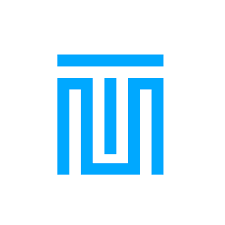
****

**UNIVERSITETI METROPOLITAN TIRANA**

**FAKULTETI I SHKENCAVE KOMPJUTERIKE DHE IT**

**DEPARTAMENTI I INXHINIERISË SOFTUERIKE**

**Program**

Software Engineering

**Year II-Semester I**

**Course: Software Engineering Analysis and Design**

**Project**

**AutoSwitch**

**Worked by:Roksana Ndreca,**

**Frenkli Koleci,**

**Enkeled Selami.**

**Accepted by:** **Dr. Igli Hakrama**

**Github:** **https://github.com/FrenkliK/AutoSwitch**

**AutoSwitch**

**Project Description**

This project is a web platform which would help people to access through our website. A country like Albania where most of the population belongs to middle-class families. Most of them can’t afford a car. We have the solution for you. People have less time and more work than force them to travel different places to work, business meeting and tourism. Also traveling with public transport is not the best choice for all of us, including traffic and the number of people inside. Apart from this feature we will like to make sure everything is automated as we would proceed with a fast manner. We will ensure the buyers the model of the car, license plate, parts of the car and everything else is guaranteed by us. Our website will include people not only companies to rent their car instead of buying a new one which we can help our buyers for this case.

**USER REQUIREMENTS**

The software should generate scheduled reports that we will declare in the front of the website. The software should keep track of how much people we have to serve, the work orders that they are searching for, the model or the type of our cars. The software should provide an easy-to-use interface where the cars can be rent as requested. The software should keep tracking which cars are being rented and which cars are able to be rented.

Development requirements

1. Inventory Management: The rental car company would need to have a system in place for managing its inventory of vehicles, including adding new vehicles, tracking their availability, and updating their status when they are rented or returned.
2. Rental Management: The company would need a system for handling rental transactions, including allowing customers to reserve vehicles, calculating rental fees, and handling payments.
3. Customer Management: The company would need a way to manage customer information, including storing contact information, rental history, and any notes about the customer.
4. Reports and Analytics: The company would need a way to generate reports and analytics to track things like revenue, occupancy rate, and fleet utilization.
5. Integration with other systems: The company could integrate with other systems such as Accounting, Fleet management, GPS tracking and customer relationship management
6. Website and mobile application: A website and mobile application can help customers to make a reservation, check availability, and make payments.
7. Security: The company should ensure all the customer information is stored securely and also ensure that the company's data is protected from malicious attacks.
8. Automation: Automating some of the processes can help to reduce the workload and increase efficiency.

Domain requirements

1. Domain Name: The company would need to register a domain name for its website, such as "[www.examplerentalcar.com](http://www.examplerentalcar.com/)".
2. Web Hosting: The company would need a web hosting service to host its website on a server, so it can be accessed by customers.
3. Content Management System: The company would need a content management system (CMS) to manage and update the content on its website.
4. Database: The company would need a database to store customer information, inventory information, and rental history.
5. Website Design: The company would need a website design that is user-friendly, visually appealing, and responsive to different screen sizes.
6. Search Engine Optimization (SEO): The company would need to optimize its website for search engines so that it can be easily found by potential customers.
7. Security: The company would need to ensure that its website is secure and protected from hacking, data breaches, and other security threats.
8. Integration: The company would need to integrate its website with other systems such as inventory management, payment processing, and customer relationship management.
9. Scalability: The company would need to ensure that the website and systems are scalable, to accommodate the growing number of customers and transactions.
10. Analytics: The company would need to implement web analytics tools to track website traffic, customer behavior, and other key performance indicators.

