3/5/2023

EZRENTAL POS

AUTO RENTAL MANAGEMENT SYSTEM

Yomi Olamide
CST 4708

Table of Contents

Executive Summary ***2***

Problem Statement & Objectives ***3***

Project Management Methodology ***4***

Database Design Deliverable #1a – Application Business requirements ***5***

Database Design Deliverable #1b – Application Development Technical Requirements ***10***

Application Physical Technical Architecture ***20***

Application Development Features and Functionalities (Agile Backlog) ***30***

Database Management System Development Environment & Physical Architecture ***40***

Project Roles & Responsibilities ***41***

Database Design Deliverable #2 – ER/EER Conceptual Model Diagram ***43***

Database Design Deliverable #3 – Normalized Logical Model Diagram ***44***

Database Design Deliverable #4 – Physical Model Data Dictionary ***46***

Database Design Deliverable #5 – Physical Model Schema Design Diagram ***48***

Database Implementation Deliverable #6 – Development & Implementation ***49***

Database Implementation Deliverable #7 – Implemented Physical Schema Diagram ***50***

Database Implementation Deliverable #8 – Database Validation Testing ***51***

Conclusion ***61***

PROJECT 1 – EZRental Auto Rental POS Management System Database Design and Implementation

Executive Summary

EZRental requires the implementation of a Database Management System (DBMS) to securely store and manage all information associated with our business operations. The primary objective of this project is to comprehensively address the diverse requirements of EZRental's business model and construct a fully functional database that accommodates all aspects of our enterprise. This database will facilitate the efficient preservation, maintenance, and presentation of customer, corporate, and operational data necessary for the seamless execution of EZRental's daily activities.

This document outlines the specific business requirements, methodologies employed, models devised, and the resulting database structure that has been established to support and optimize EZRental's operations.

Problem Statement & Objectives

The objective is to implement and design an Auto Rental Point-of-Sales Management System Application named EZRental POS. It will include an e-commerce website, EZrental.com, allowing customers to make and manage reservations from anywhere in the world. The system will offer exceptional user experience and competitive pricing across dozens of major cities in US, Canada, Mexico, United Kingdom, Japan, and Australia.

The ultimate goal is to expand further into new markets in Asia, Africa, and the Mediterranean, providing a unified customer experience across all locations. The cutting-edge system will enable Customer Service Representatives and other frontline workers to serve in-person customers efficiently, while business users in corporate offices will access essential functionalities via our corporate INTRANET Website named EZRentalCorp.com. EZRental Inc. is committed to providing a modern and hassle-free rental experience, backed by data-driven decision-making abilities.

Project Management Methodology

The EZRental POS project will adopt a hybrid project management methodology that incorporates elements of both waterfall and agile approaches.

Here is a visual representation of a waterfall methodology that consists of 6 phases, each with its own set of deliverables. The diagram below shows the sequence and the dependencies of these phases. Additionally, a table is presented to give a high-level overview of the objectives, activities and outcomes of each phase.

WATEFALL PHASE	DESCRIPTION
Phase 1 – Planning & Discovery	▪ Plan the entire project & gather requirements from business. Perform the necessary discovery via interviews, assessments, etc., to properly plan, update the methodology and execute the project.
Phase 2 – Analysis	▪ Deep analysis of the identified business requirements and other discovered data from interview and assessments. This results in the foundation to derive design strategy.
Phase 3 – Design	▪ Design the solution. Organize all the requirements and produce a detailed specification of all data elements, required web forms, reports, displays, and processing or business rules. Design database models, webforms and application connectivity.
Phase 4 – Development & Implementation	▪ Implement the solution. Develop the DBMS Server Application (ER model, logical model, physical model, tables, view, stored procedures etc.).
Phase 5 – Testing and Validation	▪ Test and validate the app to confirm it meets the requirements & works.
Phase 6 – Go Live and Operations	▪ Open the application to all users. Monitor, backup & maintain the system. ▪ Database application component needs to be ready for future upgrade such as PROJECT 2, etc., therefore, have a disaster/recovery plan in place so if your computer crashes you can recover your application.

Agile methodology is a project management approach that focuses on delivering value and meeting customer expectations in a timely manner. It involves frequent feedback and collaboration among the project team and the stakeholders. It also allows for flexibility and changes in the project scope and requirements as needed. The diagram below demonstrates the method of agile methodology.

SCRUM PROCESS

The project integrates a hybrid of agile and waterfall methodologies by applying the concept of sprints to the database development and implementation phases. Each sprint consists of a deliverable that follows the waterfall approach. The figure below provides a clear illustration of the interplay between these two methodologies and the table elaborately each phase in detail.

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

The Waterfall Phases will be divided into four Sprints using the Agile Methodology Process. Each Sprint will last five weeks, as shown in the diagram below:

Database Design Deliverable #1a – Application Business requirements

Mr. Rodriguez hired a Database Business Analyst to collect and document the Business Requirements for the EZRental Inc. application. The Business Requirements are the essential data that the business stakeholders and project team need from the database. They also form the basis of Database Design.

The Business Analyst documented the Business Requirements in the following pages:

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 <u>both</u> 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:

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

- 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 steakholders 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 store in the database as a decimal percentage value, for example 20% is stored as 0.20, 30% as 0.30. 50% as 0.50 etc. This discount percentage (0.0x) is applied to the **Vehicle Rental Categories** which determines the price of each category to determine the total discount. Therefore, when a corporate customer rents a vehicle from a vehicle category (such as economic, compact, standard etc.), this discount percentage is applied to each of the categories during the rental/reservation process. Note that every company has a different percentage rating depending on their contract with **EZ-Rentals Inc**. For example, some companies have 20% discount towards their rentals, which would be stored as 0.20 in the database, some have 30% (0.30) etc. Vehicle Rental Categories are discussed in more details later in these requirements.

Retail Customers

Retail 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..**

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 curren

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:

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

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

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

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

1. *A vehicle can be reserved to be picked up at the **INDICATED** rental agency and dropped off at the **SAME** rental agency.*
2. *A vehicle can be reserved to be picked up at the **INDICATED** rental agency and dropped off at a **DIFFERENT** rental agency.*
3. *A reservation is made only for one pick-up rental agency, but a rental agency can have many reservations for pick-ups taking place.*
4. *A reservation can only be for one drop-off rental agency, but a rental agency can have many reservations drop-offs taking place.*

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

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

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

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

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

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

Business Requirements (Cont.)

The Rental Process (Cont.):

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

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

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

Our Employees:

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

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

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-0f-Sales & Back-End Management systems. In addition to our **EZRental.com** portal by our customers. Therefore, due to security and regulatory compliance purpose, we want to separate the employee access data from the customer access data by using two separate user accounts:

- Employee user accounts
- Customer user accounts

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

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

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

Business Requirements (Cont.)

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

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

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

Vehicle Transportation:

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

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

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

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

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

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

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

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

Business Requirements (Cont.)

Conclusion:

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

Database Design Deliverable #1b – Application Development Technical Requirements

The Application Analyst/Architect interviewed the project's business and technical decision makers (TDMs) and stakeholders to understand their needs and expectations for the application design and development.

