

AUTO RENTAL MANAGEMENT SYSTEM
EZRENTAL POS

Mohamed Kaba

NYC_TECH

TABLE OF CONTENT

1. Executive Summary.....	3
2. Problem Statement & Objectives.....	4
3. Project Management Methodology.....	5
4. Database Design Deliverable #1a – Application Business requirements.....	8
5. Database Design Deliverable #1b – Application Development Technical Requirements.....	21
6. Application Physical Technical Architecture.....	31
7. Application Development Features and Functionalities (Agile Backlog)	33
8. Database Management System Development Environment & Physical Architecture.....	46
9. Project Roles & Responsibilities.....	47
10. Database Design Deliverable #2 – ER/EER Conceptual Model Diagram.....	49
11. Database Design Deliverable #3 – Normalized Logical Model Diagram.....	50
12. Database Design Deliverable #4 – Physical Model Data Dictionary.....	52
13. Database Design Deliverable #5 – Physical Model Schema Design Diagram.....	53
14. Database Implementation Deliverable #6 – Development & Implementation.....	54
15. Database Implementation Deliverable #7 – Implemented Physical Schema Diagram.....	55
16. Database Implementation Deliverable #8 – Database Validation Testing.....	56
17. Conclusion.....	57

PROJECT 1 – EZRental Auto Rental POS Management System

Database Design and Implementation

Executive Summary

We are planning to develop an auto rental company that rents vehicles such as cars, SUV minivans & cargo vans to customers. In addition, other specialized vehicles such as trucks, motorcycles, boats, mobile homes, etc

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

Problem Statement & Objectives

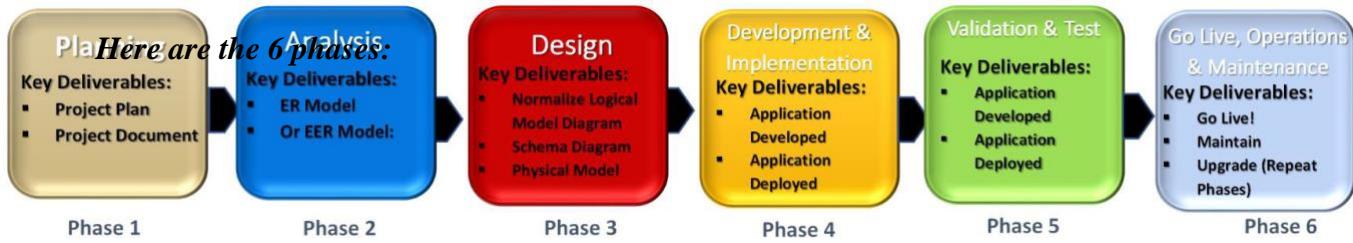
The main goal of this project is to design and implement a suite of Auto Rental Point of Sales Management System Application that include some business modules. The EZRental Auto Management System will have advanced features that are designed to allow customers, both retail and corporate customer, to reserve vehicles for renting, in person or online. Also, It will allow customers to make and manage vehicle reservations, profile and account via the public internet.

The application will be designed to support dozens of major cities around the world. In addition, it will provide 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

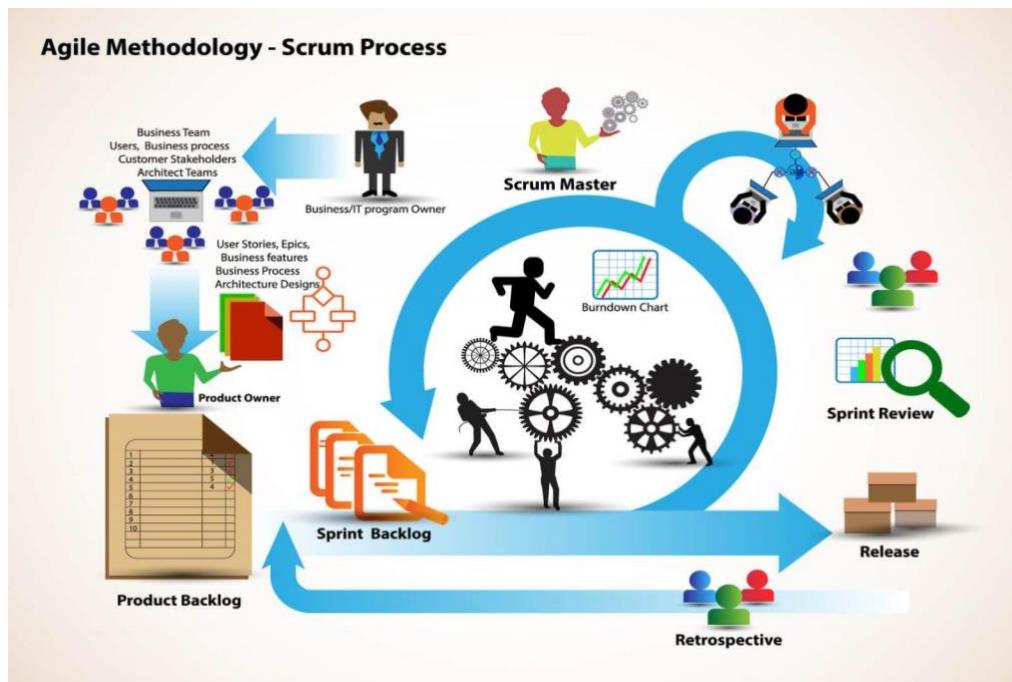
Project Management Methodology

The Project Management Methodology for the Auto Rental Management System will be implemented by using a mix of **Waterfall** and **Agile**. The methodology will provide a roadmap and methods to successfully deliver the project.

The **Waterfall Methodology** moves from stage to stage and follow a sequential project management where the project is divided into several phases.

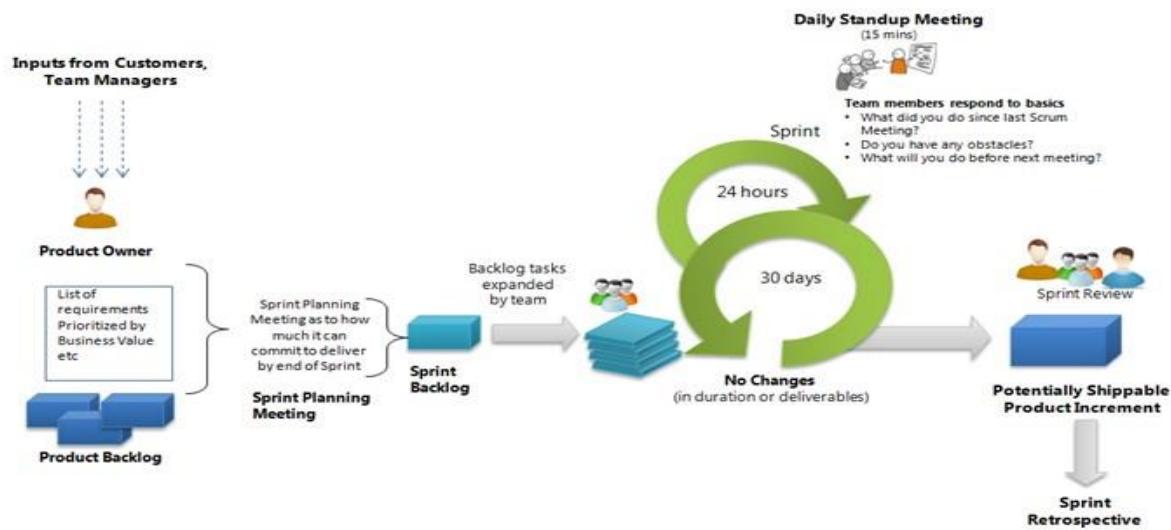


Agile methodology is a rapid development results-focused approach. It consists of dividing and executing the program in increments or subset called a Sprint. For each Sprint, the duration is 1, 2, 3 or 4 weeks,



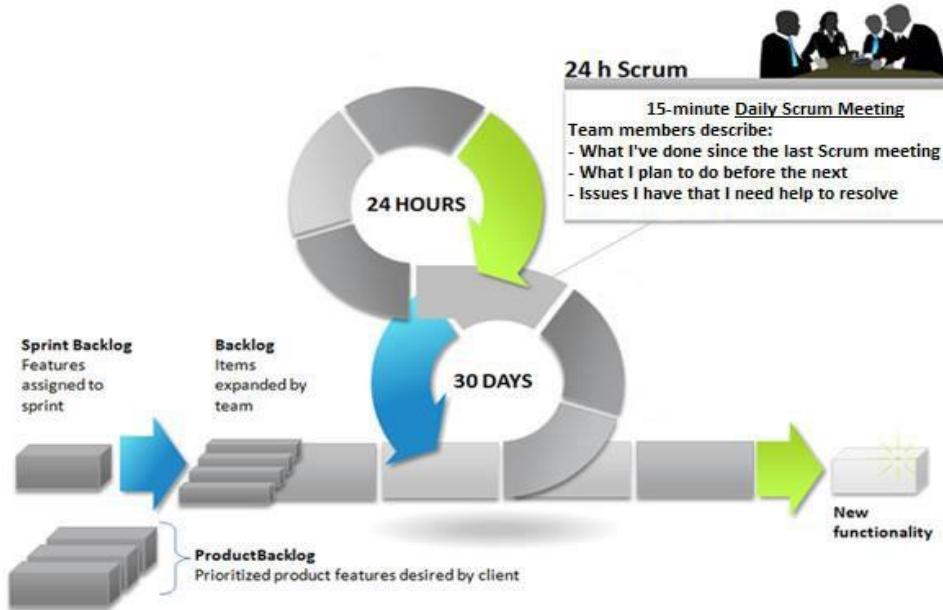
- ❖ A second pictorial illustrations of Agile SCRUM project Management Methodology:

Agile Scrum Methodology

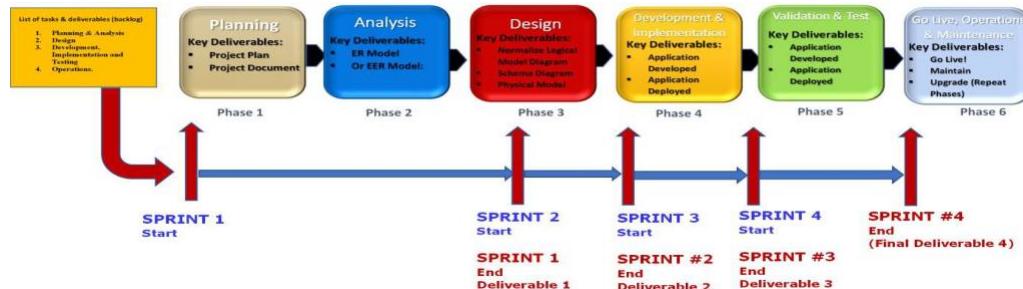


- A third pictorial illustration of **Agile SCRUM** Project Management Methodology:

SCRUM PROCESS

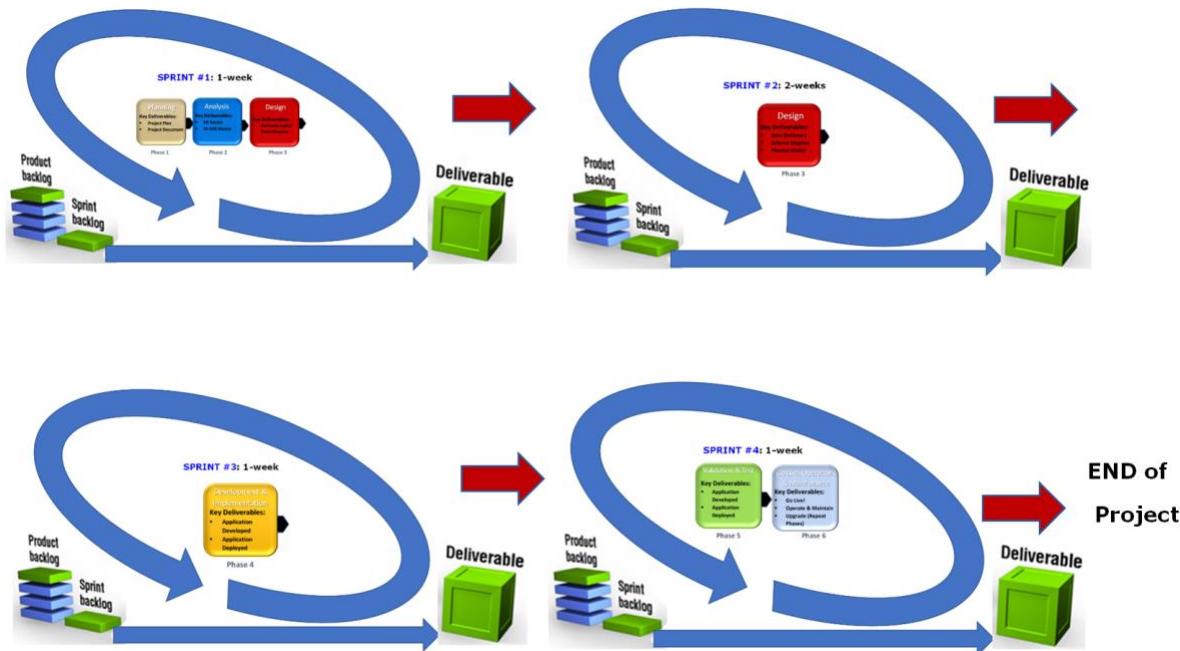


Waterfall and Agile methodology has been combined by using sprints. The combination break up each of the database development and implementation waterfall phases as SPRINTS deliverables.



