8	3	2029-04-04	444 Flatlands Ave	3rd Floor	Allentown	PA	14344	USA	5000.00	3000.00	1
5555555555555555	Sandra German-Lopez	4	2	3	2	2032-05-05	123 Villa Lane	NULL	Nueva Villa	PA	14132	USA	250000.00	245000.00	1

Update Statements:

Developers use the following update statement to make changes to row with primary key = 5555555555555555 for the following columns CreditCardOwnerName, AddressLine1, AddressLine2, City, StateCode, ZipCode, Country:

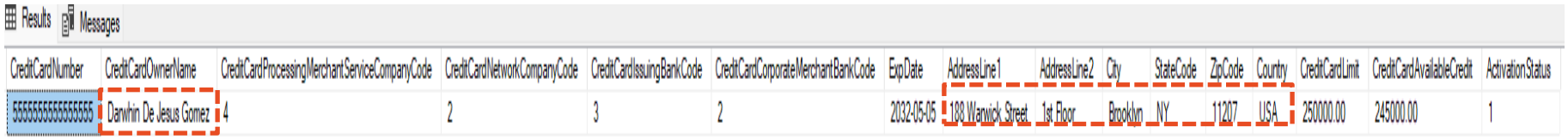
```
UPDATE CREDITCARD

SET CreditCardOwnerName= 'Darwhin De Jesus Gomez',
        AddressLine1 = '188 Warwick Street',
        AddressLine2= '1st Floor',
        City= 'Brooklyn',
        StateCode = 'NY',
        ZipCode = '11207',
        Country =  'USA'
        WHERE CreditCardNumber =  '5555555555555555' ;
```

Resulting state of table:

Developers are able to prove that they can make changes to **targeted** fields in a record on the **CREDITCARD** Table. Below is the Select query used to view the resulting record and screen shot the output of the **CREDITCARD** Table for record with primary key 5555555555555555 after the **UPDATE** proving developers can make changes to targeted columns without affecting others.

```
SELECT * FROM CREDITCARD
WHERE CreditCardNumber = '5555555555555555';
```



Results | Messages

CreditCardNumber	CreditCardOwnerName	CreditCardProcessingMerchantServiceCompanyCode	CreditCardNetworkCompanyCode	CreditCardIssuingBankCode	CreditCardCorporateMerchantBankCode	ExpDate	AddressLine1	AddressLine2	City	StateCode	ZipCode	Country	CreditCardLimit	CreditCardAvailableCredit	ActivationStatus
5555555555555555	Danwhin De Jesus Gomez	4	2	3	2	2032-05-05	188 Warwick Street	1st Floor	Brooklyn	NY	11207	USA	250000.00	245000.00	1

Updating CREDITCARD records with a NEW CREDITCARDPROCESSINGMERCHANTSERVICECOMPANYCODE

At this point of the validation process the developers are proving that it is possible to add new merchants to the database. In order to this, first make the required insertion into the CREDITCARDPROCESSINGMERCHANTSERVICECOMPANY Table. This is Because the CREDITCARD Table host a a foreign key constraint column that is the primary key of the CREDITCARDPROCESSINGMERCHANTSERVICECOMPANY. **This column is the**

SQL STATEENT USE TO SHOW STATE:

```
SELECT * FROM CREDITCARDPROCESSINGMERCHANTSERVICECOMPANY;
```

State of the CREDITCARD **Table before insertion of new record MERCHANT Service Company**:

CreditCardProcessingMerchantServiceCompanyCode	CreditCardProcessingMerchantServiceCompanyName
1	Stax by Fattmerchant
2	Helcim
3	Dharma Merchant Services
4	Payment Depot
5	National Processing
6	Block
7	Intuit Quickbooks
8	PayPal
9	Stripe
10	Flagship Merchant Services
11	Clover

SQL STATEMENT USED TO INSERT NEW CREDIT CARD PROCESSING MERCHANT SERVICE COMPANY:

```
INSERT INTO CREDITCARDPROCESSINGMERCHANTSERVICECOMPANY
VALUES (12,'US Merchant Processing Services');
```

Resuls after insertion into CREDITCARDPROCESSINGMERCHANTSERVICECOMPANY:

SQL STATEMENT used to shoe the **CREDITCARDPROCESSINGMERCHANTSERVICECOMPANY** after changes:

```
SELECT * FROM CREDITCARDPROCESSINGMERCHANTSERVICECOMPANY
ORDER BY CreditCardProcessingMerchantServiceCompanyCode;
```