Ethical requirements

1. Transparent Pricing and Policies: Rental car companies should be transparent about their pricing and policies, clearly disclosing any additional fees or charges, and not engaging in deceptive or misleading advertising.
2. Treating Customers Fairly: Rental car companies should treat all customers fairly, regardless of their race, ethnicity, gender, sexual orientation, religion, or any other protected characteristic.
3. Providing Safe and Reliable Vehicles: Rental car companies have a responsibility to provide safe and reliable vehicles to their customers. This includes ensuring that vehicles are well-maintained, regularly inspected, and meet safety standards.
4. Responsible Marketing: Rental car companies should engage in responsible marketing, avoiding targeting vulnerable populations or promoting unsafe or irresponsible driving behaviors.
5. Environmental Sustainability: Rental car companies should consider the environmental impact of their operations, and take steps to reduce their carbon footprint, such as investing in alternative fuel vehicles or implementing recycling programs.
6. Social responsibility: Rental car companies have a social responsibility to contribute to the community and society, such as by investing in local development projects, supporting charitable causes, or providing employment opportunities.
7. Diversity and Inclusion: Rental car companies should promote diversity and inclusion in the workplace, hiring and promoting employees from diverse backgrounds and creating an inclusive working environment.

Organizational requirements

1. Business Structure: The company would need to be organized as a legal entity, such as a corporation, limited liability company, or partnership, in order to separate personal and business assets.
2. Management Team: The company would need a management team to oversee the day-to-day operations of the business, including hiring and supervising employees, making financial decisions, and developing and implementing business strategies.
3. Employees: The company would need to hire employees to handle various tasks such as customer service, rental transactions, and vehicle maintenance.
4. Financial Management: The company would need to establish financial management systems, such as accounting and budgeting, in order to track expenses, revenues, and profitability.
5. Fleet management: The company would need to establish fleet management systems, such as vehicle maintenance and tracking, in order to keep track of the inventory, utilization and fuel consumption.
6. Marketing and Sales: The company would need to develop a marketing and sales strategy in order to attract and retain customers.
7. Information Technology: The company would need to establish information technology systems, such as a website, inventory management systems, and customer relationship management systems, in order to support the business operations.
8. Regulatory compliance: The company would need to establish systems and procedures to ensure compliance with all relevant laws, regulations and industry standards.
9. Organizational culture: The company would need to establish a positive organizational culture that promotes ethical behavior, collaboration, and employee satisfaction.

Regulatory Requirements

1. Business Licensing: The rental car company would need to obtain any necessary business licenses and permits in order to operate legally. This can include a general business license, as well as any industry-specific licenses required for renting cars.
2. Insurance: The rental car company would need to carry liability insurance to protect against any damages caused by its vehicles while they are being rented.
3. Vehicle Inspection and Maintenance: The company would need to comply with any state or local regulations regarding the inspection and maintenance of its vehicles. This may include regular safety inspections, emissions testing, and maintenance records.
4. Consumer Protection Laws: Rental car companies are subject to consumer protection laws that require them to disclose certain information to customers, such as the terms and conditions of the rental agreement, any additional fees, and liability for damages.
5. Zoning and land-use regulations: The rental car company should comply with local zoning and land-use regulations, ensuring that it operates in an area that is zoned for commercial use and meets local parking and other requirements.
6. Accessibility: The company should comply with accessibility laws for people with disabilities, including regulations for making their website and physical location accessible.
7. Data protection and privacy laws: The company should comply with data protection and privacy laws, ensuring that customer information is collected, stored, and used in compliance with legal requirements.

User interface requirements

1. Ease of Use: The user interface should be easy for customers to navigate and understand, with clear and simple instructions for making a reservation, checking availability, and completing a rental transaction.
2. Responsive design: The user interface should be responsive, adapting to different screen sizes, and devices, allowing customers to make a reservation from any device, such as a desktop, tablet, or mobile phone.
3. Search functionality: The user interface should have a search functionality that allows customers to search for available vehicles by criteria such as location, make and model, price, and rental dates.
4. Vehicle details: The user interface should provide detailed information about each vehicle, including photos, descriptions, and features, to help customers make an informed decision.
5. Availability calendar: The user interface should have an availability calendar, that allows customers to check the availability of vehicles for specific dates, and make a reservation accordingly.
6. Account management: The user interface should allow customers to create an account, view their rental history, and update their personal information.
7. Payment processing: The user interface should provide a secure payment processing system, allowing customers to make payments and deposit with a credit card or other forms of payment.
8. Customer support: The user interface should provide customers with easy access to customer support, such as a contact form or live chat, to answer any questions or concerns they may have.

User story

As a customer of AutoSwitch, I want to be able to easily rent a car for my personal or business needs, so that I can have reliable transportation.

I want to be able to view available cars online, including the make, model, and picture of each car, so that I can make an informed decision about which car I want to rent.