AGILE SPRINT #	WATERFALL PHASE	Output Deliverable
SPRINT #1	Planning	1. Create Project Document 2. Business Requirements List of Business & Technical Requirements from customer.
	Analysis	- ER/EER Conceptual Model Diagram
	Design Phase (Part 1)	- Normalized Logical Model Diagram
SPRINT #2	Design Phase (Part 2)	- Data Dictionary matrix - Physical Schema Design Diagram – from Normalize Logical Model + DataDictionary combination.
SPRINT #3	Development & Implementation	- Database application developed & implemented - Generate the actual Physical Schema Diagram – from the Database & compared to the Physical Schema Design Diagram
SPRINT #4	Validation & Testing	- Unit & Integration testing.
	Operations	- Operations

Second, we will break up the Waterfall Phased into SPRINT through the Agile Methodology Process for a total of 4 Sprints with a total of 5-Weeks shown in diagram below



Application Business Requirements

The Business Requirements are the FOUNDATION of the database design and used by the Database Analyst/Architect to engage in the first step of Database Design called DATA MODELING. ➤ In data modeling, from the business requirements we derived the following Diagrams:

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 AGENCIES:

A **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.

All Customers (Retail & Corporate Customers)

To run our business, the application must store the following customer information for both types of **customer** (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)**, **driver license expiration Date** and **driver license state**. In addition, note the following business rule on 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 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**. A credit card includes the following attributes: **credit card number** that uniquely identifies the credit card and is a 16-character number digits, **credit card owner name, credit card issuing company name** (such as American Express, Visa, MasterCard, Capital One, etc.), **merchant Code & merchant name** which is the credit card payment processing company that acts as an intermediary between our business and the customers' credit card companies or bank. The merchant handles the interaction between the purchase of a rental and the credit card company etc., validating credit card transaction. This merchant Code & Name attributes have business meaning and used throughout the business using a digit code for **merchant Code** and the name of the merchant associated with the code or **merchant name**. We currently use the following **merchant code** and **merchant names** throughout the world to handle our credit card processing:

<i>Merchant Code</i>	<i>Merchant Name</i>
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

- 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**.
- In addition, **credit card limit, credit card balance & activation status** which is true if the credit card is active and can be used or false when disabled.
- During the interview with business stakeholders we captured the following **Business Rules** related to a credit card:

- You cannot reserve or rent one of our vehicles without a credit card**
- A customer can have many credit cards they can use to pay for rental transactions.**
- 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.**

Business Requirements

Our Customers (Cont.):

- A very important attribute we need to capture for every customer is the **credit card**. A credit card includes the following attributes: **credit card number** that uniquely identifies the credit card and is a 16-character number digits, **credit card owner name**, **credit card issuing company name** (such as American Express, Visa, MasterCard, Capital One, etc.), **merchant Code & merchant name** which is the credit card payment processing company that acts as an intermediary between our business and the customers' credit card companies or bank. The merchant handles the interaction between the purchase of a rental and the credit card company etc., validating credit card transaction. This merchant Code & Name attributes have business meaning and used throughout the business using a digit code for **merchant Code** and the name of the merchant associated with the code or **merchant name**. We currently use the following **merchant code** and **merchant names** throughout the world to handle our credit card processing:

<i>Merchant Code</i>	<i>Merchant Name</i>
1	Stax by Fatmerchant
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

- 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**.
- In addition, **credit card limit**, **credit card balance** & **activation status** which is true if the credit card is active and can be used or false when disabled.
- During the interview with business stakeholders we captured 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.

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 stored 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 Customers can (but don't have to) leverage promotional **discounts** or coupons obtain from other businesses, internet, magazine, organizations, etc., to save money on their rentals. Therefore, data unique to a retail customer that we need to capture for the promotional discount is unique random number **discount ID** which uniquely identifies a discount, a 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..

Retail customers can opt-in to enrolled in the **EZPlus Rewards Program** where they earn points every time they rent and these 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..	Etc..

In this business, we have the following rules for our customers:

- 1. We only have two types of customers retail customer or corporate customers. No other type of customer exists.**
- 2. 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.**

Business Requirements (Cont.)

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:

1. *A reservation/rental can only be for one of these four categories of Vehicles or other vehicle types, not a combination.*
2. *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:

1. *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.*
2. *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:

1. *A reservation is NOT made for a specific vehicle, but to a vehicle rental category. Rental category examples are economy, intermediate, full size, luxury.*
2. *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:

<i>Vehicle Rental Category ID</i>	<i>Vehicle Rental Category Name</i>	<i>Category Daily Rental Rate</i>
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.

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 **reservation pick up date**, **reservation pick up time**, **reservation drop off date** and **reservation drop off time**, also the **reservation estimated rental cost**.

Business Requirements (Cont.)

Reservation Process (Cont.):

- 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) Finally, 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.
- ❖ 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, 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 <i>(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.)</i>
2	Pay for full tank in advanced at time of rental, return car empty. No refund for unused gas.	\$45.99 <i>(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.)</i>

- 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	Liability 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**.
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.
 4. Therefore, the **EmployeeID** is the unique ID for an employee to be identified and managed from a business perspective.

Security & Application Access:

To access our systems proper security and authentication is required. Only authorized users can have access our agencies Point-Of-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-Of-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 evolves, 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

An Application Analyst/Architect was hired by Mr. Rodriguez to interview EZRental Inc., project Business Decision Makers (BDMs), stakeholders and Information System Technical Decision Makers (TDMs) to compile the list of the Application Development & Technical Requirements in order to design & develop the application

Application Development & Technical Requirements

Introduction & Current Challenges

As described in the Business Requirements, the current rental system is outdated, with a poor user-experience, breaks often thus expensive to operate, does not meet our business requirements, and is not scalable so it cannot be easily updated with new features etc. Also, not elastic since it does not give us the flexibility to scale-up or scale-down based on 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 give us a great user-experience, meet new business requirements, scalable, and elastic to adopt to business trends and seasonal market changes.

We have an outdated IT infrastructure in our datacenter and there is a current initiative to modernize our datacenter and also leverage cloud technology in a hybrid environment to save on cost, streamline our operations and drive innovation.

We look forward to your proposed architecture & implementation of this new system that will meet these requirements. Next sections contain the results of our application development & technical requirements.

Rental Agencies Application & Technical Requirements:

The rental agencies are location where customers both Retail & Corporate will engage our *Customer Service Representatives* to engage in rental/return activities in addition to other transactions such as registering, searching & updating customer information etc. Therefore, the application in the rental agencies is vital to the user-experience for both our *Customer Service Representatives* as well as our *customers*.

We are forecasting that in some locations such as major city centers and airports, there will be many customers engaging throughout the day thus increasing the risk of a poor customer experience in addition to the work overload and poor experience for our *Customer Service Representatives*. We want our *Customers* to be serviced quickly and efficiently with a great experience, and our *Customer Service Representatives* to be able to process each *Customer* easily and effectively. With these criteria in mind, the application at our rental agencies must adhere to the following requirements:

Rental Agency Application Architecture Requirements:

Below are the requirements for the application used in our rental agencies by our customer service representatives, inventory team, service personnel and other employees working in our agencies:

1. Client application processing, transaction and response must be fast to minimize service time for a customer.
2. All transaction processing should be done in the user's computer or desktop for fast processing and response.
3. Application Architecture must be reusable and scalable to support future updates and new feature enhancements, without a long development lifecycle.
4. Depending on the architecture NYC-Tech Solutions Inc., decides for the application in the rental agencies (Desktop client or Web client), the primary Application Development Platform we use is **C# & .NET technologies**. For any Web related development, we support JavaScript, React, NodeJs and other standard Web Technologies. We have aligned **C#.NET & ASP.NET Web developers** that have been assigned to assist, support and update the application once NYCTech consultants complete the project and development of this system.
5. Rental Agency Desktop Application Security Authentication System – Proper security and authentication must be implemented to make sure only authorized customer service representative and other rental office employees can access the Point-of-Sales with appropriate conditional access.

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	<ul style="list-style-type: none">▪ 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	<ul style="list-style-type: none">▪ 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	<ul style="list-style-type: none">▪ 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	<ul style="list-style-type: none">▪ 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	<ul style="list-style-type: none">▪ 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.):

- o **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.

- o **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.

- o **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.

- o **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.):

- o **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:

<i>Vehicle Rental Category ID</i>	<i>Vehicle Rental Category Name</i>	<i>Category Daily Rental Rate</i>
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.

- o **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:

<i>Reservation Status ID</i>	<i>Reservation Status Description</i>
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.):

- o **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:

<i>Rental Status ID</i>	<i>Rental Status Description</i>
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.

- o **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:

<i>Rental Fuel Option ID</i>	<i>Rental Fuel Option Description</i>	<i>Rental Fuel Option Additional Cost</i>
1	Return with a full tank or on return, pay for gas that is missing.	\$13.97 <i>(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.)</i>
2	Pay for full tank in advanced at time of rental, return car empty. No refund for unused gas.	\$45.99 <i>(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.)</i>

- 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

- o **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:

<i>Rental Insurance Option ID</i>	<i>Rental Insurance Option Description</i>	<i>Rental Insurance Option Additional Cost per Day</i>
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	Liability 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

- o **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.

<i>Transport Reason ID</i>	<i>Transport Reason Description</i>
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.

- o **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.

<i>Transport Status ID</i>	<i>Transport Status Description</i>
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

- o **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:

<i>Transport Reason ID</i>	<i>Transport Reason Description</i>
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.

- o **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:

<i>Transport Status ID</i>	<i>Transport Status Description</i>
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, operate 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:

The list of features and functionalities that we have compiled for the customer self-service Web Portal are listed in the table below:

No.	Feature	Functionalities
1	EZRental.com Customer Web Portal	<ul style="list-style-type: none">▪ 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	<ul style="list-style-type: none">▪ 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.):

- o **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.

- o **Rental Agency:**

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

- o **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:

<i>Vehicle Rental Category ID</i>	<i>Vehicle Rental Category Name</i>	<i>Category Daily Rental Rate</i>
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 Technical Architecture

After the design meeting with the architects and full-stack developers a decision was made on the application architecture for the EZRental POS application.

- 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 self-service:
 - 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 our web portal.
 - This also includes good user-experience.

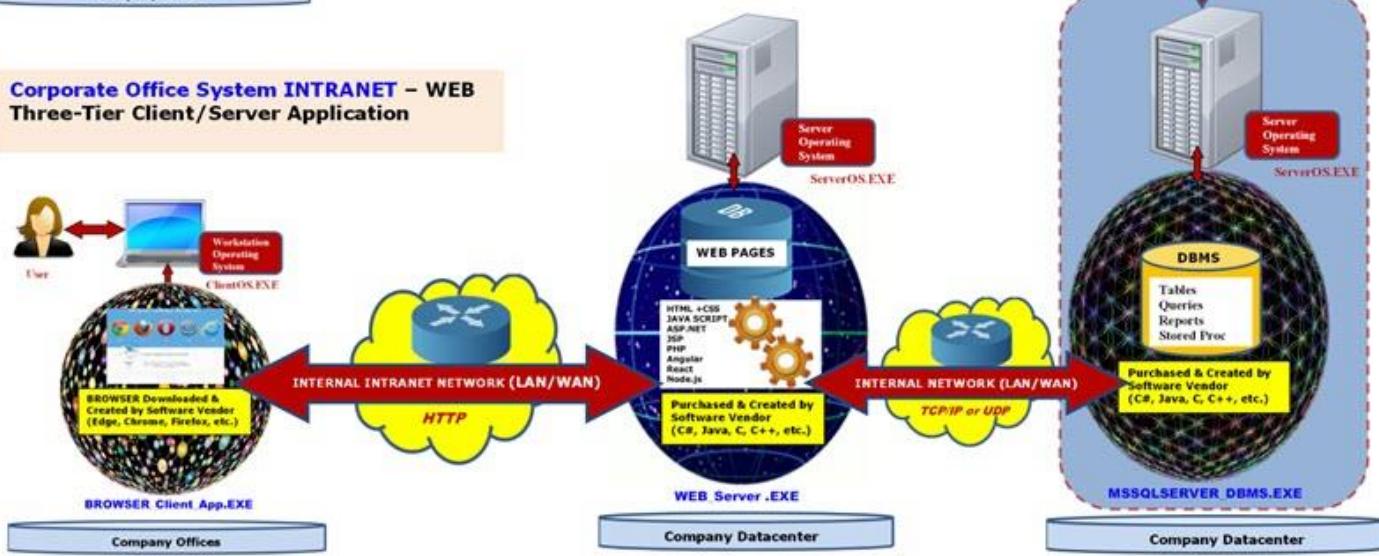
The target applications architecture and components are as follows:

- Rental Agency Two-Tiered Windows-Client 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) – Using MS SQL Server for CST4708 class and Oracle DBMS for CST3604 class. All the front-end applications (Two-Tier Window for agencies, Three-tiered Web for Corporate Offices, and ThreeTier Web for Customers Internet application) will SHARE the same DATABASE TIER. More information on the database scope will be provided in sections to follow

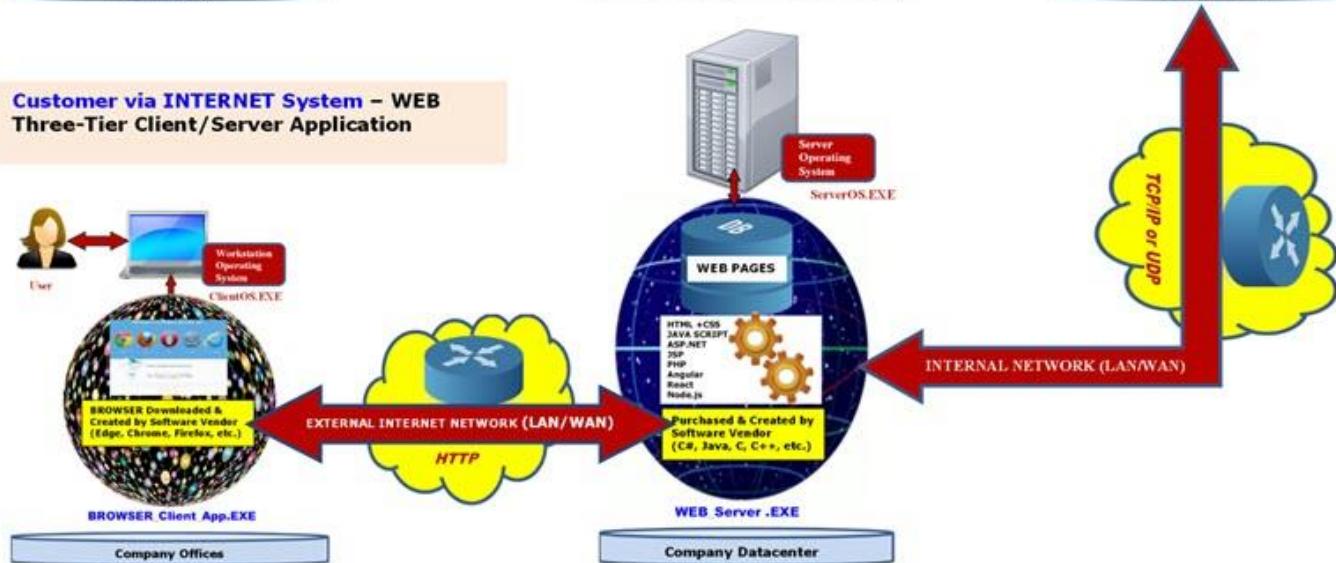
Rental Agency System – Windows Two-Tier Client/Server Application



Corporate Office System INTRANET – WEB Three-Tier Client/Server Application



Customer via INTERNET System – WEB Three-Tier Client/Server Application



Application Development Features and Functionalities (Agile Backlog)

Features & Functionality Overview

- During analysis and meetings, it was decided that the application is to deliver the following 10 features & functionality.