Below it is shown there is a new Credit Card Processing Merchant Service Company Named USM Processing Services and assigned the **CreditCardProcessingMerchantServiceCompanyCode of 12.**

Results | Messages

CreditCardProcessingMerchantServiceCompanyCode	CreditCardProcessingMerchantServiceCompanyName
1	Stax by Fattmerchant
2	Helcim
3	Dharma Merchant Services
4	Payment Depot
5	National Processing
6	Block
7	Intuit Quickbooks
8	PayPal
9	Stripe
10	Flagship Merchant Services
11	Clover
12	USM Processing Services

Now the CREDITCARD Table can begin to handle records with a **CreditCardProcessingMerchantServiceCompanyCode of 12, because now it can match it'[s foreign key constraint to an existing primary key in the** CREDITCARDPROCESSINGMERCHANTSERVICECOMPANY Table

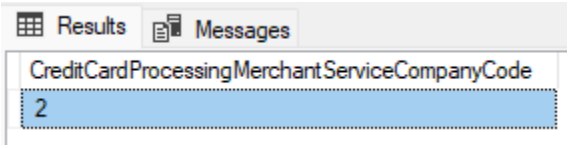
Updating CREDITCARD Table for CreditCardNumber = 1111111111111111 and changing its CreditCardProcessingMerchantServiceCompanyCode from 8 to 12

Before updating of record on CREDITCARD Table for CreditCardNumber = 1111111111111111:

The Following SQL SELECT statement was used to query database for the record:

```
SELECT CreditCardProcessingMerchantServiceCompanyCode FROM CREDITCARD
WHERE CreditCardNumber ='1111111111111111';
```

Result



SQL UPDATE STATEMENT to make desired changes:

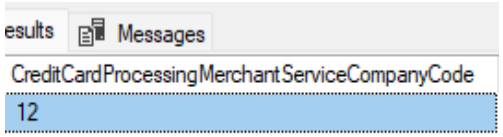
```
UPDATE CREDITCARD
SET CreditCardProcessingMerchantServiceCompanyCode = 12
WHERE CreditCardNumber='1111111111111111';
```

Result of the Update:

The Following SQL SELECT statement was used to query database for the record:

```
SELECT CreditCardProcessingMerchantServiceCompanyCode FROM CREDITCARD
WHERE CreditCardNumber ='1111111111111111';
```

The results show that following the proper steps the database is designed to be adaptable while also maintaining data integrity by restricting records added to the CREDITCARD table for columns of foreign keys if those records don't already exist in the corresponding parent tables.



Deleting Records From Tables

Here developers will test and validate the Database functionality in deleting Records based on these scenarios:

- Application features that DELETE A **CREDITCARD** RECORD OF A CUSTOMER WHO NO LONGER WANTS TO BE A CUSTOMER.
- Application features that DELETE A RECORD FROM THE **CREDITCARDPROCESSINGMERCHANTSERVICECOMPANY** TABLE WITHOUT HAVING TO DELETE ALL THE **CREDITCARD** RECORDS RELATED TO THE DELETED **CREDITCARDPROCESSINGMERCHANTSERVICECOMPANY** TABLE RECORD.

Deleting a record of customer who no longer wishes to be a customer.

First look at the current state of the CREDITCARD TABLE and highlight the customer record to DELETE using a simple select statement.

```
SELECT CreditCardOwnerName,CreditCardNumber,
CreditCardProcessingMerchantServiceCompanyCode FROM CREDITCARD;
```

Results | Messages

	CreditCardOwnerName	CreditCardNumber	CreditCardProcessingMerchantServiceCompanyCode
1	Ryan Brown	1111111111111111	2
2	Alex Rodriguez	2222222222222221	7
3	Alex Rodriguez	2222222222222222	8
4	Alex Rodriguez	2222222222222223	11
5	Michelle Apicotta	3333333333333333	1
6	Mike Greene	4444444444444444	6
7	Darwhin De Jesus Gomez	5555555555555555	4

Michelle Apicotta is now retired and no longer needs to travel for work so she will be ending her relationship with EZRENTAL.