I want to be able to quickly and easily make a reservation, either online or over the phone, by providing my contact information, rental dates, and car preference.

I want to have the option to select various rental packages and add-ons, such as GPS navigation, insurance, or a child seat, to customize my rental experience.

I want to be able to choose from a range of pickup and drop-off locations, including the option to have the car delivered to my location or to pick it up at one of AutoSwitch's many locations.

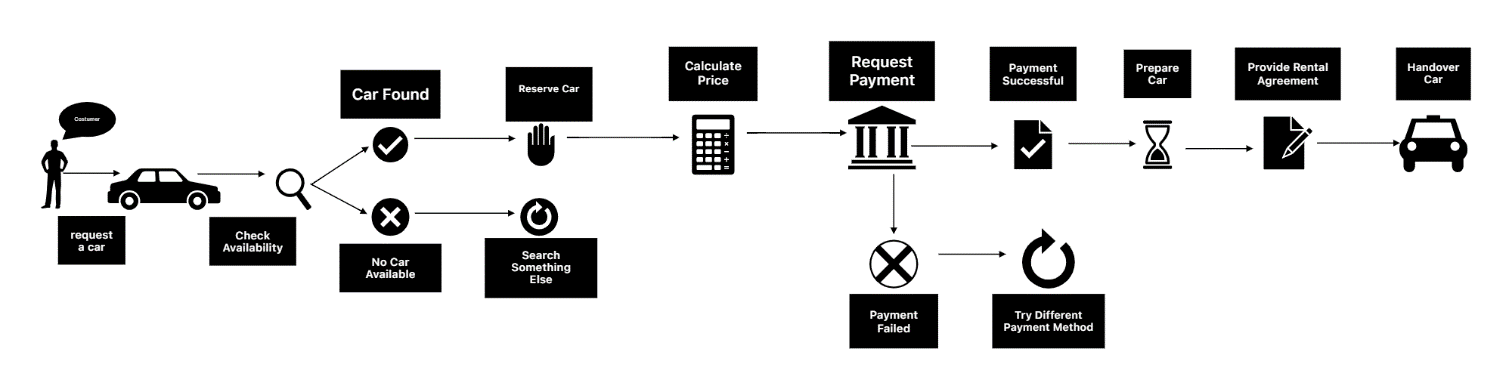
I want to have a smooth and seamless rental experience, including easy pickup and drop-off processes, reliable cars, and excellent customer service.

I want to be able to pay for my rental online or at the time of pickup, using a variety of payment methods, including credit card, debit card, or PayPal.

I want to have the peace of mind of knowing that my personal information is protected and secure, and that my rental transaction is transparent and trustworthy.

Diagrams

BPMN



Where does the process start?

The process starts when a customer makes a reservation through the online platform.

The online platform checks the availability of cars.

If a car is available, the online platform sends a booking request to the selected car rental company.

The car rental company (us) confirms the booking and sends a confirmation to the online platform.

The online platform sends the booking confirmation to the customer.

The customer pays for the rental through the online platform.

The online platform sends the payment to the car rental company.

The customer arrives at the car rental location and picks up the car.