These 10 features together make up the AGILE BACKLOG.

Feature #	Feature Description
FEATURE #1A (OUT OF SCOPE FOR CST4708 & CST3604 COURSES BUT RECOMMENDED AS PROJECT 3, ETC., TO EXECUTE ON YOUR OWN)	FEATURE #1A – OUT OF SCOPE FOR CST4708 & CST3604 COURSES FEATURE #1A – EZRental Rental Agency Point-of-Sales (POS) System CUSTOMER SERVICE – CUSTOMER MANAGEMENT SYSTEM: ▪ WINDOWS CLIENT POINT-OF SALES SYSTEM – WINDOWS UI FORM(S) FRONT-END APPLICATION & OOP PROGRAMMING features used by customer service representative employees via the <i>Point-of-Sales computer</i> machine in the <i>Rental Agencies</i> to service customer's CUSTOMER MANAGEMENT requests or transactions. ▪ The following are features and functionality that are required for this application feature: <ul style="list-style-type: none">○ POS Customer Management Feature: POS Customer Management (Retail Customer & Corporate Customer) features such as <i>Customer Search & Print, New Customer Registration, Customer Update, Customer Deletion, & Customer Listing functionalities</i>.○ Note that each transaction is saved to database immediately after execution:<ul style="list-style-type: none">- Feature UI Form Requirements: <i>Design & programming</i> of required User-Interface Forms & GUI Controls to support this feature.- Feature Processing Requirements: <i>Design & programming</i> of required Object-Oriented (OOP) Processing & Logic to support this feature. ▪ 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: ▪ 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	<p>FEATURE #2A – EZRental Rental Agency Point-of-Sales (POS) System</p> <p>CUSTOMER SERVICE VEHICLE RESERVATION, RENTAL & RETURN FEATURE MANAGEMENT:</p> <ul style="list-style-type: none"> ▪ WINDOWS CLIENT POINT-OF SALES SYSTEM – WINDOWS UI FORM(S) FRONT-END APPLICATION & OOP PROGRAMMING features used by customer service representative employees via the <i>Point-of-Sales computer</i> machine in the Rental Agencies to service customer's CUSTOMER VEHICLE RESERVATION, RENTAL & RETURN MANAGEMENT, or transactions. ▪ The following are features and functionality are required for this application feature: <ul style="list-style-type: none"> ○ POS Vehicle Reservation, Rental & Return Management Feature: POS Customer Vehicle Reservation, Rental & Return Management (Retail Customer & Corporate Customer) features such as <i>Vehicle Reservations, Vehicle Rental & Vehicle Return functionalities:</i> <ul style="list-style-type: none"> - Feature UI Form Requirements: <i>Design & programming</i> of required User-Interface Forms & GUI Controls to support this feature. - Feature Processing Requirements: <i>Design & programming</i> of required Object-Oriented (OOP) Processing & Logic to support this feature. ▪ 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	<p>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:</p> <ul style="list-style-type: none"> ▪ 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	<p>FEATURE #3A – EZRental Internal Back-Office Agency BACK-OFFICE VEHICLE INVENTORY MANAGEMENT SYSTEM (NOT A CUSTOMER FACING APPLICATION):</p> <ul style="list-style-type: none"> ▪ 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 <i>Rental Agencies</i> to service inventory needs for VEHICLE INVENTORY MANAGEMENT, or transactions. ▪ 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. ▪ The following are features and functionality are required for this application feature: <ul style="list-style-type: none"> ○ 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. <ul style="list-style-type: none"> - Feature UI Form Requirements: <i>Design & programming</i> of required User-Interface Forms & GUI Controls to support this feature. - Feature Processing Requirements: <i>Design & programming</i> of required Object-Oriented (OOP) Processing & Logic to support this feature. ▪ This back-office features is 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	<p>FEATURE #3B – EZRental Rental Agency Vehicle Inventory Management System Back-end Database Design & Implementation to support this feature:</p> <ul style="list-style-type: none"> ▪ 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	<p>FEATURE #4A – EZRental Rental Agency Point-of-Sales (POS) BACK-OFFICE CREDIT CARD MANAGEMENT SYSTEM:</p> <ul style="list-style-type: none"> ▪ WINDOWS CLIENT POINT-OF SALES SYSTEM – <i>WINDOWS UI FORM(S) FRONT-END APPLICATION & OOP PROGRAMMING</i> features is a back-end system used by customer service representative & other employees via the <i>Point-of-Sales computer</i> machine in the <i>Rental Agencies</i> to service customer's CREDIT CARD MANAGEMENT, or transactions required when servicing customers. ▪ The following are features and functionality are required for this application with features such as: <ul style="list-style-type: none"> ○ POS Credit Card Management Feature: <i>POS Customer Credit Card Management features</i> such as <i>Credit Card Search & Print, New Credit Card Registration, Credit Card Update, Credit Card Deletion, & Credit Card Listing functionalities:</i> <ul style="list-style-type: none"> - Feature UI Form Requirements: <i>Design & programming</i> of required <i>User-Interface Forms & GUI Controls</i> to support this feature. - Feature Processing Requirements: <i>Design & programming</i> of required <i>Object-Oriented (OOP) Processing & Logic</i> to support this feature. ▪ 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	<p>FEATURE #4B – EZRental Rental Agency Point-of-Sales (POS) Credit Card Management System Back-end Database Design & Implementation to support this feature:</p> <ul style="list-style-type: none"> ▪ DATABASE SERVER BACK-END SYSTEM – <i>BACK-END DATABASE DESIGN & FEATURES</i> (Create the Tables & Relationships, pre-populated tables, stored procedures, views, indexes etc.,) to support this feature.

Feature #	Feature Description
FEATURE #5A	<p style="text-align: center;">FEATURE #5A</p> <p>FEATURE #5A – EZRental Rental Agency Point-of-Sales (POS) System BACK-OFFICE EMPLOYEE & CUSTOMER USER-ACCOUNT MANAGEMENT SYSTEM:</p> <ul style="list-style-type: none"> ▪ WINDOWS CLIENT POINT-OF SALES SYSTEM – WINDOWS UI FORM(S) BACK-OFFICE APPLICATION & OOP PROGRAMMING User-Accounts Management Features used by <i>customer service representative & IT Administrator employees</i> via the <i>Point-of-Sales computer</i> machine in the <i>Rental Agencies</i> to service customer's CUSTOMER & EMPLOYEE USER ACCOUNT MANAGEMENT requests or transactions. ▪ Employee User Accounts – These are the user accounts used by <u>IT Administrators</u>, <u>Customer Service Employees</u>, <u>back-office employees</u>, and <u>any employee who qualifies</u> for access to the system. ▪ Customer User Accounts – These are the user accounts used by <u>IT Administrators</u>, <u>Customer Service Employees</u>, <u>back-office employees</u> and any <u>employee</u> who has access to the system to <u>manage</u> the Customer User Accounts for login into the Customer Web Portal. ▪ The following are features and functionality that are required for this application feature: <ul style="list-style-type: none"> ○ POS User Account (Employee & Customer) Management Feature: POS User Account Management (Employee & Customer) features such as: <ul style="list-style-type: none"> - Employee User Account Feature 5A-1 – Allows <u>IT Administrators</u>, <u>Customer Service Employees</u>, <u>back-office employees</u>, and <u>any employee who qualifies</u> to <u>manage</u> employee user accounts that allow employees to login into the POS System. And perform the following tasks: <i>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</i> functionalities. IMPORTANT! Note that the <u>password</u> is <u>NEVER DISPLAYED or LISTED</u>, only the <u>username</u>! - Customer User Account Feature 5A-2 – Allows <u>IT Administrators</u>, <u>Customer Service Employees</u>, <u>back-office employees</u>, and <u>any employee who qualifies</u> to <u>manage</u> customer user accounts that allow customers to login into the Customer Web Portal System. And perform the following tasks: <i>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</i> functionalities. IMPORTANT! Note that the <u>password</u> is <u>NEVER DISPLAYED or LISTED</u>, only the <u>username</u>! ○ Note that each transaction is saved to database immediately after execution: <ul style="list-style-type: none"> - Feature UI Form Requirements: <i>Design & programming</i> of required <u>User-Interface Forms & GUI Controls</u> to support this feature. - Feature Processing Requirements: <i>Design & programming</i> of required <u>Object-Oriented (OOP) Processing & Logic</u> to support this feature. ▪ This feature is designed only to be used by <u>IT Administrations</u> 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 <i>Rental Agencies</i>.

FEATURE #5B	FEATURE #5B
	<p>FEATURE #5B – EZRental Rental Agency Point-of-Sales (POS) User EMPLOYEE User Account Management System Back-end Database Design & Implementation to support this feature:</p> <ul style="list-style-type: none"> ▪ 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
	<p>FEATURE #5C – EZRental Rental Agency Point-of-Sales (POS) User CUSTOMER User Account Management System Back-end Database Design & Implementation to support this feature:</p> <ul style="list-style-type: none"> ▪ 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	<h2 data-bbox="409 308 698 344">FEATURE #6A</h2> <p data-bbox="409 382 1405 413">FEATURE #6A – EZRental Rental Agency Point-of-Sales (POS) System</p> <p data-bbox="409 418 1481 449">EMPLOYEES BACK-OFFICE SECURITY LOGIN AUTHENTICATION SYSTEM:</p> <ul style="list-style-type: none"> <li data-bbox="409 481 1519 614">▪ Proper <i>security and authentication</i> must be implemented to make sure only authorized employees can access the Point-Of- Sales & Back-End Management systems when they login into the Windows Two-Tiered Client/Server Application & Web Three-Tiered Corporate Client/Server Application. <li data-bbox="409 618 1519 789">▪ 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 <i>Point-of-Sales computer machine</i> in the <i>Rental Agencies</i> to service employee LOGIN AUTHENTICATION SYSTEM. <li data-bbox="409 794 1388 825">▪ The following are features and functionality that are required for this application such as: <ul style="list-style-type: none"> <li data-bbox="458 857 1519 1148">○ POS Employee Back-Office Security Login Authentication System: POS Login Authentication Access for Employees features such as: <ul style="list-style-type: none"> <li data-bbox="507 920 1519 1051">- 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 <u>designed</u> & <u>programmed</u> into the application. <li data-bbox="507 1053 1454 1148">- Employee Authentication Feature 6A-2 – Design & programming of required User-Interface Forms & GUI Controls to support the Authentication System feature! <li data-bbox="458 1184 1470 1347">○ Programming includes: <ul style="list-style-type: none"> <li data-bbox="507 1216 1470 1279">- Feature UI Form Requirements: Design & programming of required User-Interface Forms & GUI Controls to support this feature. <li data-bbox="507 1281 1454 1347">- Feature Processing Requirements: Design & programming of required Object-Oriented (OOP) Processing & Logic to support this feature. <li data-bbox="409 1379 1503 1453">▪ 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	<h2 data-bbox="409 1484 698 1520">FEATURE #6B</h2> <p data-bbox="409 1562 1442 1594">FEATURE #6B – EZRental Rental Agency Point-of-Sales (POS) Employee</p> <p data-bbox="409 1598 1426 1630">Back-Office Security Login Authentication System Back-end Database</p> <p data-bbox="409 1634 1122 1666">Design & Implementation to support this feature:</p> <ul style="list-style-type: none"> <li data-bbox="458 1698 1503 1803">▪ 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	<p align="center">FEATURE #7A</p> <p align="center">FEATURE #7A – EZRental INTERNAL CORPORATE EMPLOYEE & RENTAL AGENCIES EMPLOYEES INTRANET CORPORATE BUSINESS APPLICATIONS WEB PORTAL:</p> <ul style="list-style-type: none"> ▪ 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 <i>execute</i> Enterprise Resource Planning Systems (ERP) Applications & other Corporate Applications online intranet via a BROWSER. ▪ 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. ▪ The following are features and functionality that are required for this INTRANET WEB APPLICATION: <ul style="list-style-type: none"> ○ Corporate Business Application Intranet Web Portal Features: <ul style="list-style-type: none"> - Intranet Web Enterprise Resource Planning Systems (ERP) Portal Feature 7A-1 – Provides access to Enterprise Resource Planning Systems (ERP) Applications such as: <i>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.</i> - Web EZRental Point-of-Sales Corporate Management Feature 7A-2 <ul style="list-style-type: none"> – Allows Employees to <i>manage</i> & <i>execute</i> Point-of-Sales (POS) transaction via the Intranet Web Portal such as: <i>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.</i> ○ Programming includes: <ul style="list-style-type: none"> - Feature UI Form Requirements: <i>Design & programming</i> of required INTRANET WEB User-Interface Forms & GUI Controls to support this feature. - Feature Processing Requirements: <i>Design & programming</i> of required INTRANET WEB TECHNOLOGY, Object-Oriented (OOP) Processing & Logic PROCESSING to support this feature. ▪ 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
	<p>FEATURE #7B – EZRental Corporate Employee & Rental Agencies Employees INTRANET WEB PORTAL System Back-end Database Design & Implementation to support this feature:</p> <ul style="list-style-type: none"> ▪ 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	<p>FEATURE #8A</p> <p>FEATURE #8A – EZRental EXTERNAL CUSTOMER SELF-SERVICE INTERNET CUSTOMER FACING POINT-OF-SALES (POS) WEB PORTAL:</p> <ul style="list-style-type: none"> ▪ 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. ▪ 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. ▪ The following are features and functionality that are required for this WEB APPLICATION feature: <ul style="list-style-type: none"> ○ POS Reservation & Management Features: <ul style="list-style-type: none"> - 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-Of-Sales portal and login and out of their profile website. - Web POS Customer Self-Service Management Feature 8A-2 – Allows POS Customer (Retail Customer & Corporate Customer) Self-Service Management of their account such as: <i>Customer Profile Information, Customer Account & Login Registration, Customer Update Profile, Customer Delete Profile, & Customer Listing functionalities such as listing of Reservations & Rental History etc.</i> - 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: <i>Make Reservations of a Vehicle, Manage an existing Rental, etc.</i> - 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: <i>Reset Username & Reset Password.</i> ○ Programming includes: <ul style="list-style-type: none"> - Feature UI Form Requirements: <i>Design & programming</i> of required WEB User-Interface Forms & GUI Controls to support this feature. - Feature Processing Requirements: <i>Design & programming</i> of required WEB TECHNOLOGY, Object-Oriented (OOP) Processing & Logic PROCESSING to support this feature. ▪ 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**