The Application Analyst/Architect documented the Application Development & Technical Requirements below:

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 is 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-0f-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	▪ Car Rental, Car Return, New Customer Registration & Search/Print Customer Information, Customer Update, Customer Deletion, Customer Listing operations etc.
2	**EZRental** Rental Agency Back-Office Vehicle Inventory Management System	▪ Back-office system meant for employees to perform bulk IN-MEMORY inventory processing or management tasks on vehicles such as adding vehicles to the system, searching for vehicles, updating vehicles etc. ▪ This system is NOT meant for Point-of-Sales, but for the inventory management employees who need to search, add, remove etc., a large/bulk number of vehicles or employees during a session. ▪ Back-office vehicle Management features – Allows inventory personnel and employees to bulk-manage Cars, SUVs, Mini-Vans, Cargo Vans to be searched, added, removed, printed, listed etc.
3	**EZRental** Rental Agency Back-Office Credit Card Management System	▪ The EZRental Credit Card Management System is a Back-office system meant for the Credit Card Department Employees to manage Credit Card Information. These uses can Search/Print, Add, Edit & Delete credit card information in the database
4	**EZRental** Rental Agency Back-Office Employee & Customer User Account Management System	▪ The EZRental Customer & Employee User Account Management System is a Back-end system meant for IT ADMINISTRATOR Employees to manage both Employee & Customer USER ACCOUNTS.
5	**EZRental** Rental Agency Desktop Application Security Authentication System	▪ Proper security and authentication must be implemented to make sure only authorized employees can access the Point-0f-Sales, Back-End Management system or any other access to the applications.

Rental Agency Application Graphical User Interface Requirements:

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

  - **Addresses**

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

  - **Discount Codes:**

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

Application Development & Technical Requirements

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

- **EZPlus Rewards Codes:**

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

- **Company Name:**

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

- **Vehicle Status:**

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

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

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

- **Rental Agency:**

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

Application Development & Technical Requirements

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

- **Vehicle Rental Category:**

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

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

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

- **Reservation Status:**

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

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

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

Application Development & Technical Requirements

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

- **Rental Status:**

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

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

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

- **Rental Fuel Option:**

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

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

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

Application Development & Technical Requirements

- **Rental Insurance Option:**

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

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

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

Application Development & Technical Requirements

- **Transportation Reason Option:**

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

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

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

- **Transportation Reason Option:**

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

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

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

Application Development & Technical Requirements (Cont.)

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

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

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

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

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

Web Portal Application Web Pages User Interface Requirements:

The web pages graphical UI requirements are listed below:

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

  - **Addresses**

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

  - **Discount Codes:**

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

Application Development & Technical Requirements

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

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

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

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

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

Application Physical Technical Architecture

This architecture was developed subsequent to a comprehensive analysis of the **business** and **technical requirements**.

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.

- Efficient response time and optimal performance are critical in swiftly processing customer reservations, rentals, returns, and management, particularly in high-traffic locations such as airports. To ensure a rich user-interface experience for our customer service agents, the INTRANET Web Portal must be equipped with advanced features. Furthermore, the back-end personnel, including vehicle inventory managers, administrators, service personnel, and vehicle transport drivers, rely on the system in the agencies, underscoring the importance of optimal performance in meeting their needs.

- **Corporate Offices:**

  The corporate offices serve as the central hub for managing our business operations through the INTRANET Web Portal, which is accessed by both our business employees and employees at the rental agencies. Notable features of this portal include:

  - Intranet Web Enterprise Resource Planning Systems (ERP) Portal Feature such as providing access to Enterprise Resource Planning Systems (ERP) Applications such as: Customer Credit Card Management System, Vehicle Inventory Management System, Customer Relationship Management (CRM), Human Resource Management System, & Finance & Operations System, Marketing System, Customer & Field Service System etc.

  - Web EZRental Point-of-Sales Corporate Management Feature which allows employees to manage & execute Point-of-Sales (POS) transaction via the Intranet Web Poral such as: Search Customer Profile Information, Customer Account Management, Customer Registration, Customer Update, Customer Delete, & Customer Listing functionalities, Manage & Make Reservations of a Vehicle, Manage an existing Rental, etc.

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

- **Customer self-service:**
  - 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.

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

Application Development Features and Functionalities (Agile Backlog)

After conducting thorough analysis and meetings, it was determined that the application must incorporate the following 10 features and functionalities, collectively forming the AGILE BACKLOG.

Note: It is important to note that the features highlighted in BLUE and ORANGE are within the scope of PROJECT #1, while the features highlighted in GREY are not.

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

Feature #	Feature Description
FEATURE #3A	**FEATURE #3A** **FEATURE #3A – EZRental Internal Back-Office Agency BACK-OFFICE VEHICLE INVENTORY MANAGEMENT SYSTEM (NOT A CUSTOMER FACING APPLICATION):** ▪ **WINDOWS CLIENT POINT-OF SALES SYSTEM** – ***WINDOWS UI FORM(S) FRONT-END APPLICATION & OOP PROGRAMMING*** features used by **Back-Office Inventory Team employees** via a **computer machine** in the ***Rental Agencies*** to service inventory needs for **VEHICLE INVENTORY MANAGEMENT,** or transactions. ▪ 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: ○ **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. - **Feature UI Form Requirements:** ***Design & programming*** of required ***User-Interface Forms & GUI Controls*** to support this feature. - **Feature Processing Requirements:** ***Design & programming*** of required ***Object-Oriented (OOP) Processing & Logic*** to in the 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	**FEATURE #3B** **FEATURE #3B – EZRental Rental Agency Vehicle Inventory 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 #4A	**FEATURE #4A – IN-SCOPE FOR CST4708 CLASS ONLY END OF THIS SEMESTER IN PROJECT #2** **FEATURE #4A – EZRental Rental Agency Point-of-Sales (POS) BACK-OFFICE CREDIT CARD MANAGEMENT SYSTEM:** ▪ **WINDOWS CLIENT POINT-OF SALES SYSTEM** – ***WINDOWS UI FORM(S) FRONT-END APPLICATION & OOP PROGRAMMING*** features is a **back-end system** used by ***customer service representative & other employees*** via the ***Point-of-Sales computer*** machine in the ***Rental Agencies*** to service customer's **CREDIT CARD MANAGEMENT,** or transactions required when servicing customers. ▪ The following are features and functionality are required for this application with features such as: ○ **POS Credit Card Management Feature:** **POS Customer Credit Card Management features** such as ***Credit Card Search & Print, New Credit Card Registration***, ***Credit Card Update***, ***Credit Card Deletion***, & ***Credit Card Listing functionalities:*** - **Feature UI Form Requirements:** ***Design & programming*** of required ***User-Interface Forms & GUI Controls*** to support this feature. - **Feature Processing Requirements:** ***Design & programming*** 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 #4B	**FEATURE #4B – IN-SCOPE FOR CST4708 & CST3604 CLASSES FOR PROJECT #1 & PROJECT #2** **FEATURE #4B – EZRental Rental Agency Point-of-Sales (POS) Credit Card 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 #5A	**FEATURE #5A** **FEATURE #5A – EZRental Rental Agency Point-of-Sales (POS) System BACK-OFFICE EMPLOYEE & CUSTOMER USER-ACCOUNT MANAGEMENT SYSTEM:** ▪ **WINDOWS CLIENT POINT-OF SALES SYSTEM** – ***WINDOWS UI FORM(S) BACK-OFFICE APPLICATION & OOP PROGRAMMING*** **User-Accounts Management Features** used by ***customer service representative & IT Administrator employees*** via the ***Point-of-Sales computer*** machine in the ***Rental Agencies*** to service customer's **CUSTOMER & EMPLOYEE USER ACCOUNT MANAGEMENT** requests or transactions. ▪ **Employee User Accounts –** These are the user accounts used by ***IT Administrators***, ***Customer Service Employees***, ***back-office employees***, and ***any employee who qualifies*** for access to the system. ▪ **Customer User Accounts –** These are the user accounts used by ***IT Administrators***, ***Customer Service Employees***, ***back-office employees*** and any ***employee*** who has access to the system to ***manage*** the **Customer User Accounts** for login into the **Customer Web Portal**. ▪ The following are features and functionality that are required for this application feature: ○ **POS User Account (Employee & Customer) Management Feature:** **POS User Account Management (Employee & Customer) features** such as: - **Employee User Account Feature 5A-1 –** Allows ***IT Administrators***, ***Customer Service Employees***, ***back-office employees***, and ***any employee who qualifies*** to ***manage*** employee user accounts that allow employees to login into the POS System. And perform the following tasks: ***Employee User Account Search by username, New Employee User Account Registration***, ***Employee User Account Update by username***, ***Employee User Account Deletion by username***, & ***Employee User Account Listing functionalities***. **IMPORTANT!** Note that the **password** is ***NEVER DISPLAYED or LISTED***, only the **username**! - **Customer User Account Feature 5A-2 –** Allows ***IT Administrators***, ***Customer Service Employees***, ***back-office employees***, and ***any employee who qualifies*** to ***manage*** customer user accounts that allow customers to login into the **Customer Web Portal System**. And perform the following tasks: ***Customer User Account Search by username, New Customer User Account Registration***, ***Customer User Account Update by username***, ***Customer User Account Deletion deletion by username***, & ***Customer User Account Listing functionalities***. **IMPORTANT!** Note that the **password** is ***NEVER DISPLAYED or LISTED***, only the **username**! ○ ***Note that each transaction is saved to database immediately after execution:*** - **Feature UI Form Requirements:** ***Design & programming*** of required ***User-Interface Forms & GUI Controls*** to support this feature. - **Feature Processing Requirements:** ***Design & programming*** of required ***Object-Oriented (OOP) Processing & Logic*** to support this feature. ▪ This feature is designed only to be used by ***IT Administrations*** and other employees who qualify to use this system to manage both employees & customers user accounts using the **Windows Two-Tiered Client/Server Application** in the **Rental Agencies**.
FEATURE #5B	**FEATURE #5B** **FEATURE #5B – EZRental Rental Agency Point-of-Sales (POS) User EMPLOYEE User Account 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 to store and manage EMPLOYEE USER ACCOUNTS.**