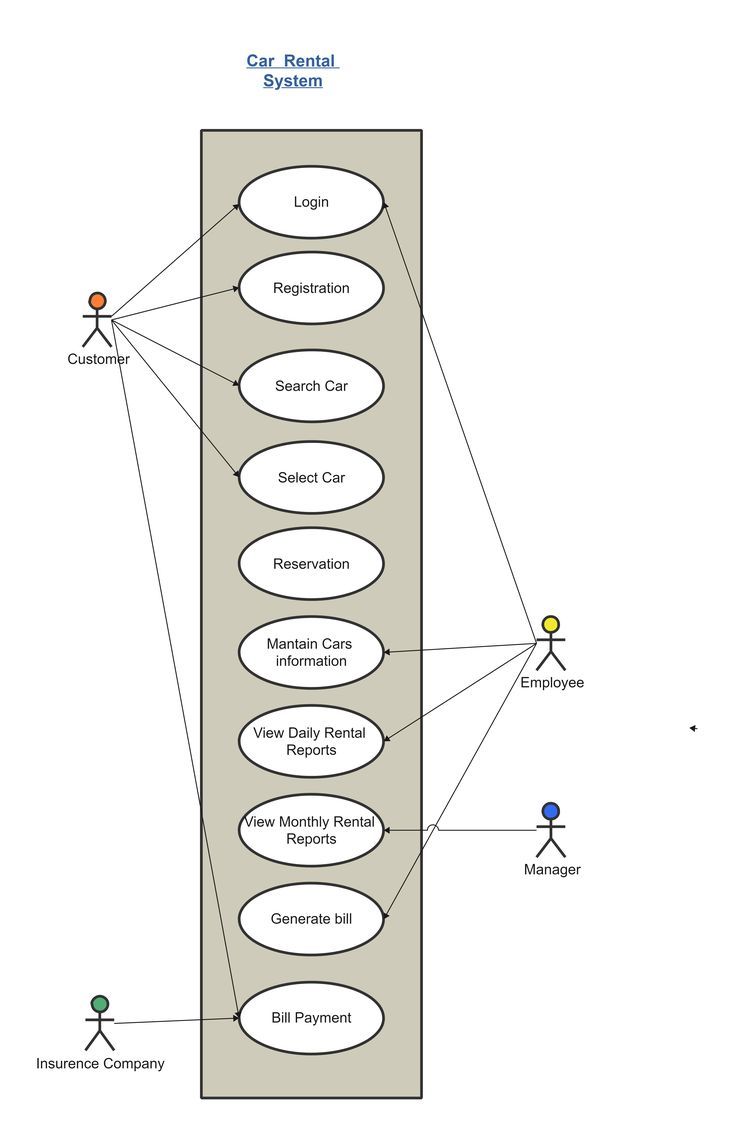
The car rental company checks the car and hands it over to the customer.

The customer returns the car to the car rental company(location agreed) at the end of the rental period.

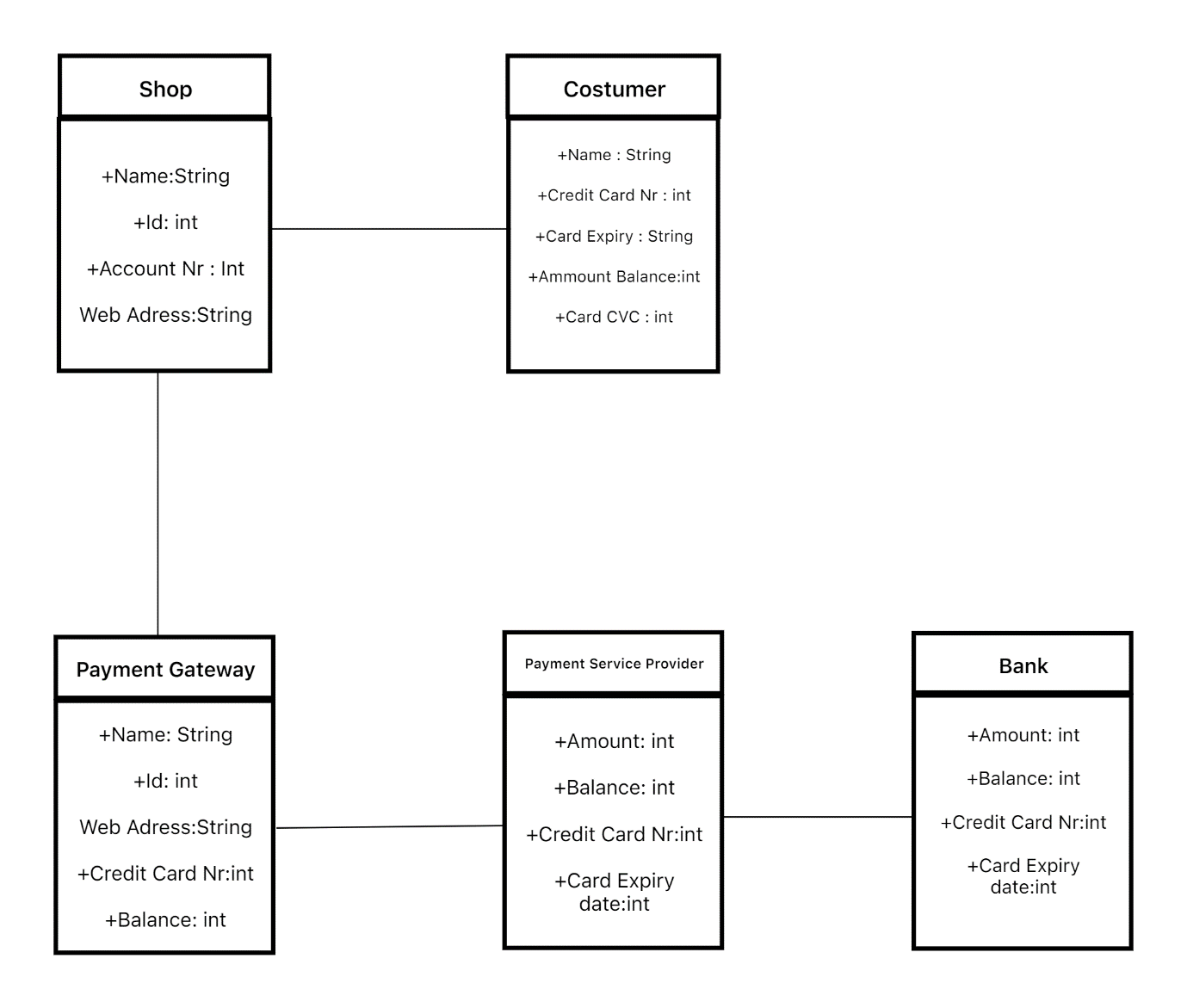
The car rental company checks the car for any damages and charges the customer accordingly.