FEATURE #8B – EZRental Customer Self-Service internet Customer facing Point-of-Sales (POS) WEB PORTAL System Back-end Database Design & Implementation to support this feature:

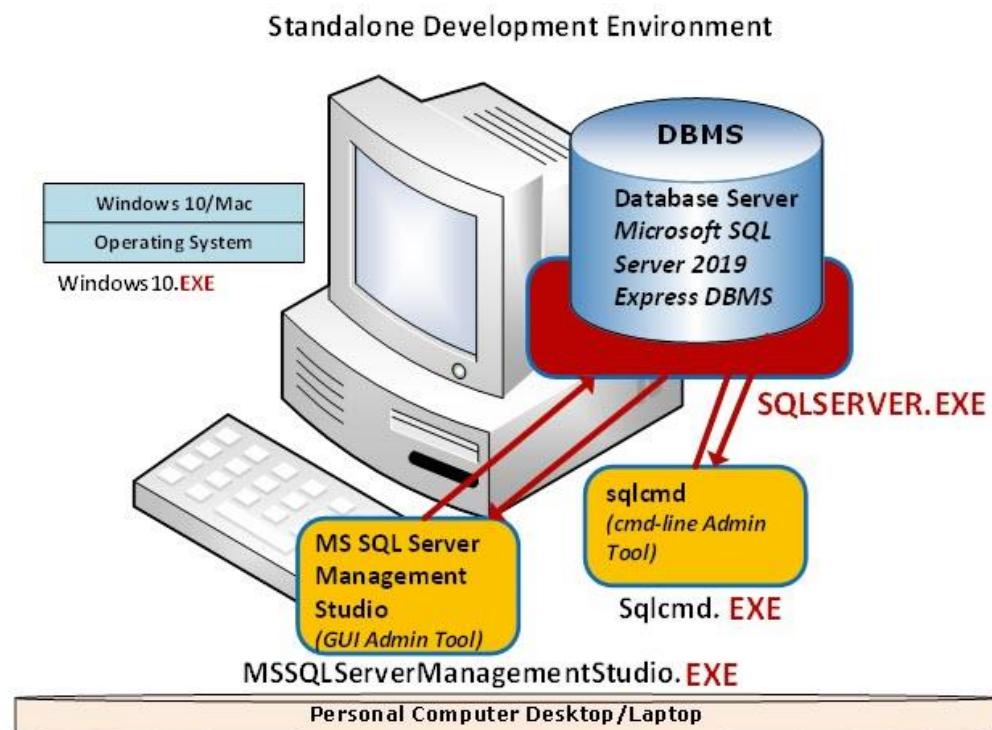
- **DATABASE SERVER BACK-END SYSTEM – BACK-END DATABASE DESIGN & FEATURES** (Create the Tables & Relationships, pre-populated tables, stored procedures, views, indexes etc.,) to support this feature.

Feature #	Feature Description
FEATURE #9A	<p>FEATURE #9A</p> <p>FEATURE #9A – EZRental Customer Point-of-Sales (POS) System CUSTOMER FACING WEB PORTAL SECURITY LOGIN AUTHENTICATION SYSTEM:</p> <ul style="list-style-type: none"> ▪ Proper <i>security and authentication</i> 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. ▪ 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. ▪ The following are features and functionality that are required for this application such as: <ul style="list-style-type: none"> ○ Customer Self-Service Web Portal Security Login Authentication System. Self-Service Web Portal Login Authentication Access for Customer features such as: <ul style="list-style-type: none"> - 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 <u>designed</u> & <u>programmed</u> into the application. - Customer Authentication Feature 9A-2 – Design & programming of required Web User-Interface Forms & GUI Controls to support the Web Portal Authentication System feature! ○ Programming includes: <ul style="list-style-type: none"> - Feature Web UI Form Requirements: Design & programming of required <i>User-Interface Forms & GUI Controls</i> to support this feature. - Feature Web Processing Requirements: Design & programming of required <i>Object-Oriented (OOP) Processing & Logic</i> to support this feature. ▪ 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	<p>FEATURE #9B</p> <p>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:</p> <ul style="list-style-type: none"> ▪ 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	<h2 data-bbox="421 297 719 333">FEATURE #10A</h2> <p data-bbox="421 369 1346 466">FEATURE #10A – EZRental INTERNAL CORPORATE EMPLOYEE & RENTAL AGENCIES EMPLOYEES INTRANET BACK-OFFICE VEHICLE TRANSPORT MANAGEMENT SYSTEM WEB PORTAL:</p> <ul style="list-style-type: none"> <li data-bbox="421 502 1437 663">▪ This INTRANET Web Portal EZRentalHub.com, is a Web-based Three-Tiered Client/Server physical & Software Development Architecture used by CORPORATE & AGENCY EMPLOYEES to <i>manage</i> Transportation of Vehicles by Employee Drivers to and from Rental Agencies, Vehicle Distribution Centers, and other Locations via a BROWSER. <li data-bbox="421 671 1383 832">▪ WEB TRANSPORT 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. <li data-bbox="421 846 1437 903">▪ The following are features and functionality that are required for this INTRANET Transport Management WEB APPLICATION: <ul style="list-style-type: none"> <li data-bbox="470 939 1334 996">○ Corporate Vehicle Transport Application Intranet Web Portal Features: <ul style="list-style-type: none"> <li data-bbox="518 1032 1449 1110">- Transport Scheduling Feature – handle the day-to-day creating & scheduling of a pic-up & delivery (Any vehicle type) such as: <i>Creation of NEW Vehicle Transport Request, Vehicle Pick-up, Vehicle Drop-off & Vehicle Transport Status etc.</i> <li data-bbox="470 1142 752 1170">○ Programming includes: <ul style="list-style-type: none"> <li data-bbox="518 1205 1400 1262">- Feature UI Form Requirements: <i>Design & programming</i> of required INTRANET WEB User-Interface Forms & GUI Controls to support this feature. <li data-bbox="518 1271 1449 1370">- Feature Processing Requirements: <i>Design & programming</i> of required INTRANET WEB TECHNOLOGY, Object-Oriented (OOP) Processing & Logic PROCESSING to support this feature. <li data-bbox="421 1406 1432 1495">▪ 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	<h2 data-bbox="421 1569 719 1605">FEATURE #10B</h2> <p data-bbox="421 1641 1346 1738">FEATURE #10B – EZRental Corporate Vehicle Transport Application Intranet Web Portal Features System Back-end Database Design & Implementation to support this feature:</p> <ul style="list-style-type: none"> <li data-bbox="470 1774 1383 1852">▪ 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 Management System (DBMS) in scope is MS SQL Server Community Edition since this is the standard DBMS used at EZRental Inc. The objectives are to install MS SQL Server Community Edition which includes SQLCMD command-line admin tool on your personal computer along with MS SQL Server Management Studio graphical admin tool to create the following database development Environment.



Project Roles & Responsibilities

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

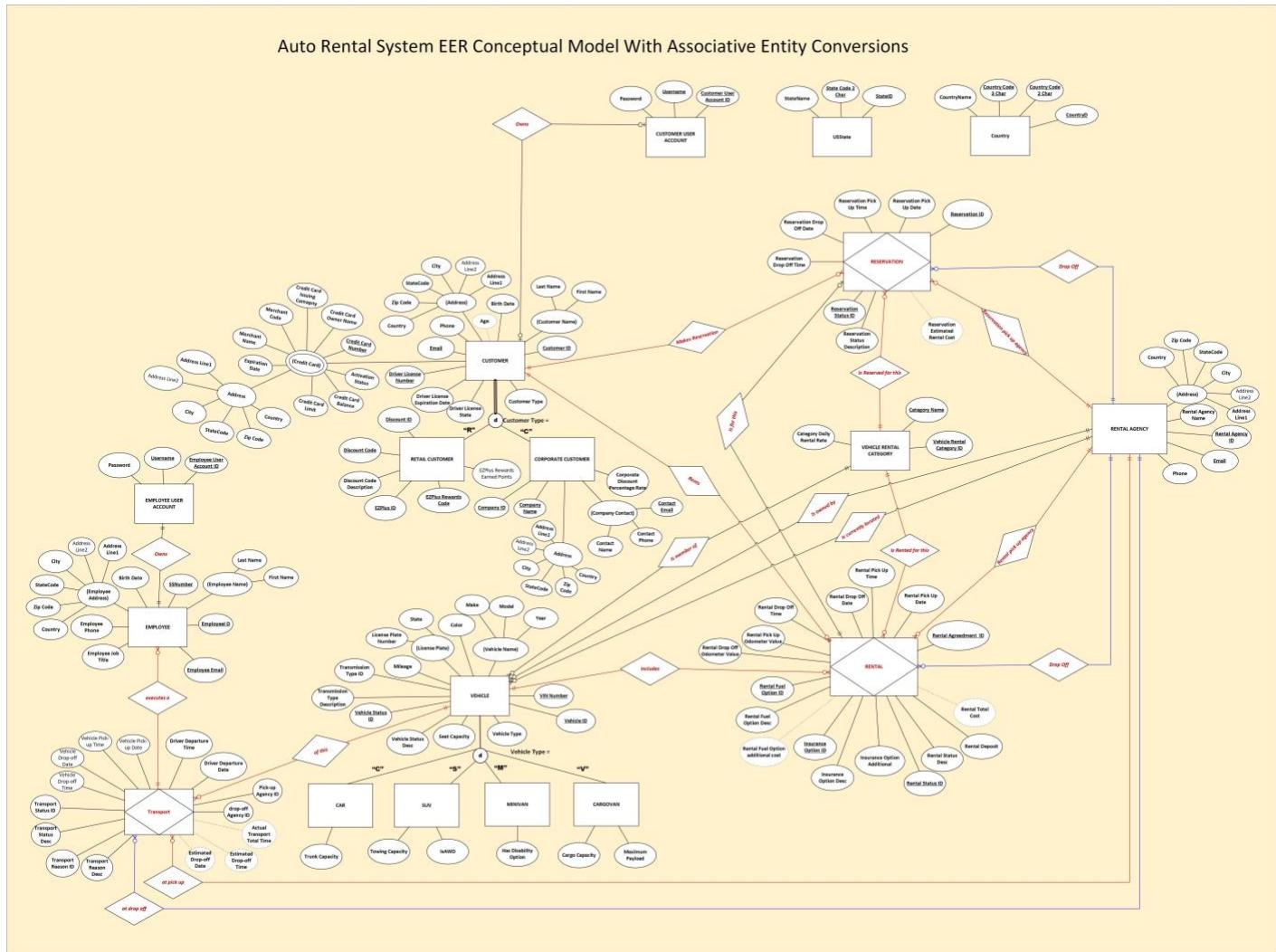
Person	Role	Description
Prof. Rodriguez	Program Manager, AgileScrum Master & ProjectManager	<ul style="list-style-type: none"> ▪ Owner of the project and liaison to Manage the EZRental Inc., the customer. ▪ Activities include but not limited to: <ol style="list-style-type: none"> 1. Owner of project responsible for the success of the project. 2. Project Management 3. Scrum Master ensures the project stays on time and moving in the right direction. Clear any obstacles impeding the team's progress etc.
Consultant #1: Prof. Rodriguez	Business & DatabaseAnalyst	<ul style="list-style-type: none"> ▪ 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 include but not limited to: <ol style="list-style-type: none"> 1. Engage in discovery activities & interview the stakeholders at EZRental Inc. 2. From the interview and discovery create 1) ER/EER Conceptual Data Model from the business requirements & 2) Normalized Logical Model.
Consultant #2, 3, 4 & 5	Database Developers	<ul style="list-style-type: none"> ▪ This role uses the Normalized Logical Model created by consultant #2 to create the Data Dictionary, Physical Schema Diagram, and Implement the Database Application for the Auto Rental System. ▪ Activities include but not limited to: <ol style="list-style-type: none"> 1. Use the Normalized Logical Model created by consultant #2 to do the following: <ol style="list-style-type: none"> 1) Create Data Dictionary tables for eachlogical table targeting MS SQL Server 2) Create Physical Schema Diagram. 2. From these two deliverables, <ol style="list-style-type: none"> 1) implement the Database Application using Oracle 18c for the Auto Rental System.
Consultant #6	Database Administrator	<ul style="list-style-type: none"> ▪ The DB Admin, install the DBMS, maintain, and operate the DBMS throughout its lifetime. ▪ Activities include but not limited to: <ol style="list-style-type: none"> 1. As DB Admin, you are to 1) Setup & install MS SQL Server 2) Administrative tools for target DBMS. 2. Also, as DB Admin, you are to 3) Operate & Maintain the DBMS.