FEATURE #5C	**FEATURE #5C – IN-SCOPE FOR CST4708 & CST3604 CLASSES FOR PROJECT #1 & PROJECT #2** **FEATURE #5C – EZRental Rental Agency Point-of-Sales (POS) User CUSTOMER User Account 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 to store and manage CUSTOMER USER ACCOUNTS.**

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

Feature #	Feature Description
FEATURE #7A	**FEATURE #7A** **FEATURE #7A – EZRental INTERNAL CORPORATE EMPLOYEE & RENTAL AGENCIES EMPLOYEES INTRANET CORPORATE BUSINESS APPLICATIONS WEB PORTAL:** ▪ This **INTRANET (NOT THE PUBLIC INTERNET) Web Portal** EZRentalHub.com, is a **Web-based Three-Tiered Client/Server physical & Software Development Architecture** used by **CORPORATE & OTHER EMPLOYEES** to ***execute*** **Enterprise Resource Planning Systems (ERP) Applications & other Corporate Applications** online intranet via a **BROWSER**. ▪ **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**: ○ **Corporate Business Application Intranet Web Portal Features:** - **Intranet Web Enterprise Resource Planning Systems (ERP) Portal Feature 7A-1 –** Provides access to **Enterprise Resource Planning Systems (ERP) Applications** such as: ***Customer Credit Card Management System***, ***Vehicle Inventory Management System***, ***Customer Relationship Management (CRM)***, ***Human Resource Management System***, & ***Finance & Operations System, Marketing System, Customer & Field Service System etc***. - **Web EZRental Point-of-Sales Corporate Management Feature 7A-2 –** Allows Employees to ***manage*** & ***execute*** **Point-of-Sales (POS)** transaction via the **Intranet Web Poral** such as: ***Search Customer Profile Information***, ***Customer Account Management***, ***Customer Registration***, ***Customer Update***, ***Customer Delete***, & ***Customer Listing functionalities, Manage & Make Reservations of a Vehicle***, ***Manage an existing Rental***, ***etc.*** ○ Programming includes: - **Feature UI Form Requirements:** ***Design & programming*** of required ***INTRANET WEB*** ***User-Interface Forms & GUI Controls*** to support this feature. - **Feature Processing Requirements:** ***Design & programming*** of required ***INTRANET WEB TECHONLOGY***, ***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** **FEATURE #7B – EZRental Corporate Employee & Rental Agencies Employees INTRANET 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 #8A	**FEATURE #8A** **FEATURE #8A – EZRental EXTERNAL CUSTOMER SELF-SERVICE INTERNET CUSTOMER FACING POINT-OF-SALES (POS) WEB PORTAL:** ▪ 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: ○ **POS Reservation & Management Features:** - **Web POS Authentication System Feature 8A-1 –** Proper **security and authentication** must be implemented to make sure **only the authorized customer can access to its Point-0f-Sales portal and login and out of their profile website**. - **Web POS Customer Self-Service Management Feature 8A-2 –** Allows POS Customer (Retail Customer & Corporate Customer) Self-Service Management of their account such as: ***Customer Profile Information***, ***Customer Account & Login Registration***, ***Customer Update Profile***, ***Customer Delete Profile***, & ***Customer Listing functionalities such as listing of Reservations & Rental History etc***. - **Web POS Customer Self-Service Point-of-Sales Management Feature 8A-3 –** Allows POS Customer (Retail Customer & Corporate Customer) Self-Service features to make reservations and manage rentals such as: ***Make Reservations of a Vehicle***, ***Manage an existing Rental***, ***etc***. - **Web POS User Account Management Feature 8A-4 –** Allows POS Customer (Retail Customer & Corporate Customer) Self-Service features to enable customer to manage its **User Account** and perform the following operations: ***Reset Username*** & ***Reset Password.*** ○ Programming includes: - **Feature UI Form Requirements:** ***Design & programming*** of required ***WEB*** ***User-Interface Forms & GUI Controls*** to support this feature. - **Feature Processing Requirements:** ***Design & programming*** of required ***WEB TECHONLOGY***, ***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 Serf-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 #10A	**FEATURE #10A** **FEATURE #10A – EZRental INTERNAL CORPORATE EMPLOYEE & RENTAL AGENCIES EMPLOYEES INTRANET BACK-OFFICE VEHICLE TANSPORT MANAGEMENT SYSTEM WEB PORTAL:** ▪ This **INTRANET Web Portal** EZRentalHub.com, is a **Web-based Three-Tiered Client/Server physical & Software Development Architecture** used by **CORPORATE & AGENCY EMPLOYEES** to *manage* **Transportation of Vehicles by Employee Drivers to and from Rental Agencies, Vehicle Distribution Centers, and other Locations** via a **BROWSER**. ▪ **WEB TRANSOPORT MANAGEMENT SYSTEM APPLICATION** – ***WEB UI FORM(S) FRONT-END APPLICATION & OOP PROGRAMMING*** **Transport Management Features** used by ***Vehicle Transportation Managers & Drivers Employees*** to handle the day-to-day vehicle transportation process via the **CORPORATE INTRANET** *PORTAL*. ▪ The following are features and functionality that are required for this **INTRANET Transport Management WEB APPLICATION**: ○ **Corporate Vehicle Transport Application Intranet Web Portal Features:** - **Transport Scheduling Feature –** handle the day-to-day creating & scheduling of a pic-up & delivery (Any vehicle type) such as: ***Creation of NEW Vehicle Transport Request***, ***Vehicle Pick-up***, ***Vehicle Drop-off*** & ***Vehicle Transport Status etc***. ○ Programming includes: - **Feature UI Form Requirements:** ***Design & programming*** of required ***INTRANET WEB*** *User-Interface Forms & GUI Controls* to support this feature. - **Feature Processing Requirements:** ***Design & programming*** of required ***INTRANET WEB TECHONLOGY***, ***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 #10B	**FEATURE #10B** **FEATURE #10B – EZRental Corporate Vehicle Transport Application Intranet Web Portal Features 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.**

Database Management System Development Environment & Physical Architecture

To develop the database environment for this project, EZRental utilized Microsoft SQL Server 2019 Express as the Database Management System. This robust system is equipped with several helpful tools, including the SQL command-line admin tool and Microsoft SQL Server Management Studio GUI admin tools.