Execution of The Delete statement:

Then developers build the following DELETE Statement and execute it to make the changes to the Table;

```
DELETE FROM CREDITCARD
WHERE CreditCardNumber = '3333333333333333';
SELECT CreditCardOwnerName,CreditCardNumber,
CreditCardProcessingMerchantServiceCompanyCode FROM CREDITCARD;
```

Results of the Delete Statement:

Results | Messages

	CreditCardOwnerName	CreditCardNumber	CreditCardProcessingMerchantServiceCompanyCode
1	Ryan Brown	1111111111111111	2
2	Alex Rodriguez	2222222222222221	7
3	Alex Rodriguez	2222222222222222	8
4	Alex Rodriguez	2222222222222223	11
5	Mike Greene	4444444444444444	6
6	Darwhin De Jesus Gomez	5555555555555555	4

The record of Michelle Apicotta has been Deleted

DELETE A RECORD FROM THE CREDITCARDPROCESSINGMERCHANTSERVICECOMPANY TABLE WITHOUT HAVING TO DELETE ALL THE CREDITCARD RECORDS RELATED TO THE DELETED CREDITCARDPROCESSINGMERCHANTSERVICECOMPANY TABLE RECORD

In order to achieve this work is need circumvent the cascade feature that was originally built between the two Tables this can be done by inserting a new record into **CREDITCARDPROCESSINGMERCHANTSERVICECOMPANY** table as a place holder, then changing the records in the CREDITCARD Table that belong to that **MERCHANT SERVICE COMPANY** being placed to the place holder company in the **CREDITCARDPROCESSINGMERCHANTSERVICECOMPANY** table.

First, insert a place holder Company into the parent **CREDITCARDPROCESSINGMERCHANTSERVICECOMPANY** TABLE give it code 13 and name it 'Holding' with the following statement:

```
INSERT INTO CREDITCARDPROCESSINGMERCHANTSERVICECOMPANY
VALUES (13,'Holding');
```

Result:

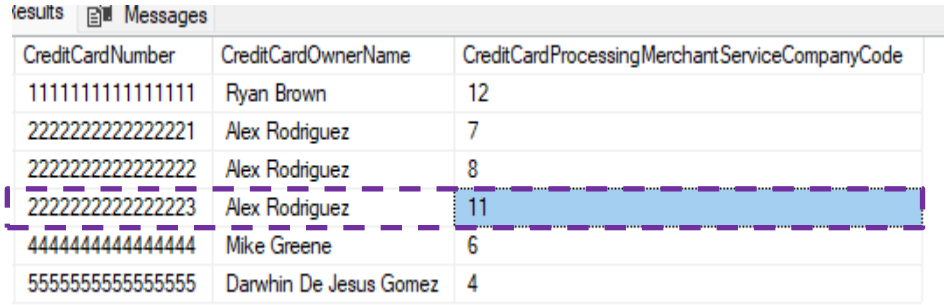
CreditCardProcessingMerchantServiceCompanyCode	CreditCardProcessingMerchantServiceCompanyName
1	Stax by Fattmerchant
2	Helcim
3	Dharma Merchant Services
4	Payment Depot
5	National Processing
6	Block
7	Intuit Quickbooks
8	PayPal
9	Stripe
10	Flagship Merchant Services
11	Clover
12	USM Processing Services
13	Holding

With this record now in the Parent table(CREDITCARDPROCESSINGMERCHANTSERVICECOMPANY), now change records in **CREDITCARD** to this Foreign Key

EZRENTAL No longer has a relationship with company 11 "CLOVER"

STATE of the CREDITCARD Table:

```
SELECT CreditCardNumber, CreditCardOwnerName,
CreditCardProcessingMerchantServiceCompanyCode
FROM CREDITCARD;
```



Results | Messages

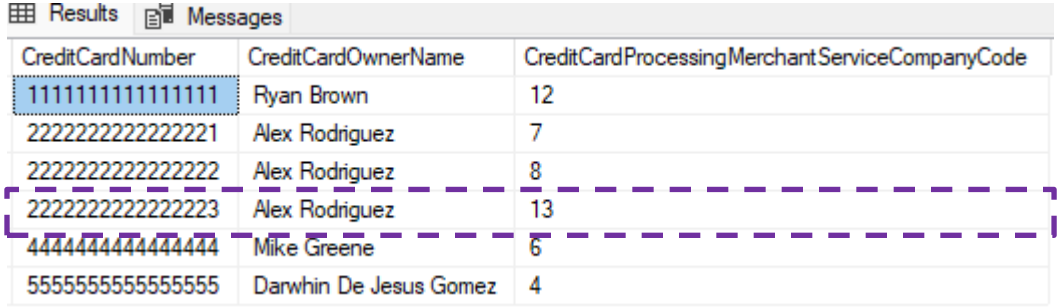
CreditCardNumber	CreditCardOwnerName	CreditCardProcessingMerchantServiceCompanyCode
1111111111111111	Ryan Brown	12
2222222222222221	Alex Rodriguez	7
2222222222222222	Alex Rodriguez	8
2222222222222223	Alex Rodriguez	11
4444444444444444	Mike Greene	6
5555555555555555	Darwhin De Jesus Gomez	4