- 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	<ul style="list-style-type: none"> ▪ 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: <ol style="list-style-type: none"> 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,	Full Stack Windows Application Developers & UI/UX Client Application Developer	<ul style="list-style-type: none"> ▪ 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 include but not limited to: <ol style="list-style-type: none"> 1. As full stack developer, Programming & implementation of the Object-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 #	Full Stack Web Application Developer & UI/UX Web Application Developer	<ul style="list-style-type: none"> ▪ Object-Oriented-Programming developer to implement the Web Browser Application using C# & ASP.NET technologies. ▪ Activities include but not limited to: <ol style="list-style-type: none"> 1. As full stack developer, Programming & implementation of the Object-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

Database Design Deliverable #2 - ER/ERR Conceptual Model

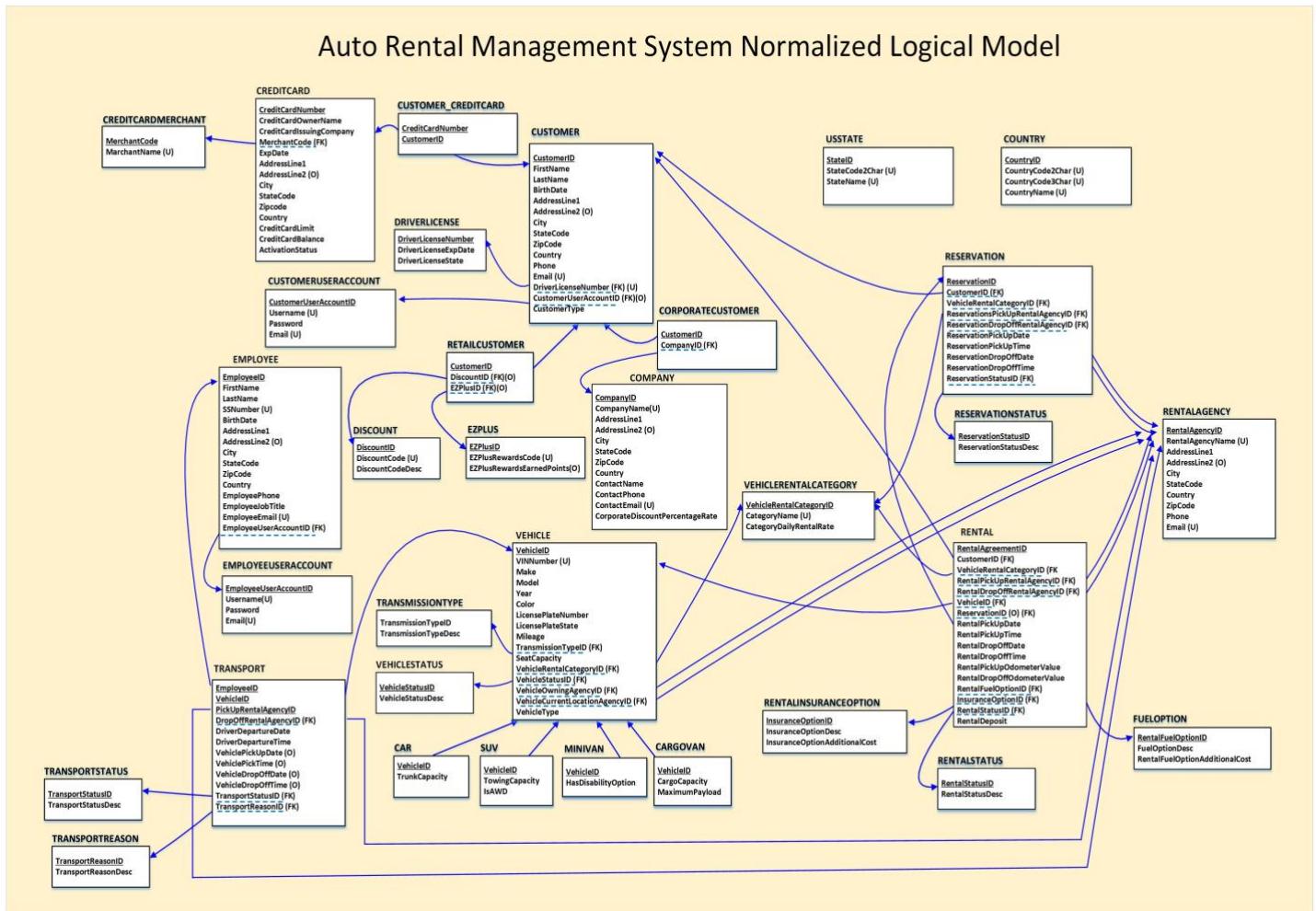
This EER Conceptual Model Diagram is the foundation of the Database Design for the DBMS Auto Rental Management System Application. Its goal is to take the entities and relationships identified in the Application Business Requirements and connect them together to build a DIAGRAM or high-level picture of how the Application key Business data relate to each other.



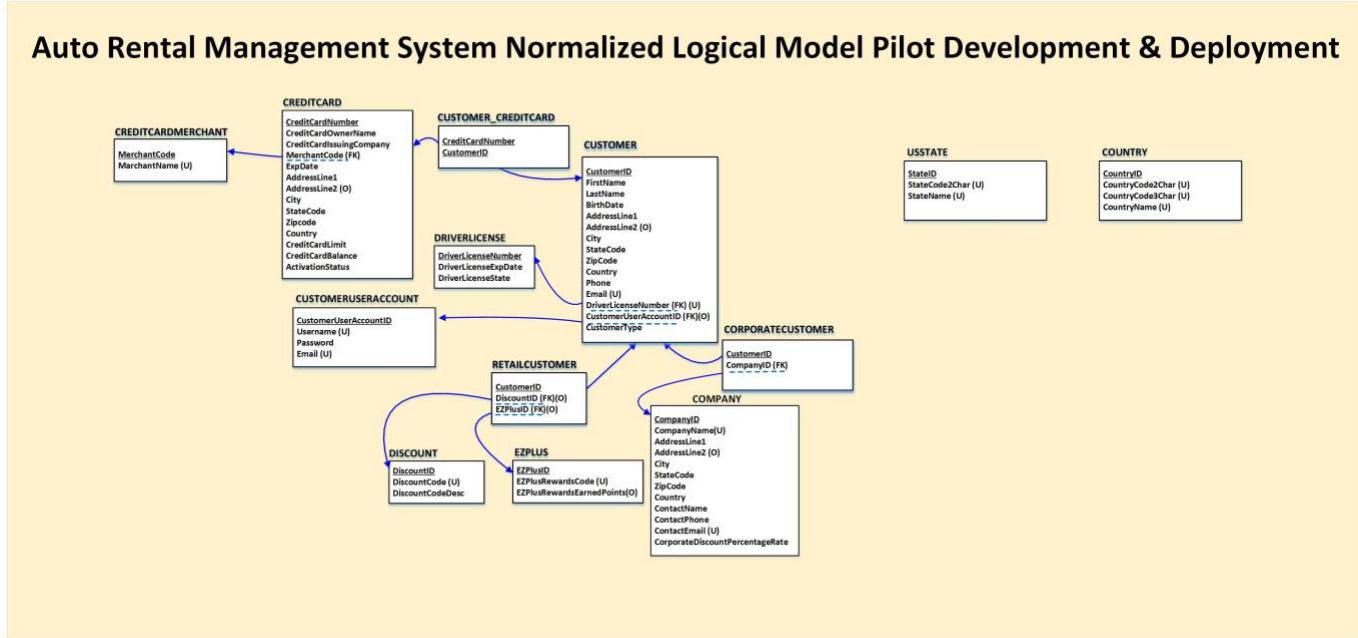
Database Design Deliverable #3 – Normalized Logical Model

The Normalized Logical Model Diagram is derived using the following 2 steps by the Database Analyst/Architect:

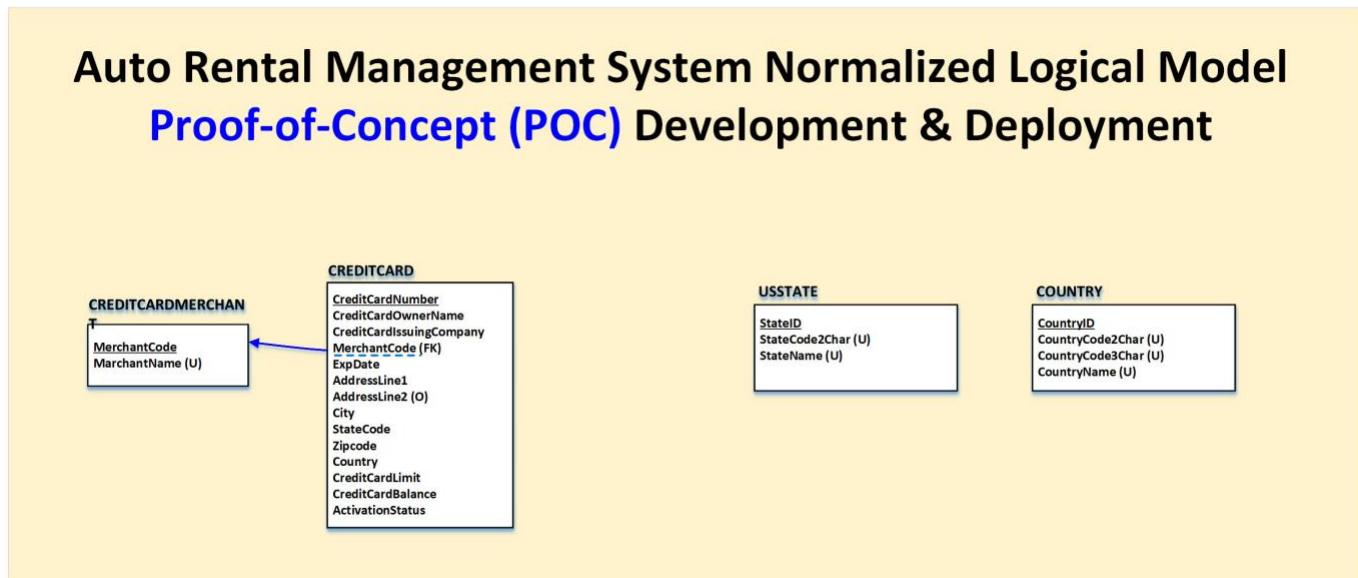
- TRANSFORM the Data Modeling ER/EER Diagram to a Logical Model Diagram which converts the ER/EER Diagram into database tables (If interested in learning more in the future, see CST3504 Database Design Course Lectures to learn the steps for this process.).
- NORMALIZE the Logical Model Diagram using the 3-Normal Forms Rules to create the Normalized Logical Model Diagram (If interested in learning more in the future, see CST3504 Database Design Course Lectures to learn the steps for this process.).



Below is the LIMITED PILOT Normalized Logical Model that will be deployed to a limited number of users throughout the organization as a final production test. This diagram only contains the 13 logical tables in scope of the PILOT:



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. This diagram only contains the 4 logical tables in scope of the POC:



Database Design Deliverable #4 – Physical Model Data Dictionary

The physical Model Data Dictionary is the second component of our Waterfall Methodology DESIGN PHASE. The Physical Model Data Dictionary is a tabular listing of column name, column database type, column constraints, and column description. The purpose is to show the relationship of what DBMS or Physical Database looks like.