Project Roles & Responsibilities

The Business/Database Analyst will be responsible for assembling the necessary database development team. The table below outlines each of the roles required for the project and the corresponding individual(s) who will carry out these roles:

Person	Role	Description
Mr. Rodriguez	**Program Manager, AgileScrum Master & ProjectManager**	▪ Owner of the project and liaison to Manage the **EZRental Inc**., the customer. ▪ Activities include but not limited to. 1. Owner of project responsible for the success of the project. 2. Project Management 3. Scrum Master ensures the project stays on time and moving in the right direction. Clear any obstacles impeding the team's progress etc.
Consultant #1: **Mr. Rodriguez**	**Business & Database Analyst**	▪ 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: 1. Engage in discovery activities & interview the stakeholders at **EZRental Inc**. ▪ From the interview and discovery ***create*** **1) ER/EER ConceptualData Model from the business requirements & 2) Normalized Logical Model**.
Consultant #2, 3, 4 & 5 **Olamide Yomi**	**Database Developers**	▪ This role uses the Normalized Logical Model created by consultant #2 to create the Data Dictionary, Physical Schema Diagram, and Implement the Database Application forthe Auto Rental System. ▪ Activities include but not limited to: 1. Use the ***Normalized Logical Model*** created by consultant #2 to do the **following:** 1) **Create Data Dictionary tables for each logical table targeting MS SQL Server 2019 Express DBMS Data Types.** 2) **Create Physical Schema Diagram**. 2. From these two deliverables, 1) **implement the Database Application using Oracle 18c** for the Auto Rental System.
Consultant #6 **Olamide Yomi**	**Database Administrator**	▪ The DB Admin, install the DBMS, maintain, and operate the DBMS throughout its lifetime. ▪ Activities include but not limited to: 1. As DB Admin, you are to **1) Setup & install MS SQL Server 2019 Express DBMS.** **2) Administrative tools for target DBMS**. 2. Also, as DB Admin, you are to **3) Operate & Maintain the DBMS**.

The Application Technical Architect/Analyst coordinated the necessary application development team, and the table below outlines the roles for each Windows client and Browser client application development project, as well as the individuals who will perform these roles:

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

Database Design Deliverable #2 – ER/EER Conceptual Model Diagram

As part of the analysis phase deliverables under the waterfall methodology, the ER/EER conceptual model is designed to illustrate the relationship between the business entities identified in the application's business requirements.

Below is detailed visual representation on the ER/EER Conceptual Model Diagram:

Database Design Deliverable #3 – Normalized Logical Model Diagram

In the design phase of the waterfall methodology, the Database Analyst/Architect has created the Normalized Logical Model by converting the entities in the ER/EER Diagram (the Logical Model Diagram) into database tables. This model is derived from the ER/EER Diagram and is a crucial part of the design process.

A small pilot project has been implemented to test and validate the database design based on the ER/EER Diagram. This was done to ensure that the design is functional and effective before proceeding with full-scale implementation.

Below is a Proof of Concept Normalized Logical Model for a prototype application that will be demonstrated to the business. The diagram includes only the four logical tables within the scope of the Proof of Concept.

Database Design Deliverable #4 – Physical Model Data Dictionary

The Physical Model Data Dictionary outlined below offers a comprehensive overview of the schema, featuring table names, column names, data types, key types, and constraints for each attribute. The four Physical Model Data Dictionary tables that constitute a Customer Dictionary Table are designed for compatibility with Microsoft SQL Server DBMS and Oracle Server DBMS Data Types:

1. USSTATE table: Stores information about U.S. states, comprising unique IDs, two-character state codes, and complete state names.
2. COUNTRY table: Holds data on countries, including unique IDs, two-character and three-character country codes, and full country names.
3. CREDITCARD table: Captures credit card details, such as card numbers, owner names, issuing companies, merchant codes, expiration dates, addresses, city, state codes, zip codes, countries, credit card limits, balances, and activation statuses.
4. CREDITCARDMERCHANT table: Contains information on credit card merchants, featuring unique merchant codes and comprehensive merchant names.

Below is a diagram that illustrates the relationships and descriptions of the four tables, complete with detailed definitions and documentation of the data elements within the data model:

#1

CREDITCARD							
Column Num.	**Attribute/Column Name**	**Generic Data Type Name**	**MS SQL SERVER Data Type Name**	**Is it Required?**	**Length/Size /Format**	**Constraints**	**Description/ purpose**
1.	CreditCardNumber	String	VARCHAR (16)	Yes	16	Primary key	Primary Key credit card number
2.	CreditCardOwnerName	String	VARCHAR (50)	Yes	50	Not Null	Full name of the credit card owner
3.	CreditCardIssuingCompany	String	VARCHAR (50)	Yes	50	Not Null	Issuing company/bank of the credit card (e.g., Visa, Mastercard)
4.	MerchantCode	String	TINYINT	Yes	N/A	Not Null	Unique merchant code for identification
5.	ExpDate	Date	DATE	Yes	MM-DD-YY	Not Null	Expiration date of the credit card
6.	AddressLine1	String	VARCHAR (50)	Yes	50	Not Null	Address line 1 of the credit card billing address
7.	AddressLine2	String	VARCHAR (50)	No	50	Nullable	Address line 2 of the credit card billing address (optional)
8.	City	String	VARCHAR (50)	Yes	100	Not Null	City of the credit card billing address
9.	StateCode	String	CHAR (2)	Yes	2	Not Null	2-character state code of the credit card billing address (e.g., CA for California)
10.	Zipcode	String	VARCHAR (10)	Yes	10	Not Null	Zip code of the credit card billing address
11.	Country	String	VARCHAR (50)	Yes	50	Not Null	Country of the credit card billing address
12.	CreditCardLimit	Decimal	DECIMAL (8,2)	Yes	N/A	Not Null, Greater than or equal to 0	Credit limit of the credit card
13.	CreditCardBalance	Decimal	DECIMAL (8,2)	Yes	N/A	Not Null, Greater than or equal to 0	Current balance on the credit card
14.	ActivationStatus	Boolean	BIT	Yes	N/A	Not Null	Activation status of the credit card (True: Active, False: Inactive)

#2

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.	**MerchantCode**	INT	TINYINT	Yes	N/A	Primary key, Check, (MerchantCode between 1 and 15)	Unique merchant code for identification
2.	**MerchantName**	String	VARCHAR (50)	Yes	50	Unique, Not Null	Full name of the credit card merchant

#3

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	Integer	TINYINT	Yes	N/A	Primary Key, Check (StateID between 1 and 75)	Unique identifier for each US state
2.	StateCode2Char	String	CHAR (2)	Yes	2	Unique, Not Null	2-character state code abbreviation (e.g., CA for California)
3.	StateName	String	VARCHAR (50)	Yes	50	Unique, Not Null	Full name of the US state (e.g., California)

#4

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	Integer	TINYINT	Yes	N/A	Primary key, Check (CountryID between 1 and 250)	Unique identifier for each country
2.	CountryCode2Char	String	CHAR (2)	Yes	2	Unique, Not Null	2-character country code abbreviation (e.g., US for United States)
3.	CountryCode3Char	String	CHAR (3)	Yes	3	Unique, Not Null	3-character country code abbreviation (e.g., USA for United States)
4.	CountryName	String	VARCHAR (100)	Yes	100	Unique, Not Null	Full name of the country (e.g., United States)

Database Design Deliverable #5 – Physical Model Schema Design Diagram

This physical schema represents a well-structured and efficient database design tailored to address the needs of EZ-Rental POS. It comprises four interrelated tables, each containing essential information about specific entities: US states, countries, credit cards, and credit card merchants.