EZRental has terminated it relationship with CLOVER change any record in **CREDITCARD** Table with **CreditCardProcessingMerchantServiceCompanyCode** = 11 TO the Holding number 13:

```
UPDATE CREDITCARD

SET CreditCardProcessingMerchantServiceCompanyCode=13
WHERE CreditCardProcessingMerchantServiceCompanyCode=11;
```

RESULTS



Results | Messages

CreditCardNumber	CreditCardOwnerName	CreditCardProcessingMerchantServiceCompanyCode
1111111111111111	Ryan Brown	12
2222222222222221	Alex Rodriguez	7
2222222222222222	Alex Rodriguez	8
2222222222222223	Alex Rodriguez	13
4444444444444444	Mike Greene	6
5555555555555555	Darwhin De Jesus Gomez	4

All records that had Code 11 have now been updating to the Holding Code 13.

DELETE From CREDITCARDPROCESSINGMERCHANTSERVICECOMPANY

State of table Before and AFTER the DELETE:

```
SELECT * from CREDITCARDPROCESSINGMERCHANTSERVICECOMPANY

ORDER BY CreditCardProcessingMerchantServiceCompanyCode;
```

BEFORE

CreditCardProcessingMerchantServiceCompanyCode	CreditCardProcessingMerchantServiceCompanyName
1	Stax by Fattmerchant
2	Helcim
3	Dharma Merchant Services
4	Payment Depot
5	National Processing
6	Block
7	Intuit Quickbooks
8	PayPal
9	Stripe
10	Flagship Merchant Services
11	Clover
12	USM Processing Services
13	Holding

EXECUTE the Deletion of the company can be achieved with the following SQL DELETE Statement:

```
DELETE FROM CREDITCARDPROCESSINGMERCHANTSERVICECOMPANY
WHERE CreditCardProcessingMerchantServiceCompanyCode=11;
```

After

```
SELECT * from CREDITCARDPROCESSINGMERCHANTSERVICECOMPANY

ORDER BY CreditCardProcessingMerchantServiceCompanyCode;
```

CreditCardProcessingMerchantServiceCompanyCode	CreditCardProcessingMerchantServiceCompanyName
1	Stax by Fattmerchant
2	Helcim
3	Dharma Merchant Services
4	Payment Depot
5	National Processing
6	Block
7	Intuit Quickbooks
8	PayPal
9	Stripe
10	Flagship Merchant Services
12	USM Processing Services
13	Holding

11,"Clover " Has ben Deleted

Original records in the CREDITCARD Table Have been preserved under a HOLDING company with code 13:

```
SELECT CreditCardNumber, CreditCardOwnerName,
CreditCardProcessingMerchantServiceCompanyCode
FROM CREDITCARD;
```

Results Messages

CreditCardNumber	CreditCardOwnerName	CreditCardProcessingMerchantServiceCompanyCode
1111111111111111	Ryan Brown	12
2222222222222221	Alex Rodriguez	7
2222222222222222	Alex Rodriguez	8
2222222222222223	Alex Rodriguez	13
4444444444444444	Mike Greene	6
5555555555555555	Darwhin De Jesus Gomez	4

Conclusion

Design and Development of the Database

Throughout the design and development process, all business requirements were addressed, ensuring a comprehensive and robust database solution. Prototype schemas were crafted, tested rigorously for functionality, and refined to meet the standards of efficiency and effectiveness.

The iterative nature of schema development allowed for careful consideration of various business scenarios, leading to a database structure that aligns seamlessly with the operational needs of the organization. The thorough testing ensures that the database not only meets but exceeds expectations in terms of reliability and performance.

As a result, the database delivers a solution that not only meets the present business requirements but also lays a solid foundation for future scalability and adaptability. The careful consideration of functionality, coupled with the testing and refinement process, ensures that the database is well-positioned to support the organization's data management needs effectively.

In conclusion, the collaborative efforts of the team have yielded a database that is not only technically sound but also aligned with the overarching goals of the EZRENTAL business.