CREDITCARDMERCHANT							
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.	<u>MerchantCode</u>	Number	TINYINT	Yes	Default Size	PRIMARY KEY CHECK(between 0 to 20)	Specific code of merchant
2.	<u>MerchantName</u>	String	VARCHAR(50)	Yes	30	NOT NULL UNIQUE	Name of Merchant

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.	<u>CreditCardNumber</u>	Number	VARCHAR(16)	Yes	16	Primary key	Credit card #
2.	<u>CreditCardOwnerName</u>	String	VARCHAR(50)	Yes	50	NOT NULL	Name on the credit card
3.	<u>CreditCardIssuingCompany</u>	String	VARCHAR(50)	Yes	50	NOT NULL	Name of the credit card company
4.	<u>MerchantCode</u>	Number	TINYINT	Yes	Default Size	FOREIGN KEY(1-20) NOT NULL	Specific code of merchant
5.	<u>ExpDate</u>	Date	DATE	Yes	MM/DD/YY	NOT NULL	Credit card expiration date
6.	<u>AddressLine1</u>	String	VARCHAR(50)	Yes	50	NOT NULL	Home address
7.	<u>AddressLine2</u>	String	VARCHAR(50)	No	50	NULL	Home address
8.	<u>City</u>	String	VARCHAR(30)	Yes	30	NOT NULL	Name of city
9.	<u>StateCode</u>	Character	CHAR(2)	Yes	2	NOT NULL	State code
10.	<u>Zipcode</u>	String	VARCHAR(10)	Yes	10	NOT NULL	Zip code
11.	<u>Country</u>	String	VARCHAR(100)	Yes	100	NOT NULL	Country name
12.	<u>CreditCardLimit</u>	Number	DECIMAL(6,2)	Yes	(6,2)	NOT NULL	Credit card limit
13.	<u>CreditCardBalance</u>	Number	DECIMAL(6,2)	Yes	(6,2)	NOT NULL	Credit card balance
14.	<u>ActivationStatus</u>	Boolean	BIT	Yes	Default Size	NOT NULL	Activation Status (T/F)

USSTATE

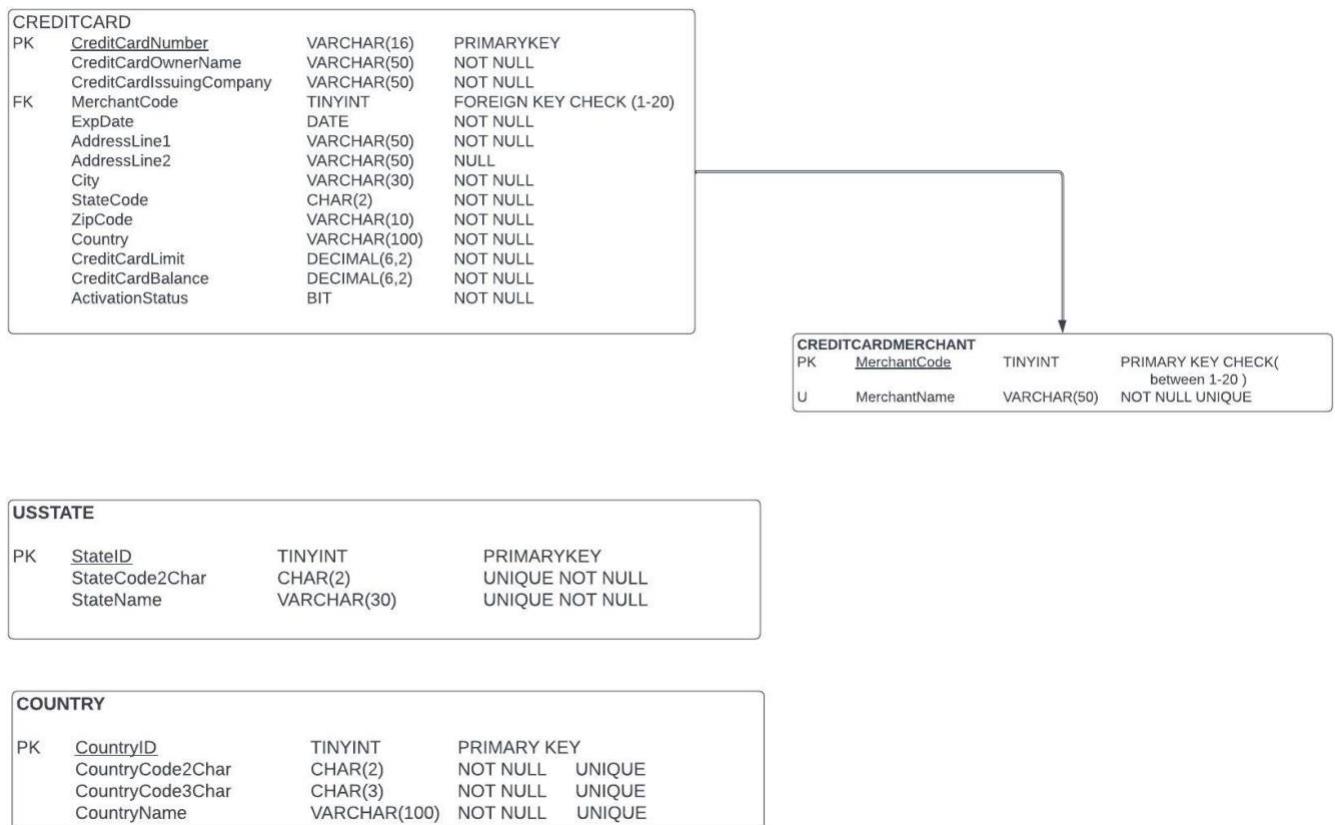
Column Num.	Attribute/Column Name	Generic Data Type Name	MS SQL SERVER Data Type Name	Is it Required?	Length/Size /Format	Constraints	Description/purpose
1.	StateID	Number	TinyInt	Yes	2	PRIMARY KEY CHECK(state ID between 1 and 75)	State ID
2.	StateCode2Char	Character	Char(2)	Yes	2	NOT NULL UNIQUE	State code in character
3.	StateName	String	VARCHAR(30)	Yes	30	NOT NULL UNIQUE	Name of the 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)	Country ID
2.	CountryCode2Char	Character	CHAR (2)	Yes	2	NOT NULL UNIQUE	2 char Country code
3.	CountryCode3Char	Character	CHAR (3)	Yes	3	NOT NULL UNIQUE	3 char Country code
4.	CountryName	String	VARCHAR (100)	Yes	100	NOT NULL UNIQUE	Country Name

Database Design Deliverable #5 – Physical Model Schema Diagram Design

This section is the Physical Model Schema Design Diagram which is the Design of what the physical Database or DBMS tables and their relationships will look like. It is used to implement the database in the DBMS application. This diagram summarizes the tables and relationships required to implement the database in the DBMS application.



Database Implementation Deliverable #6 – Development & Implementation

```
--select database
Use EzRentalDB
--Create CreditCardMerchant

CREATE TABLE CreditCardMerchant
(
    MerchantCode          TINYINT      PRIMARY KEY CHECK(MerchantCode>=1 AND MerchantCode<21) ,
    MerchantName          VARCHAR(50)  UNIQUE NOT NULL
);

--Create CreditCard

CREATE TABLE CreditCard
(
    CreditCardNumber       VARCHAR(16)   PRIMARY KEY,
    CreditCardOwnerName    VARCHAR(50)   NOT NULL,
    CreditCardIssuingCompagy VARCHAR(50)  NOT NULL,
    MerchantCode           TINYINT      NOT NULL,
    ExpDate                DATE         NOT NULL,
    AddressLine1            VARCHAR(50)   NOT NULL,
    AddressLine2            VARCHAR(50)   NULL,
    City                   VARCHAR(30)   NOT NULL,
    StateCode               CHAR(2)      NOT NULL,
    ZipCode                 VARCHAR(10)   NOT NULL,
    Country                VARCHAR(100)  NOT NULL,
    CreditCardLimit         DECIMAL(6,2) NOT NULL,
    CreditCardBalance       DECIMAL(6,2) NOT NULL,
    ActivationStatus        BIT          NOT NULL,
    CONSTRAINT constraintForeingKey
    FOREIGN KEY (MerchantCode)
    REFERENCES CreditCardMerchant(MerchantCode)
    ON DELETE CASCADE ON UPDATE CASCADE
);

--Create a USSTATE TABLE
CREATE TABLE USSTATE
(
    StateID               TINYINT      PRIMARY KEY CHECK(StateID>1 AND StateID<=75 ),
    StateName              VARCHAR(50)   NOT NULL
);
```

```

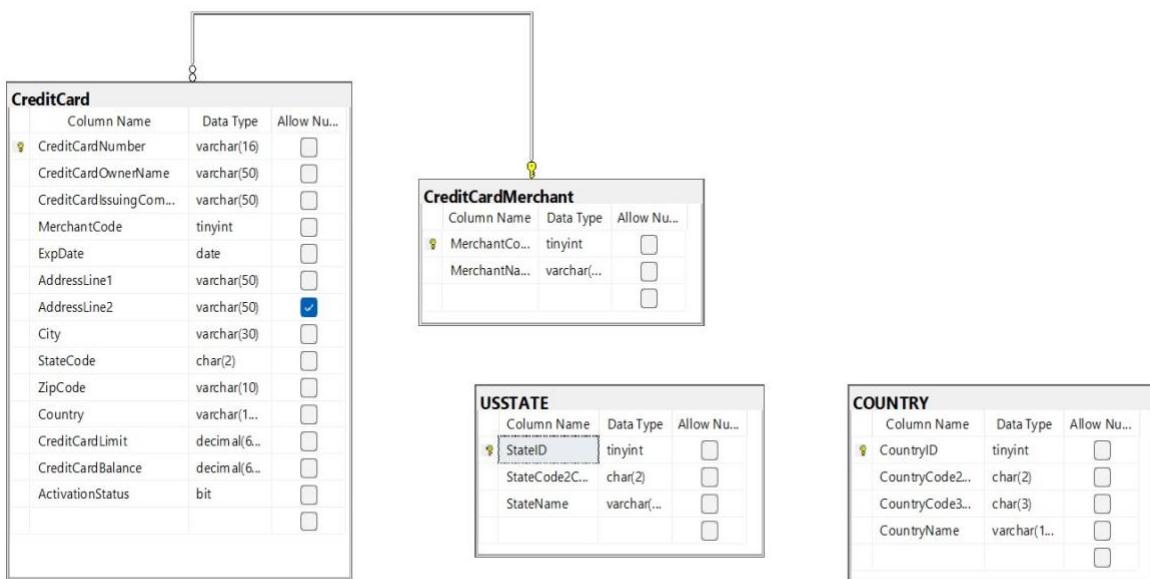
StateCode2Char      CHAR(2)          UNIQUE NOT NULL,
StateName          VARCHAR(30)       UNIQUE NOT NULL
);

--Create a USSTATE TABLE
CREATE TABLE COUNTRY
(
    CountryID        TINYINT         PRIMARY KEY CHECK(CountryID>1 AND CountryID<=250),
    CountryCode2Char CHAR(2)          UNIQUE NOT NULL,
    CountryCode3Char CHAR(3)          UNIQUE NOT NULL,
    CountryName       VARCHAR(100)     UNIQUE NOT NULL
);


```

Database Implementation Deliverable #7 – Implemented Physical Schema Diagram

This sub section describes the seventh database deliverable of the database design and implementation process, which generate the implemented physical schema diagram from the database. This implemented physical diagram is an exported diagram of every table and their relationship that was created using Microsoft SQL server.



Database Implementation Deliverable #8 – Database Validation Testing

Populating The CREDITCARDMERCHANT

This section shows the content of the CreditCardMerchant Table before and after populating it with records using an SQL INSERT STATEMENT

Table Before Executing of SQL INSERT STATEMENTS:

The following SQL SELECT Statement was used to list the content of the CreditCardMerchant Table.

SELECT * FROM CreditCardMerchant

The current state of the CreditCardMerchant Table is:



A screenshot of a SQL query results window. The query is "SELECT * FROM CreditCardMerchant;". The results pane shows a table with two columns: MerchantCode and MerchantName. There are no rows present in the table.

List of SQL INSERT STATEMENTS used to Populate the table:

The following SQL ISNERT STATEMENTS were used to INSERT records to the [CreditCard](#) Table:

```
--Inserting records in the CreditCardMerchant Table

insert into CreditCardMerchant(MerchantCode,MerchantName)
values('1','PayPal');

insert into CreditCardMerchant(MerchantCode,MerchantName)
values('2','Alphabet');
insert into CreditCardMerchant(MerchantCode,MerchantName)
values('4','Zelle');

insert into CreditCardMerchant(MerchantCode,MerchantName)
values('3','CashApp');
insert into CreditCardMerchant(MerchantCode,MerchantName)
values('5','ABC');
```

```
SELECT * FROM CreditCardMerchant
```

The final state of the CreditCardMerchant Table is:

	MerchantCode	MerchantName
1	5	ABC
2	2	Alphabet
3	3	CashApp
4	1	PayPal
5	4	Zelle

Populating The CREDITCARD

This section shows the content of the CreditCard Table before and after populating it with records using an SQL INSERT STATEMENT

Table Before Executing of SQL INSERT STATEMENTS:

The following SQL SELECT Statement was used to list the content of the CreditCardTable.

```
SELECT * FROM CreditCard
```

The current state of the CreditCard Table is:

List of SQL INSERT STATEMENTS used to Populate the table:

The following SQL ISNERT STATEMENTS were used to INSERT records to the [CreditCard](#) Table:

Populating The CREDITCARD Table

```
--Inserting records in the CreditCard Table
```

```
insert into
CreditCard(CreditCardNumber,CreditCardOwnerName,CreditCardIssuingCompagy,MerchantCode,ExpDate,AddressLine1,
```

```

AddressLine2,City,StateCode,ZipCode,Country,CreditCardLimit,CreditCardBalance,ActivationStatus)
values(1235894985613446, 'Jake Paul', 'Clover', 5, '01/09/24', '14 ST', 'Apt 1', 'New York
City', 'NY', '11102', 'USA', 1500.00,100.00,1);

insert into
CreditCard(CreditCardNumber,CreditCardOwnerName,CreditCardIssuingCompagy,MerchantCode,ExpDate,Add
ressLine1,
AddressLine2,City,StateCode,ZipCode,Country,CreditCardLimit,CreditCardBalance,ActivationStatus)
values(2486543594210024, 'Diallo Ibrahima', 'PayPal', 2, '01/12/26', 'Saint Park', 'Apt
8', 'Philadelphia', 'PA', '11002', 'USA', 2500,980.00,1);

insert into
CreditCard(CreditCardNumber,CreditCardOwnerName,CreditCardIssuingCompagy,MerchantCode,ExpDate,Add
ressLine1,
AddressLine2,City,StateCode,ZipCode,Country,CreditCardLimit,CreditCardBalance,ActivationStatus)
values(6543259786214586, 'Adam Eve', 'Google', 3, '03/03/25', '123 ST', 'Apt 5', 'Newark',
'NJ', '10102', 'USA', 1990.00,55.00,1);

insert into
CreditCard(CreditCardNumber,CreditCardOwnerName,CreditCardIssuingCompagy,MerchantCode,ExpDate,Add
ressLine1,
AddressLine2,City,StateCode,ZipCode,Country,CreditCardLimit,CreditCardBalance,ActivationStatus)
values(9865325698745126, 'Lionel Messi', 'Amazon', 1, '05/29/24', '49 ST', 'Apt
1b', 'Calgary', 'AB', 'T6A 2P6', 'Canada', 5000.00,785,1);

insert into
CreditCard(CreditCardNumber,CreditCardOwnerName,CreditCardIssuingCompagy,MerchantCode,ExpDate,Add
ressLine1,
AddressLine2,City,StateCode,ZipCode,Country,CreditCardLimit,CreditCardBalance,ActivationStatus)
values(4865236589745126, 'Cristiano Ronaldo', 'Discover', 4, '4/01/26', '838 Malcom ', 'Apt
7', 'Vancouver', 'BC', 'V6Z 2B1', 'Canada', 4500.00,500.00,1);