The physical schema has been designed with EZRental POS in mind, ensuring optimal performance and data integrity to support the efficient operation of the system.

Database Implementation Deliverable #6 – Development & Implementation

The EZRental POS system's database schema is designed to optimize data management and maintain referential integrity. The schema includes four essential tables:

1. USSTATE table: Stores U.S. states information.
2. COUNTRY table: Holds data on countries.
3. CREDITCARD table: Contains credit card details.
4. CREDITCARDMERCHANT table: Captures information about credit card merchants.

A comprehensive Database Implementation Deliverable has been prepared, outlining the SQL statements required to create and configure the database and its associated tables. This deliverable ensures seamless integration with existing systems and processes.

Below is a diagram of the SCRIPT FILE, which illustrates the SQL statements for creating the tables and defining their relationships and constraints. This diagram serves as a visual guide to understanding the structure and design of the EZ-Rental POS system's database schema.

```sql
--CREATE DATABASE
CREATE DATABASE EZRentalDB;

--USE DATABASE
USE EZRentalDB;

-- CREDITCARDMERCHANT table
CREATE TABLE CreditCardMerchant (
  MerchantCode TINYINT PRIMARY KEY CHECK (MerchantCode BETWEEN 1 AND 15),
  MerchantName VARCHAR(50) UNIQUE NOT NULL
);

-- CREDITCARD table
CREATE TABLE CreditCard (
  CreditCardNumber VARCHAR(16) PRIMARY KEY,
  CreditCardOwnerName VARCHAR(50) NOT NULL,
  CreditCardIssuingCompany VARCHAR(50) NOT NULL,
  MerchantCode TINYINT NOT NULL CHECK(MerchantCode >= 1 AND MerchantCode <= 15),
  ExpDate DATE NOT NULL,
  AddressLine1 VARCHAR(50) NOT NULL,
  AddressLine2 VARCHAR(50) NULL,
  City VARCHAR(50) NOT NULL,
  StateCode CHAR(2) NOT NULL,
  Zipcode VARCHAR(10) NOT NULL,
  Country VARCHAR(50) NOT NULL,
  CreditCardLimit DECIMAL(8, 2) NOT NULL,
  CreditCardBalance DECIMAL(8, 2) NOT NULL,
  ActivationStatus BIT NOT NULL,

  CONSTRAINT fk_CreditCardMerchant_CreditCard
  FOREIGN KEY (MerchantCode)
  REFERENCES CreditCardMerchant(MerchantCode)
  ON DELETE CASCADE ON UPDATE CASCADE
);

-- USSTATE table
CREATE TABLE USState (
  StateID TINYINT PRIMARY KEY CHECK(StateID BETWEEN 1 AND 75),
  StateCode2Char CHAR(2) UNIQUE NOT NULL,
  StateName VARCHAR(30) UNIQUE NOT NULL
);

-- COUNTRY table
CREATE TABLE Country (
  CountryID TINYINT PRIMARY KEY CHECK(CountryID BETWEEN 1 AND 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

The EZRental POS system's database schema is designed for efficient data management and referential integrity. A database diagram has been created to visualize the organization and relationships between the four core tables: USSTATE, COUNTRY, CREDITCARD, and CREDITCARDMERCHANT.

The diagram offers an intuitive representation of the tables and their connections within the schema. Below is the attached database diagram which helps to understand the architecture and functionality of the EZRental POS system's database.

Insert Statements.

Insert statements for CREDITCARDMERCHANT table.

Insert statements contains the following information for the **CREDITCARDMERCHANT** table:

1. Merchant Code

2. Merchant Name

Below is an empty table before any insert statements were executed.

```
Select *
from CREDITCARDMERCHANT;
```

We created the **INSERT QUERY** for this table and the updated table now looks like this:

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

Results | Messages

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

Insert statements for CREDITCARD table.

Insert statements contains the following information for the **CREDITCARD** table:

1. Credit Card Number
2. Credit Card Owner Name
3. Credit Card Issuing Company
4. Merchant Code
5. Expiration Date
6. Address Line 1
7. Address Line 2
8. City
9. State Code
10. Zip Code
11. Country
12. Credit Card Limit
13. Credit Card Balance
14. Activation Status

Below is an empty table before any insert statements were executed.

```
Select *
from CREDITCARD;
```

Results | Messages

CreditCardNumber	CreditCardOwnerName	CreditCardIssuingCompany	MerchantCode	ExpDate	AddressLine1	AddressLine2	City	StateCode	Zipcode	Country	CreditCardLimit	CreditCardBalance	ActivationStatus

We created the **INSERT QUERY** for this table and the updated table now looks like this:

```
Insert into CREDITCARD
       (CreditCardNumber, CreditCardOwnerName, CreditCardIssuingCompany,
       MerchantCode, ExpDate, AddressLine1, AddressLine2, City, StateCode, ZipCode, Country,
       CreditCardLimit, CreditCardBalance, ActivationStatus)
values
       ('1111111111111111', 'John Williams', 'American Express', 1, '01/01/2024', '1111 Jay St.',
              '', 'Brooklyn', 'NY', '11209', 'USA', 10000.00, 256.21, 'True');
Insert into CREDITCARD
       (CreditCardNumber, CreditCardOwnerName, CreditCardIssuingCompany,
       MerchantCode, ExpDate, AddressLine1, AddressLine2, City, StateCode, ZipCode, Country,
       CreditCardLimit, CreditCardBalance, ActivationStatus)
values
       ('2222222222222222', 'Samuel Nichols', 'TD Bank', 3, '03/20/2024', '2222 Jay St.',
              '', 'Trenton', 'NJ', '11229', 'USA', 8000.00, 4230.86, 'True');
Insert into CREDITCARD
       (CreditCardNumber, CreditCardOwnerName, CreditCardIssuingCompany,
       MerchantCode, ExpDate, AddressLine1, AddressLine2, City, StateCode, ZipCode, Country,
       CreditCardLimit, CreditCardBalance, ActivationStatus)
values
       ('3333333333333333', 'Silas Decker', 'American Express', 4, '05/22/2025', '3333 Jay St.',
              '', 'Trenton', 'NJ', '11239', 'USA', 10000.00, 256.21, 'True');
Insert into CREDITCARD
       (CreditCardNumber, CreditCardOwnerName, CreditCardIssuingCompany,
       MerchantCode, ExpDate, AddressLine1, AddressLine2, City, StateCode, ZipCode, Country,
       CreditCardLimit, CreditCardBalance, ActivationStatus)
values
       ('4444444444444444', 'Michael Hussein', 'Chase Bank', 2, '01/01/2024', '4444 Jay St.',
              '', 'Manhattan', 'NY', '11249', 'USA', 10000.00, 256.21, 'False');
Insert into CREDITCARD
```

```sql
        (CreditCardNumber, CreditCardOwnerName, CreditCardIssuingCompany,
        MerchantCode, ExpDate, AddressLine1, AddressLine2, City, StateCode, ZipCode, Country,
        CreditCardLimit, CreditCardBalance, ActivationStatus)
values
        ('5555555555555555', 'Valerie Holmes', 'American Express', 1, '01/01/2024', '5555 Jay St.',
                '', 'Brooklyn', 'NY', '11259', 'USA', 500.00, 256.21, 'True');
Insert into CREDITCARD
        (CreditCardNumber, CreditCardOwnerName, CreditCardIssuingCompany,
        MerchantCode, ExpDate, AddressLine1, AddressLine2, City, StateCode, ZipCode, Country,
        CreditCardLimit, CreditCardBalance, ActivationStatus)
values
        ('6666666666666666', 'Arthur Smith', 'American Express', 9, '01/01/2024', '6666 Jay St.',
                '', 'Brooklyn', 'NY', '11269', 'USA', 500.00, 0.00, 'True');
Insert into CREDITCARD
        (CreditCardNumber, CreditCardOwnerName, CreditCardIssuingCompany,
        MerchantCode, ExpDate, AddressLine1, AddressLine2, City, StateCode, ZipCode, Country,
        CreditCardLimit, CreditCardBalance, ActivationStatus)
values
        ('7777777777777777', 'Emma Watson', 'Bank of America', 7, '06/15/2024', '7777 Jay St.',
                '', 'Los Angeles', 'CA', '90001', 'USA', 12000.00, 1000.00, 'True');
Insert into CREDITCARD
        (CreditCardNumber, CreditCardOwnerName, CreditCardIssuingCompany,
        MerchantCode, ExpDate, AddressLine1, AddressLine2, City, StateCode, ZipCode, Country,
        CreditCardLimit, CreditCardBalance, ActivationStatus)
values
        ('8888888888888888', 'Oliver Johnson', 'Wells Fargo', 2, '08/01/2025', '8888 Main St.',
                '', 'Austin', 'TX', '73301', 'USA', 15000.00, 1200.00, 'True');
Insert into CREDITCARD
        (CreditCardNumber, CreditCardOwnerName, CreditCardIssuingCompany,
        MerchantCode, ExpDate, AddressLine1, AddressLine2, City, StateCode, ZipCode, Country,
        CreditCardLimit, CreditCardBalance, ActivationStatus)
values
        ('9999999999999999', 'Sophia Brown', 'PNC Bank', 10, '12/01/2024', '9999 Oak St.',
                '', 'Chicago', 'IL', '60601', 'USA', 11000.00, 500.00, 'True');
Insert into CREDITCARD
        (CreditCardNumber, CreditCardOwnerName, CreditCardIssuingCompany,
        MerchantCode, ExpDate, AddressLine1, AddressLine2, City, StateCode, ZipCode, Country,
        CreditCardLimit, CreditCardBalance, ActivationStatus)
values
        ('1234567890123456', 'Noah Davis', 'Citi Bank', 8, '11/01/2025', '1234 Elm St.',
                '', 'San Francisco', 'CA', '94102', 'USA', 9000.00, 300.00, 'True');
```

Results | Messages

	CreditCardNumber	CreditCardOwnerName	CreditCardIssuingCompany	MerchantCode	ExpDate	AddressLine1	AddressLi...	City	StateCode	Zipcode	Country	CreditCardLimit	CreditCardBalance	ActivationStatus
1	1111111111111111	John Williams	American Express	1	2024-01-01	1111 Jay St.		Brooklyn	NY	11209	USA	10000.00	256.21	1
2	1234567890123456	Noah Davis	Citi Bank	8	2025-11-01	1234 Elm St.		San Francisco	CA	94102	USA	9000.00	300.00	1
3	2222222222222222	Samuel Nichols	TD Bank	3	2024-03-20	2222 Jay St.		Trenton	NJ	11229	USA	8000.00	4230.86	1
4	3333333333333333	Silas Decker	American Express	4	2025-05-22	3333 Jay St.		Trenton	NJ	11239	USA	10000.00	256.21	1
5	4444444444444444	Michael Hussein	Chase Bank	2	2024-01-01	4444 Jay St.		Manhattan	NY	11249	USA	10000.00	256.21	0
6	5555555555555555	Valerie Holmes	American Express	1	2024-01-01	5555 Jay St.		Brooklyn	NY	11259	USA	500.00	256.21	1
7	6666666666666666	Arthur Smith	American Express	9	2024-01-01	6666 Jay St.		Brooklyn	NY	11269	USA	500.00	0.00	1
8	7777777777777777	Emma Watson	Bank of America	7	2024-06-15	7777 Jay St.		Los Angeles	CA	90001	USA	12000.00	1000.00	1
9	8888888888888888	Oliver Johnson	Wells Fargo	2	2025-08-01	8888 Main St.		Austin	TX	73301	USA	15000.00	1200.00	1
10	9999999999999999	Sophia Brown	PNC Bank	10	2024-12-01	9999 Oak St.		Chicago	IL	60601	USA	11000.00	500.00	1

Select statements.

Select statement #1

Select statement that queries **CREDITCARD** table returning only **1 record** that includes all columns based on the primary key.

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

The result of the query that has been executed, is all information that was previously populated and

associated with the given primary key (in this case with CreditCard)

Results | Messages

	CreditCardNumber	CreditCardOwnerName	CreditCardIssuingCompany	MerchantCode	ExpDate	AddressLine1	AddressLine2	City	StateCode	Zipcode	Country	CreditCardLimit	CreditCardBalance	ActivationStatus
1	1111111111111111	John Williams	American Express	1	2024-01-01	1111 Jay St..		Brooklyn	NY	11209	USA	10000.00	256.21	1

Select statement #2

Select statement that queries **CREDITCARD** table returning **all record** that includes all columns.

based on the criteria.

```
SELECT *
FROM CREDITCARD
WHERE CreditCardIssuingCompany = 'American Express';
```

The result of the query that has been executed, is all rows and columns that was previously populated

and associated with the given criteria in the CreditCard table (in this case all records that have American Express in

the CreditCardIssuingCompany Column).

Results | Messages

	CreditCardNumber	CreditCardOwnerName	CreditCardIssuingCompany	MerchantCode	ExpDate	AddressLine1	AddressLine2	City	StateCode	Zipcode	Country	CreditCardLimit	CreditCardBalance	ActivationStatus
1	1111111111111111	John Williams	American Express	1	2024-01-01	1111 Jay St..		Brooklyn	NY	11209	USA	10000.00	256.21	1
2	3333333333333333	Silas Decker	American Express	4	2025-05-22	3333 Jay St..		Trenton	NJ	11239	USA	10000.00	256.21	1
3	5555555555555555	Valerie Holmes	American Express	1	2024-01-01	5555 Jay St..		Brooklyn	NY	11259	USA	500.00	256.21	1
4	6666666666666666	Arthur Smith	American Express	9	2024-01-01	6666 Jay St..		Brooklyn	NY	11269	USA	500.00	0.00	1

Select statement #3

Here is a **SELECT** query using just the **CREDITCARD** and **CREDITCARDMERCHANT** tables:

```
USE EZRentalDB;