```

SELECT * FROM CreditCard

The final state of the CreditCard Table is:

	CreditCardNumber	CreditCardOwnerName	CreditCardIssuingCompagy	MerchantCode	ExpDate	AddressLine1	AddressLine2	City	StateCode	ZipCode	Country	CreditCardLimit	CreditCardBalance	ActivationStatus
1	1235894985613446	Jake Paul	Clover	5	2024-01-09	14 ST	Apt 1	NYC	NY	11102	USA	1500.00	100.00	1
2	2486543594210024	Diallo Ibrahima	PayPal	2	2026-01-12	Saint Park	Apt 8	Philadelphia	PA	11002	USA	2500.00	980.00	1
3	4865236589745126	Cristiano Ronaldo	Discover	4	2026-04-01	838 Malcom	Apt 7	Vancouver	BC	10458	Canada	4500.00	500.00	1
4	6543259786214586	Adam Eve	Google	3	2025-03-03	123 ST	Apt 5	EWR	NJ	10102	USA	1990.00	55.00	1
5	9865325698745126	Lionel Messi	Amazon	1	2024-05-29	49 ST	Apt 1b	Calgary	AB	16485	Canada	5000.00	785.00	1

Populating US State

	StateID	StateCode2Char	StateName
1	1	AL	ALABAMA
2	2	AK	ALASKA
3	3	AS	AMERICAN SAMOA
4	4	AZ	ARIZONA
5	5	AR	ARKANSAS
6	6	CA	CALIFORNIA
7	7	CO	COLORADO
8	8	CT	CONNECTICUT
9	9	DE	DELAWARE
10	10	DC	DISTRICT OF COLUMBIA
11	11	FL	FLORIDA
12	12	GA	GEORGIA
13	13	GU	GUAM
14	14	HI	HAWAII
15	15	ID	IDAHO
16	16	IL	ILLINOIS
17	17	IN	INDIANA
18	18	IA	IOWA
19	19	KS	KANSAS
20	20	KY	KENTUCKY
21	21	LA	LOUISIANA
22	22	ME	MAINE
23	23	MP	MARIANA ISLANDS
24	24	MD	MARYLAND
25	25	MA	MASSACHUSETTS
26	26	MI	MICHIGAN
27	27	MN	MINNESOTA
28	28	MS	MISSISSIPPI
29	29	MO	MISSOURI
30	30	MT	MONTANA
31	31	NE	NEBRASKA
32	32	NV	NEVADA
33	33	NH	NEW HAMPSHIRE
34	34	NJ	NEW JERSEY
35	35	NM	NEW MEXICO
36	36	NY	NEW YORK
37	37	NC	NORTH CAROLINA
38	38	ND	NORTH DAKOTA
39	39	OH	OHIO
40	40	OK	OKLAHOMA
41	41	OR	OREGON
42	42	PA	PENNSYLVANIA
43	43	PR	PUERTO RICO
44	44	RI	RHODE ISLAND
45	45	SC	SOUTH CAROLINA
46	46	SD	SOUTH DAKOTA
47	47	TN	TENNESSEE
48	48	TX	TEXAS
49	49	UT	UTAH
50	50	VT	VERMONT
51	51	VI	VIRGIN ISLANDS
52	52	VA	VIRGINIA
53	53	WA	WASHINGTON
54	54	WV	WEST VIRGINIA
55	55	WI	WISCONSIN
56	56	WY	WYOMING

✓ Query executed successfully.

30	30	MT	MONTANA
31	31	NE	NEBRASKA
32	32	NV	NEVADA
33	33	NH	NEW HAMPSHIRE
34	34	NJ	NEW JERSEY
35	35	NM	NEW MEXICO
36	36	NY	NEW YORK
37	37	NC	NORTH CAROLINA
38	38	ND	NORTH DAKOTA
39	39	OH	OHIO
40	40	OK	OKLAHOMA
41	41	OR	OREGON
42	42	PA	PENNSYLVANIA
43	43	PR	PUERTO RICO
44	44	RI	RHODE ISLAND
45	45	SC	SOUTH CAROLINA
46	46	SD	SOUTH DAKOTA
47	47	TN	TENNESSEE
48	48	TX	TEXAS
49	49	UT	UTAH
50	50	VT	VERMONT
51	51	VI	VIRGIN ISLANDS
52	52	VA	VIRGINIA
53	53	WA	WASHINGTON
54	54	WV	WEST VIRGINIA
55	55	WI	WISCONSIN
56	56	WY	WYOMING

Showing prove for US Countries

100 %

Results Messages

	CountryID	CountryCode2Char	CountryCode3Char	CountryName
1	1	AX	ALA	AALAND ISLANDS
2	2	AF	AFG	AFGHANISTAN
3	3	AL	ALB	ALBANIA
4	4	DZ	DZA	ALGERIA
5	5	AS	ASM	AMERICAN SAMOA
6	6	AD	AND	ANDORRA
7	7	AO	AGO	ANGOLA
8	8	AI	AIA	ANGUILLA
9	9	AQ	ATA	ANTARCTICA
10	10	AG	ATG	ANTIGUA AND BARBUDA
11	11	AR	ARG	ARGENTINA
12	12	AM	ARM	ARMENIA
13	13	AW	ABW	ARUBA
14	14	AU	AUS	AUSTRALIA
15	15	AT	AUT	AUSTRIA
16	16	AZ	AZE	AZERBAIJAN
17	17	BS	BHS	BAHAMAS
18	18	BH	BHR	BAHRAIN
19	19	BD	BGD	BANGLADESH
20	20	BB	BRB	BARBADOS
21	21	BY	BLR	BELARUS
22	22	BE	BEL	BELGIUM
23	23	BZ	BLZ	BELIZE
24	24	BJ	BEN	BENIN
25	25	BM	BMU	BERMUDA
26	26	BT	BTN	BHUTAN
27	27	BO	BOL	BOLIVIA
28	28	BA	BIH	BOSNIA AND HERZEG...
29	29	BW	BWA	BOTSWANA
30	30	BV	BVT	BOUVET ISLAND
31	31	BR	BRA	BRAZIL
32	32	IO	IOT	BRITISH INDIAN OCEA...
33	33	BN	BRN	BRUNEI DARUSSALAM
34	34	BG	BGR	BULGARIA
35	35	BF	BFA	BURKINA FASO
36	36	BI	BDI	BURUNDI

SELECT STATEMENT 1

This query will allow us to select a table in our database in the CreditCard table

```
SELECT * FROM CreditCard;
```

After this code is execute, in the console we will see the table of CreditCard.

	CreditCardNumber	CreditCardOwnerName	CreditCardIssuingCompagy	MerchantCode	ExpDate	AddressLine1	AddressLine2	City	StateCode	ZipCode	Country	CreditCardLimit	CreditCardBalance	ActivationStatus
1	1235894985613446	Jake Paul	Clover	5	2024-01-09	14 ST	Apt 1	New York City	NY	11102	USA	1500.00	100.00	1
2	2486543594210024	Diallo Ibrahima	PayPal	2	2026-01-12	Saint Park	Apt 8	Philadelphia	PA	11002	USA	2500.00	980.00	1
3	4865236589745126	Cristiano Ronaldo	Discover	4	2026-04-01	838 Malcom	Apt 7	Vancouver	BC	10458	Canada	4500.00	500.00	1
4	6543259786214586	Adam Eve	Google	3	2025-03-03	123 ST	Apt 5	Newark	NJ	10102	USA	1990.00	55.00	1
5	9865325698745126	Lionel Messi	Amazon	1	2024-05-29	49 ST	Apt 1b	Calgary	AB	16485	Canada	5000.00	785.00	1

This select statement is to select a record in one table part of a binary table relationship (creditcard table), based on the primary key

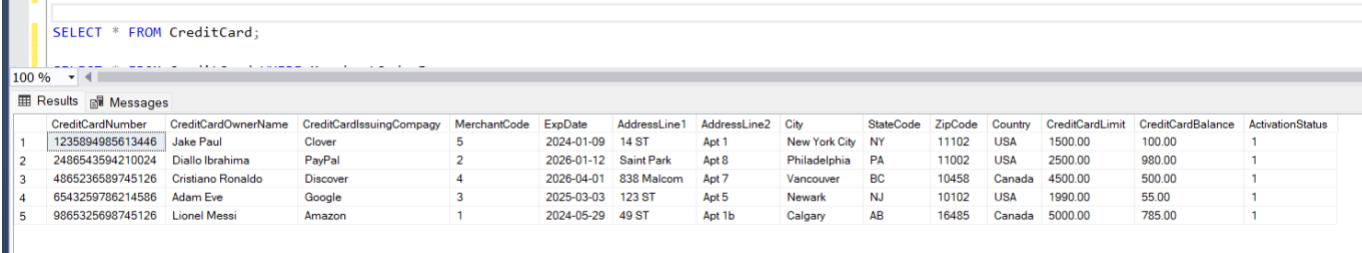
```
SELECT * FROM CreditCard WHERE MerchantCode=5;
```

	CreditCardNumber	CreditCardOwnerName	CreditCardIssuingCompagy	MerchantCode	ExpDate	AddressLine1	AddressLine2	City	StateCode	ZipCode	Country	CreditCardLimit	CreditCardBalance	ActivationStatus
1	1235894985613446	Jake Paul	Clover	5	2024-01-09	14 ST	Apt 1	New York City	NY	11102	USA	1500.00	100.00	1

SELECT STATEMENT 2

This select statement will return all filter rows from the table based on our specific criteria.

Before executing the select statement, this is the actual records on the table



The screenshot shows the SQL Server Management Studio interface. In the top-left pane, there is a code editor with the following SQL query:

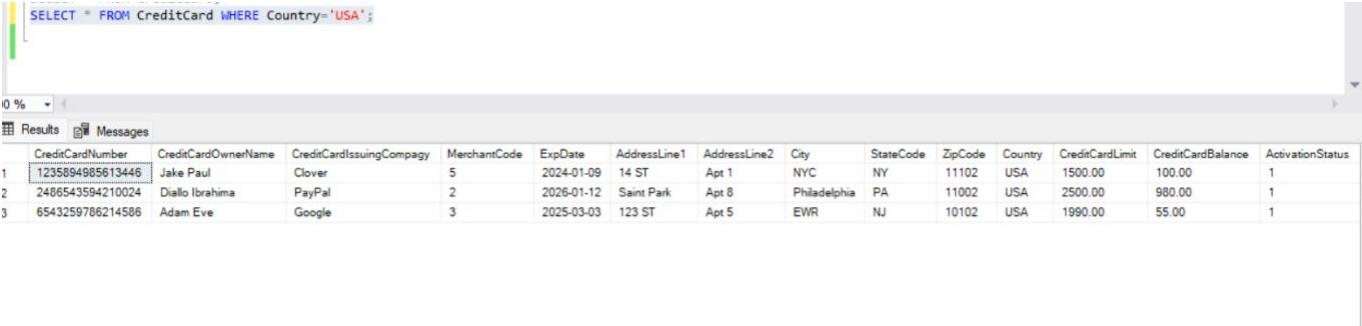
```
SELECT * FROM CreditCard;
```

In the bottom-right pane, there is a results grid titled "Results". The grid displays 5 rows of data from the CreditCard table. The columns are: CreditCardNumber, CreditCardOwnerName, CreditCardIssuingCompany, MerchantCode, ExpDate, AddressLine1, AddressLine2, City, StateCode, ZipCode, Country, CreditCardLimit, CreditCardBalance, and ActivationStatus. The data is as follows:

	CreditCardNumber	CreditCardOwnerName	CreditCardIssuingCompany	MerchantCode	ExpDate	AddressLine1	AddressLine2	City	StateCode	ZipCode	Country	CreditCardLimit	CreditCardBalance	ActivationStatus
1	1235894985613446	Jake Paul	Clover	5	2024-01-09	14 ST	Apt 1	New York City	NY	11102	USA	1500.00	100.00	1
2	2486543594210024	Diallo Ibrahima	PayPal	2	2026-01-12	Saint Park	Apt 8	Philadelphia	PA	11002	USA	2500.00	980.00	1
3	4865236569745126	Cristiano Ronaldo	Discover	4	2026-04-01	838 Malcom	Apt 7	Vancouver	BC	10458	Canada	4500.00	500.00	1
4	6543259786214586	Adam Eve	Google	3	2025-03-03	123 ST	Apt 5	Newark	NJ	10102	USA	1990.00	55.00	1
5	9865325698745126	Lionel Messi	Amazon	1	2024-05-29	49 ST	Apt 1b	Calgary	AB	16485	Canada	5000.00	785.00	1

```
SELECT * FROM CreditCard WHERE Country= 'USA' ;
```

After this code is executed, in the console we will see a table of a CreditCard



The screenshot shows the SQL Server Management Studio interface. In the top-left pane, there is a code editor with the following SQL query:

```
SELECT * FROM CreditCard WHERE Country='USA' ;
```

In the bottom-right pane, there is a results grid titled "Results". The grid displays 3 rows of data from the CreditCard table, filtered by Country='USA'. The columns are: CreditCardNumber, CreditCardOwnerName, CreditCardIssuingCompany, MerchantCode, ExpDate, AddressLine1, AddressLine2, City, StateCode, ZipCode, Country, CreditCardLimit, CreditCardBalance, and ActivationStatus. The data is as follows:

	CreditCardNumber	CreditCardOwnerName	CreditCardIssuingCompany	MerchantCode	ExpDate	AddressLine1	AddressLine2	City	StateCode	ZipCode	Country	CreditCardLimit	CreditCardBalance	ActivationStatus
1	1235894985613446	Jake Paul	Clover	5	2024-01-09	14 ST	Apt 1	NYC	NY	11102	USA	1500.00	100.00	1
2	2486543594210024	Diallo Ibrahima	PayPal	2	2026-01-12	Saint Park	Apt 8	Philadelphia	PA	11002	USA	2500.00	980.00	1
3	6543259786214586	Adam Eve	Google	3	2025-03-03	123 ST	Apt 5	EWR	NJ	10102	USA	1990.00	55.00	1

In my actual creditcard table I have 5 insertions, but because of the where statement I filtered the table so now the results is three rows with country name USA.