SELECT cc.CreditCardNumber, cc.CreditCardOwnerName, cc.CreditCardIssuingCompany, cc.MerchantCode,
       cc.ExpDate, cc.CreditCardLimit, cc.CreditCardBalance, cc.ActivationStatus, ccm.MerchantName
FROM CREDITCARD cc, CREDITCARDMERCHANT ccm
WHERE cc.MerchantCode = ccm.MerchantCode AND cc.CreditCardIssuingCompany = 'American Express';
```

This SELECT query returns multiple records that include multiple columns from both the **CREDITCARD** (aliased as 'cc') and **CREDITCARDMERCHANT** (aliased as 'ccm') tables. The query is based on the relationship between the two tables using the MerchantCode and filters the results for credit cards issued by American Express.

Results | Messages

	CreditCardNumber	CreditCardOwnerName	CreditCardIssuingCompany	MerchantCode	ExpDate	CreditCardLimit	CreditCardBalance	ActivationStatus	MerchantName
1	1111111111111111	John Williams	American Express	1	2024-01-01	10000.00	256.21	1	Stax by Fattmerchant
2	3333333333333333	Silas Decker	American Express	4	2025-05-22	10000.00	256.21	1	Payment Depot
3	5555555555555555	Valerie Holmes	American Express	2	2024-01-01	500.00	15632.26	0	Helcim

Update statements

Update statement #1

An **UPDATE** query for the **CREDITCARD** table that updates all columns except the primary key column (CreditCardNumber).

The **CREDITCARD** before executing an update statement:

Results | Messages

	CreditCardNumber	CreditCardOwnerName	CreditCardIssuingCompany	MerchantCode	ExpDate	AddressLine1	AddressLine2	City	StateCode	Zipcode	Country	CreditCardLimit	CreditCardBalance	ActivationStatus
1	1111111111111111	John Williams	American Express	1	2024-01-01	1111 Jay St.		Brooklyn	NY	11209	USA	10000.00	256.21	1
2	1234567890123456	Noah Davis	Citi Bank	8	2025-11-01	1234 Elm St.		San Francisco	CA	94102	USA	9000.00	300.00	1
3	2222222222222222	Samuel Nichols	TD Bank	3	2024-03-20	2222 Jay St.		Trenton	NJ	11229	USA	8000.00	4230.86	1
4	3333333333333333	Silas Decker	American Express	4	2025-05-22	3333 Jay St.		Trenton	NJ	11239	USA	10000.00	256.21	1
5	4444444444444444	Michael Hussein	Chase Bank	2	2024-01-01	4444 Jay St.		Manhattan	NY	11249	USA	10000.00	256.21	0
6	5555555555555555	Valerie Holmes	American Express	2	2024-01-01	5555 Jay St.		Brooklyn	NY	11259	USA	500.00	15632.26	0
7	6666666666666666	Arthur Smith	American Express	9	2024-01-01	6666 Jay St.		Brooklyn	NY	11269	USA	500.00	0.00	1
8	7777777777777777	Emma Watson	Bank of America	7	2024-06-15	7777 Jay St.		Los Angeles	CA	90001	USA	12000.00	1000.00	1
9	8888888888888888	Oliver Johnson	Wells Fargo	2	2025-08-01	8888 Main St.		Austin	TX	73301	USA	15000.00	1200.00	1
10	9999999999999999	Sophia Brown	PNC Bank	10	2024-12-01	9999 Oak St.		Chicago	IL	60601	USA	11000.00	500.00	1

This query updates all columns except the primary key column (CreditCardNumber) for a single row in the **CREDITCARD** table where the primary key matches the specified value.

```
UPDATE CREDITCARD
SET CreditCardOwnerName = 'Emily Thompson',
    CreditCardIssuingCompany = 'Bank of America',
    MerchantCode = 2,
    ExpDate = '01/01/2025',
    AddressLine1 = '123 Elm St',
    AddressLine2 = 'Apt 4B',
    City = 'Raleigh',
    StateCode = 'NC',
    ZipCode = '27601',
    Country = 'USA',
    CreditCardLimit = 12000.00,
    CreditCardBalance = 500.00,
    ActivationStatus = 'True'
WHERE CreditCardNumber = '1111111111111111';
```

Results | Messages

	CreditCardNumber	CreditCardOwnerName	CreditCardIssuingCompany	MerchantCode	ExpDate	AddressLine1	AddressLine2	City	StateCode	Zipcode	Country	CreditCardLimit	CreditCardBalance	ActivationStatus
1	1111111111111111	John Williams	American Express	1	2024-01-01	1111 Jay St.		Brooklyn	NY	11209	USA	10000.00	256.21	1
2	1234567890123456	Noah Davis	Citi Bank	8	2025-11-01	1234 Elm St.		San Francisco	CA	94102	USA	9000.00	300.00	1
3	2222222222222222	Samuel Nichols	TD Bank	3	2024-03-20	2222 Jay St.		Trenton	NJ	11229	USA	8000.00	4230.86	1
4	3333333333333333	Silas Decker	American Express	4	2025-05-22	2 Metro Tech Center		Trenton	NJ	11239	USA	10000.00	256.21	1
5	4444444444444444	Michael Hussein	Chase Bank	2	2024-01-01	4444 Jay St.		Manhattan	NY	11249	USA	10000.00	256.21	0
6	5555555555555555	Valerie Holmes	American Express	2	2024-01-01	5555 Jay St.		Brooklyn	NY	11259	USA	500.00	15632.26	0
7	6666666666666666	Arthur Smith	American Express	9	2024-01-01	6666 Jay St.		Brooklyn	NY	11269	USA	500.00	0.00	1
8	7777777777777777	Emma Watson	Bank of America	7	2024-06-15	7777 Jay St.		Los Angeles	CA	90001	USA	12000.00	1000.00	1
9	8888888888888888	Oliver Johnson	Wells Fargo	2	2025-08-01	8888 Main St.		Austin	TX	73301	USA	15000.00	1200.00	1
10	9999999999999999	Sophia Brown	PNC Bank	10	2024-12-01	9999 Oak St.		Chicago	IL	60601	USA	11000.00	500.00	1

Update statement #2

Update statement that updates **one record**

The **CREDITCARD** before executing an update statement:

Results | Messages

	CreditCardNumber	CreditCardOwnerName	CreditCardIssuingCompany	MerchantCode	ExpDate	AddressLine1	AddressLine2	City	StateCode	Zipcode	Country	CreditCardLimit	CreditCardBalance	ActivationStatus
1	1111111111111111	John Williams	American Express	1	2024-01-01	1111 Jay St.		Brooklyn	NY	11209	USA	10000.00	256.21	1
2	1234567890123456	Noah Davis	Citi Bank	8	2025-11-01	1234 Elm St.		San Francisco	CA	94102	USA	9000.00	300.00	1
3	2222222222222222	Samuel Nichols	TD Bank	3	2024-03-20	2222 Jay St.		Trenton	NJ	11229	USA	8000.00	4230.86	1
4	3333333333333333	Silas Decker	American Express	4	2025-05-22	3333 Jay St.		Trenton	NJ	11239	USA	10000.00	256.21	1
5	4444444444444444	Michael Hussein	Chase Bank	2	2024-01-01	4444 Jay St.		Manhattan	NY	11249	USA	10000.00	256.21	0
6	5555555555555555	Valerie Holmes	American Express	1	2024-01-01	5555 Jay St.		Brooklyn	NY	11259	USA	500.00	256.21	1
7	6666666666666666	Arthur Smith	American Express	9	2024-01-01	6666 Jay St.		Brooklyn	NY	11269	USA	500.00	0.00	1
8	7777777777777777	Emma Watson	Bank of America	7	2024-06-15	7777 Jay St.		Los Angeles	CA	90001	USA	12000.00	1000.00	1
9	8888888888888888	Oliver Johnson	Wells Fargo	2	2025-08-01	8888 Main St.		Austin	TX	73301	USA	15000.00	1200.00	1
10	9999999999999999	Sophia Brown	PNC Bank	10	2024-12-01	9999 Oak St.		Chicago	IL	60601	USA	11000.00	500.00	1

Update statement that overrides the MerchantCode, CreditCardBalance and ActivationStatus in the **CREDITCARD** table:

```
Update CREDITCARD
set MerchantCode = '2', CreditCardBalance = 15632.26, ActivationStatus = 'False'
where CreditCardNumber = '5555555555555555';
```

Results | Messages

	CreditCardNumber	CreditCardOwnerName	CreditCardIssuingCompany	MerchantCode	ExpDate	AddressLine1	AddressLine2	City	StateCode	Zipcode	Country	CreditCardLimit	CreditCardBalance	ActivationStatus
1	1111111111111111	John Williams	American Express	1	2024-01-01	1111 Jay St.		Brooklyn	NY	11209	USA	10000.00	256.21	1
2	1234567890123456	Noah Davis	Citi Bank	8	2025-11-01	1234 Elm St.		San Francisco	CA	94102	USA	9000.00	300.00	1
3	2222222222222222	Samuel Nichols	TD Bank	3	2024-03-20	2222 Jay St.		Trenton	NJ	11229	USA	8000.00	4230.86	1
4	3333333333333333	Silas Decker	American Express	4	2025-05-22	3333 Jay St.		Trenton	NJ	11239	USA	10000.00	256.21	1
5	4444444444444444	Michael Hussein	Chase Bank	2	2024-01-01	4444 Jay St.		Manhattan	NY	11249	USA	10000.00	256.21	0
6	5555555555555555	Valerie Holmes	American Express	2	2024-01-01	5555 Jay St.		Brooklyn	NY	11259	USA	500.00	15632.26	0
7	6666666666666666	Arthur Smith	American Express	9	2024-01-01	6666 Jay St.		Brooklyn	NY	11269	USA	500.00	0.00	1
8	7777777777777777	Emma Watson	Bank of America	7	2024-06-15	7777 Jay St.		Los Angeles	CA	90001	USA	12000.00	1000.00	1
9	8888888888888888	Oliver Johnson	Wells Fargo	2	2025-08-01	8888 Main St.		Austin	TX	73301	USA	15000.00	1200.00	1
10	9999999999999999	Sophia Brown	PNC Bank	10	2024-12-01	9999 Oak St.		Chicago	IL	60601	USA	11000.00	500.00	1

Delete Statements

Delete statement #1

Delete statement that deletes one record of an associative entity table.

The **CREDITCARDMERCHANT** before executing delete statement:

Results | Messages

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

Delete statement that deletes all records of credit cards merchant name that have balance of clover:

```
DELETE FROM CREDITCARDMERCHANT
WHERE MerchantName = 'Clover';
```

Results | Messages

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

Delete statement #2

Delete statement that deletes one record of an associative entity table.

The **CREDITCARD** before executing delete statement:

Results | Messages

	CreditCardNumber	CreditCardOwnerName	CreditCardIssuingCompany	MerchantCode	ExpDate	AddressLine1	AddressLine2	City	StateCode	Zipcode	Country	CreditCardLimit	CreditCardBalance	ActivationStatus
1	1111111111111111	John Williams	American Express	1	2024-01-01	1111 Jay St.		Brooklyn	NY	11209	USA	10000.00	256.21	1
2	1234567890123456	Noah Davis	Citi Bank	8	2025-11-01	1234 Elm St.		San Francisco	CA	94102	USA	9000.00	300.00	1
3	2222222222222222	Samuel Nichols	TD Bank	3	2024-03-20	2222 Jay St.		Trenton	NJ	11229	USA	8000.00	4230.86	1
4	3333333333333333	Silas Decker	American Express	4	2025-05-22	2 MetroTech Center		Trenton	NJ	11239	USA	10000.00	256.21	1
5	4444444444444444	Michael Hussein	Chase Bank	2	2024-01-01	4444 Jay St.		Manhattan	NY	11249	USA	10000.00	256.21	0
6	5555555555555555	Valerie Holmes	American Express	2	2024-01-01	5555 Jay St.		Brooklyn	NY	11259	USA	500.00	15632.26	0
7	6666666666666666	Arthur Smith	American Express	9	2024-01-01	6666 Jay St.		Brooklyn	NY	11269	USA	500.00	0.00	1
8	7777777777777777	Emma Watson	Bank of America	7	2024-06-15	7777 Jay St.		Los Angeles	CA	90001	USA	12000.00	1000.00	1
9	8888888888888888	Oliver Johnson	Wells Fargo	2	2025-08-01	8888 Main St.		Austin	TX	73301	USA	15000.00	1200.00	1
10	9999999999999999	Sophia Brown	PNC Bank	10	2024-12-01	9999 Oak St.		Chicago	IL	60601	USA	11000.00	500.00	1

Delete statement that deletes all records of credit cards that have balance of 0:

```
Delete
from CREDITCARD
where CreditCardBalance = 0;
```

Results | Messages

	CreditCardNumber	CreditCardOwnerName	CreditCardIssuingCompany	MerchantCode	ExpDate	AddressLine1	AddressLine2	City	StateCode	Zipcode	Country	CreditCardLimit	CreditCardBalance	ActivationStatus
1	1111111111111111	John Williams	American Express	1	2024-01-01	1111 Jay St.		Brooklyn	NY	11209	USA	10000.00	256.21	1
2	1234567890123456	Noah Davis	Citi Bank	8	2025-11-01	1234 Elm St.		San Francisco	CA	94102	USA	9000.00	300.00	1
3	2222222222222222	Samuel Nichols	TD Bank	3	2024-03-20	2222 Jay St.		Trenton	NJ	11229	USA	8000.00	4230.86	1
4	3333333333333333	Silas Decker	American Express	4	2025-05-22	2 Metro Tech Center		Trenton	NJ	11239	USA	10000.00	256.21	1
5	4444444444444444	Michael Hussein	Chase Bank	2	2024-01-01	4444 Jay St.		Manhattan	NY	11249	USA	10000.00	256.21	0
6	5555555555555555	Valerie Holmes	American Express	2	2024-01-01	5555 Jay St.		Brooklyn	NY	11259	USA	500.00	15632.26	0
7	7777777777777777	Emma Watson	Bank of America	7	2024-06-15	7777 Jay St.		Los Angeles	CA	90001	USA	12000.00	1000.00	1
8	8888888888888888	Oliver Johnson	Wells Fargo	2	2025-08-01	8888 Main St.		Austin	TX	73301	USA	15000.00	1200.00	1
9	9999999999999999	Sophia Brown	PNC Bank	10	2024-12-01	9999 Oak St.		Chicago	IL	60601	USA	11000.00	500.00	1

Conclusion

a) Objective

The primary objective of this project was to comprehensively analyze and understand the diverse elements and requirements of EZRental's operations in order to develop a robust database that facilitates efficient data storage and smooth business functioning.

b) Methodology

A combination of Agile and Waterfall methodologies was employed to systematically dissect and address various aspects of the project. Through meticulous planning, analysis, design, development, and validation, a centralized database management system has been established to securely store and manage all requisite data for the seamless operation of EZ-Rental's business.

c) Key Project Milestones

The project was divided into several sprints to ensure effective execution:

SPRINT #1: Creation of a documentation framework to outline the workflow for customers.

SPRINT #2: Development of the EZRental System Management Physical Model Schema.

SPRINT #3: Transformation of the Physical Model for EZRental's DBMS into an actual database using Microsoft SQL Server.

SPRINT #4: Performance of various DML SQL validations on the EZRental DBMS.

d) Outcome

A sophisticated database system has been successfully developed to store, maintain, and present all necessary information in compliance with business requirements, thereby providing a solid foundation for the smooth operation of EZ-Rental's enterprise.