SELECT STATEMENT 3

This select statement queries two of the chosen table group that are part of an associate entity that has a binary relationship.

The screenshot shows a SQL query window with the following content:

```
EZRentalDB_Test_S...7DRORP\saran (62)*
SELECT CreditCard.CreditCardOwnerName, CreditCard.CreditCardLimit, CreditCardMerchant.MerchantName, CreditCard.AddressLine1, CreditCard.ExpDate
FROM CreditCard, CreditCardMerchant ;
```

The results pane displays a table with the following data:

	CreditCardOwnerName	CreditCardLimit	MerchantName	AddressLine1	ExpDate
1	Jake Paul	1500.00	ABC	14 ST	2024-01-09
2	Diallo Ibrahima	2500.00	ABC	Saint Park	2026-01-12
3	Sergio Ramos	5500.00	ABC	87 Columbus	2029-08-06
4	Cristiano Ronaldo	4500.00	ABC	838 Malcom	2026-04-01
5	Adam Eve	1990.00	ABC	123 ST	2025-03-03
6	Lionel Messi	5000.00	ABC	49 ST	2024-05-29
7	Jake Paul	1500.00	Alphabet	14 ST	2024-01-09
8	Diallo Ibrahima	2500.00	Alphabet	Saint Park	2026-01-12
9	Sergio Ramos	5500.00	Alphabet	87 Columbus	2029-08-06
10	Cristiano Ronaldo	4500.00	Alphabet	838 Malcom	2026-04-01
11	Adam Eve	1990.00	Alphabet	123 ST	2025-03-03
12	Lionel Messi	5000.00	Alphabet	49 ST	2024-05-29
13	Jake Paul	1500.00	Alphabet	14 ST	2024-01-09
14	Diallo Ibrahima	2500.00	Alphabet	Saint Park	2026-01-12
15	Sergio Ramos	5500.00	Alphabet	87 Columbus	2029-08-06
16	Cristiano Ronaldo	4500.00	Alphabet	838 Malcom	2026-04-01
17	Adam Eve	1990.00	Alphabet	123 ST	2025-03-03
18	Lionel Messi	5000.00	Alphabet	49 ST	2024-05-29
19	Jake Paul	1500.00	CashApp	14 ST	2024-01-09
20	Diallo Ibrahima	2500.00	CashApp	Saint Park	2026-01-12
21	Sergio Ramos	5500.00	CashApp	87 Columbus	2029-08-06
22	Cristiano Ronaldo	4500.00	CashApp	838 Malcom	2026-04-01
23	Adam Eve	1990.00	CashApp	123 ST	2025-03-03
24	Lionel Messi	5000.00	CashApp	49 ST	2024-05-29
25	Jake Paul	1500.00	OrangeMoney	14 ST	2024-01-09
26	Diallo Ibrahima	2500.00	OrangeMoney	Saint Park	2026-01-12
27	Sergio Ramos	5500.00	OrangeMoney	87 Columbus	2029-08-06
28	Cristiano Ronaldo	4500.00	OrangeMoney	838 Malcom	2026-04-01
29	Adam Eve	1990.00	OrangeMoney	123 ST	2025-03-03
30	Lionel Messi	5000.00	OrangeMoney	49 ST	2024-05-29
31	Jake Paul	1500.00	Zelle	14 ST	2024-01-09
32	Diallo Ibrahima	2500.00	Zelle	Saint Park	2026-01-12

This statement combine multiple table together. It joins two distinct table together regarding their requirement.

```
SELECT CreditCard.CreditCardOwnerName, CreditCard.CreditCardLimit,
CreditCardMerchant.MerchantName, CreditCard.AddressLine1, CreditCard.ExpDate
FROM CreditCard, CreditCardMerchant WHERE CreditCardMerchant.MerchantCode=CreditCard.MerchantCode
AND Country='USA' ;
```

The screenshot shows a SQL query window with the following content:

```
SELECT CreditCard.CreditCardOwnerName, CreditCard.CreditCardLimit, CreditCardMerchant.MerchantName, CreditCard.AddressLine1, CreditCard.ExpDate
FROM CreditCard, CreditCardMerchant WHERE CreditCardMerchant.MerchantCode=CreditCard.MerchantCode AND Country='USA' ;
```

The results pane displays a table with the following data:

	CreditCardOwnerName	CreditCardLimit	MerchantName	AddressLine1	ExpDate
1	Jake Paul	1500.00	ABC	14 ST	2024-01-09
2	Diallo Ibrahima	2500.00	Alphabet	Saint Park	2026-01-12
3	Sergio Ramos	5500.00	Alphabet	87 Columbus	2029-08-06
4	Adam Eve	1990.00	CashApp	123 ST	2025-03-03

UPDATE STATEMENT #1

This is my current table where the merchantcode is 1 before the update statement.

```
SELECT * FROM CreditCard WHERE MerchantCode=1;
```

CreditCardNumber	CreditCardOwnerName	CreditCardIssuingCompagy	MerchantCode	ExpDate	AddressLine1	AddressLine2	City	StateCode	ZipCode	Country	CreditCardLimit	CreditCardBalance	ActivationStatus
9865325698745126	Lionel Messi	Amazon	1	2024-05-29	49 ST	Apt 1b	Calgary	AB	16485	Canada	5000.00	785.00	1

```
UPDATE CreditCard
    SET CreditCardOwnerName='Thiago Messi',
        CreditCardbalance=2000.00,
        CreditCardIssuingCompagy='Barclays',
        CreditCardNumber=7845798784514623,
        ExpDate='11/18/27',
        AddressLine1='45 Malcom x',
        City= 'New York City',
        StateCode='NY',
        Country='USA',
        ZipCode=20045,
        ActivationStatus=0
    WHERE MerchantCode=1
```

This is the same table after the update statement

CreditCardNumber	CreditCardOwnerName	CreditCardIssuingCompagy	MerchantCode	ExpDate	AddressLine1	AddressLine2	City	StateCode	ZipCode	Country	CreditCardLimit	CreditCardBalance	ActivationStatus
7845798784514623	Thiago Messi	Barclays	1	2027-11-18	45 Malcom x	Apt 1b	New York City	NY	20045	USA	5000.00	2000.00	0

UPDATE STATEMENT #2

This update is to update one column only of one record of the creditcard table. This column to update is the merchantcode column. We have to update an existing merchantcode column with a new merchantcode that is not in the creditcardmerchant table

This is my current table where my filter is based on the primary key.

```
SELECT * FROM CreditCard WHERE MerchantCode=5;
```

Results														
Messages														
	CreditCardNumber	CreditCardOwnerName	CreditCardIssuingCompagy	MerchantCode	ExpDate	AddressLine1	AddressLine2	City	StateCode	ZipCode	Country	CreditCardLimit	CreditCardBalance	ActivationStatus
1	1235894985613446	Jake Paul	Clover	5	2024-01-09	14 ST	Apt 1	New York City	NY	11102	USA	1500.00	100.00	1

This statement will update the merchant code, and the filter is based on a primary key

```
UPDATE CreditCard  
    SET MerchantCode= 8  
    WHERE MerchantCode=5;
```

This is the same table after the update statement is executed

Results														
Messages														
	CreditCardNumber	CreditCardOwnerName	CreditCardIssuingCompagy	MerchantCode	ExpDate	AddressLine1	AddressLine2	City	StateCode	ZipCode	Country	CreditCardLimit	CreditCardBalance	ActivationStatus
1	1235894985613446	Jake Paul	Clover	8	2024-01-09	14 ST	Apt 1	New York City	NY	11102	USA	1500.00	100.00	1

As we can see now the merchantCode is updated.

DELETE STATEMENT 1

Before we execute the delete statement, this is the state of my current creditCard table with multiple records.

The screenshot shows a SQL query results window with the following details:

SQL Query:

```
SELECT * FROM CreditCard;
```

Results Grid:

	CreditCardNumber	CreditCardOwnerName	CreditCardIssuingCompagy	MerchantCode	ExpDate	AddressLine1	AddressLine2	City	StateCode	ZipCode	Country	CreditCardLimit	CreditCardBalance	ActivationStatus
1	1235894985613446	Jake Paul	Clover	8	2024-01-09	14 ST	Apt 1	New York City	NY	11102	USA	1500.00	100.00	1
2	2486543594210024	Diallo Ibrahima	PayPal	2	2026-01-12	Saint Park	Apt 8	Philadelphia	PA	11002	USA	2500.00	980.00	1
3	4579546889542345	Sergio Ramos	Discover	7	2029-08-06	87 Columbus	Apt 9	New York City	NY	10056	USA	5500.00	1000.00	1
4	4865236589745126	Cristiano Ronaldo	Discover	4	2026-04-01	838 Malcom	Apt 7	Vancouver	BC	10458	Canada	4500.00	500.00	1
5	6543259786214586	Adam Eve	Google	3	2025-03-03	123 ST	Apt 5	Newark	NJ	10102	USA	1990.00	55.00	1
6	7845798784514623	Thiago Messi	Barclays	1	2027-11-18	45 Malcom x	Apt 1b	New York City	NY	20045	USA	5000.00	2000.00	0

This delete statement is to delete a record in the creditCard table based on a specific criteria (Primary key) which is to delete the record with a merchantCode 3.

```
DELETE
FROM CreditCard
WHERE MerchantCode=3;
```

The screenshot shows a SQL query results window with the following details:

SQL Query:

```
SELECT * FROM CreditCard;
```

Execution Statement:

```
DELETE
FROM CreditCard
WHERE MerchantCode=3;
```

Resulting Query:

```
SELECT * FROM CreditCard;
```

Results Grid:

	CreditCardNumber	CreditCardOwnerName	CreditCardIssuingCompagy	MerchantCode	ExpDate	AddressLine1	AddressLine2	City	StateCode	ZipCode	Country	CreditCardLimit	CreditCardBalance	ActivationStatus
1	1235894985613446	Jake Paul	Clover	8	2024-01-09	14 ST	Apt 1	New York City	NY	11102	USA	1500.00	100.00	1
2	2486543594210024	Diallo Ibrahima	PayPal	2	2026-01-12	Saint Park	Apt 8	Philadelphia	PA	11002	USA	2500.00	980.00	1
3	4579546889542345	Sergio Ramos	Discover	7	2029-08-06	87 Columbus	Apt 9	New York City	NY	10056	USA	5500.00	1000.00	1
4	4865236589745126	Cristiano Ronaldo	Discover	4	2026-04-01	838 Malcom	Apt 7	Vancouver	BC	10458	Canada	4500.00	500.00	1
5	7845798784514623	Thiago Messi	Barclays	1	2027-11-18	45 Malcom x	Apt 1b	New York City	NY	20045	USA	5000.00	2000.00	0

DELETE STATEMENT 2

Before we execute the delete statement, this is the state of my current creditCardMerchant table with multiple records.

The screenshot shows the SQL Server Management Studio interface. In the top pane, there is a code editor with the following SQL query:

```
SELECT * FROM CreditCardMerchant;
```

In the bottom pane, there are two tabs: "Results" and "Messages". The "Results" tab is selected and displays a table with the following data:

	MerchantCode	MerchantName
1	5	ABC
2	7	Alphabel
3	2	Alphabet
4	8	Block
5	3	CashApp
6	1	OrangeMoney
7	4	Zelle

This delete statement is to delete a record in the creditCardMerchant table based on a specific criterion (Primary key) which is to delete the record with a merchantCode 7.

```
DELETE
FROM CreditCardMerchant
WHERE MerchantCode=7;
```

The screenshot shows the SQL Server Management Studio interface. In the top pane, there is a code editor with the following SQL queries:

```
SELECT * FROM CreditCardMerchant;

DELETE
FROM CreditCardMerchant
WHERE MerchantCode=7;

SELECT * FROM CreditCardMerchant;
```

In the bottom pane, there are two tabs: "Results" and "Messages". The "Results" tab is selected and displays a table with the following data:

	MerchantCode	MerchantName
1	5	ABC
2	2	Alphabet
3	8	Block
4	3	CashApp
5	1	OrangeMoney
6	4	Zelle

Conclusion

This project was done by a team of seven-teen people in our headquarter which is located in NY. The project was divided into four different sprints so that our work can be faster and more effective. We have met our customers' condition as we were asked to do and the results were far better then what we expected. Our team tested the database and checked if it's running correctly. We deleted and also added new data alongside updating it so that our client will be able to easily use this application without any error. We also implemented the waterfall methodology with Agile for far more effective results because waterfall alone isn't the best as you can see.