The process ends when the customer pays for any damages and the car rental company closes the rental period.

**UML**

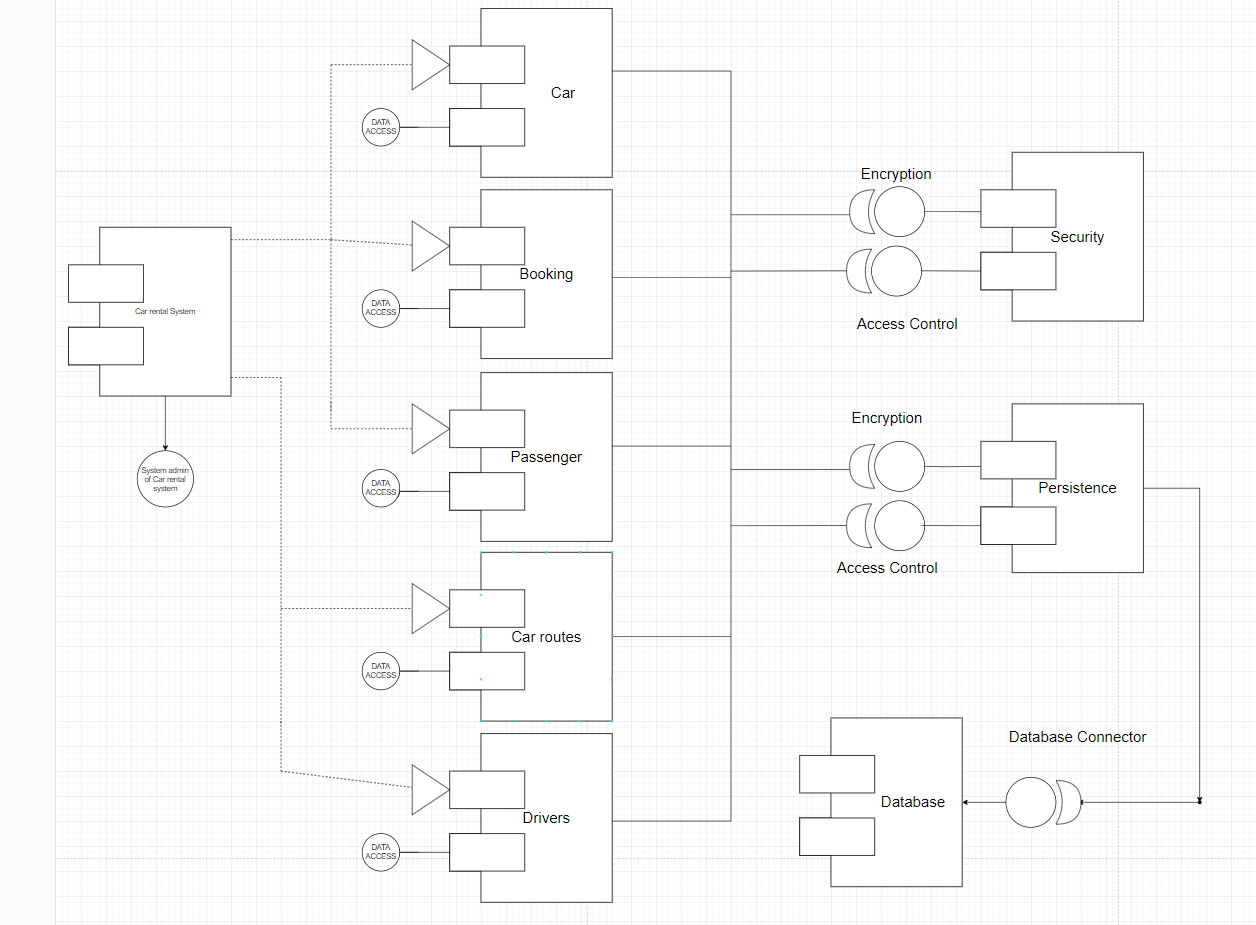
****

**Payment Object**



A payment object diagram for a car rental company would show the relationships between various elements involved in the payment process for car rental services. The diagram would typically include objects such as the customer, rental agreement, payment method, rental charges and receipt. The customer object would have details such as name, contact information and payment method information. The rental agreement object would have details such as the type of car rented, rental period, and rental charges. The payment method object would show different types of payments available such as credit card, debit card and cash. The monthly rent charge breakdown might include base rate x 1 month; taxes x 1 month; and any additional fees or surcharges added at the time of booking. Finally, the receipt object would show a summary of transaction including total amount paid and payment method used

**Component Diagram**

****

The UML component diagram for a car rental system provides a visual representation of the components that make up the system and their relationships with one another. The diagram is created using the Unified Modeling Language (UML) which is a standardized language used for visualizing and documenting software systems.

The car rental component diagram might include components like customer management, car management, reservation management, payment management, and more. These components are represented as rectangular boxes and the relationships between them are depicted through connecting lines. The components in the diagram may have inputs and outputs, which are represented as ports and are used to show how data is passed between the components.

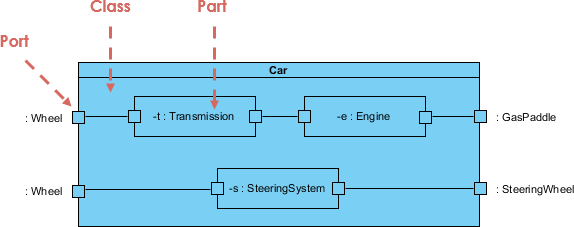
The customer management component is responsible for storing and maintaining customer information, such as name, address, contact information, and more. The car management component is used to manage information about the cars available for rental, such as make, model, year, and current location. The reservation management component is used to manage reservations made by customers, including the dates, the car being rented, and the customer's information. The payment management component is used to process payments made by customers for the rental of a car.

Each component in the diagram interacts with one another through interfaces to perform various tasks such as making a reservation, checking the availability of cars, processing payments, and more. The interfaces between components are depicted as connecting lines that have either a solid or dotted arrowhead indicating the direction of the interaction.

The component diagram helps to provide a clear understanding of the system architecture and the relationships between the components. This makes it easier to maintain and modify the system as needed. It also provides a clear picture of the components that make up the system and their interactions, which is useful when developing the software or making changes to it.

Overall, the UML component diagram for a car rental system is a valuable tool for software developers, stakeholders, and stakeholders, as it provides a clear and concise representation of the system structure, relationships, and interactions between components.

**COMPOSITE STRUCTURE DIAGRAM**



A car rental company composite structure diagram is a type of UML diagram that provides a visual representation of the internal structure of a car rental system. It is used to depict the components and relationships between the various parts of the system, including objects, classes, and interfaces. The diagram is used to help understand the system's architecture, how the various parts work together, and how they can be modified or maintained.

The composite structure diagram for a car rental company typically includes components such as customer management, car management, reservation management, payment management, and more. These components are represented as rectangles in the diagram and their relationships are depicted using connecting lines. The components can also have inputs and outputs, represented as ports, which are used to show how data is passed between the components.

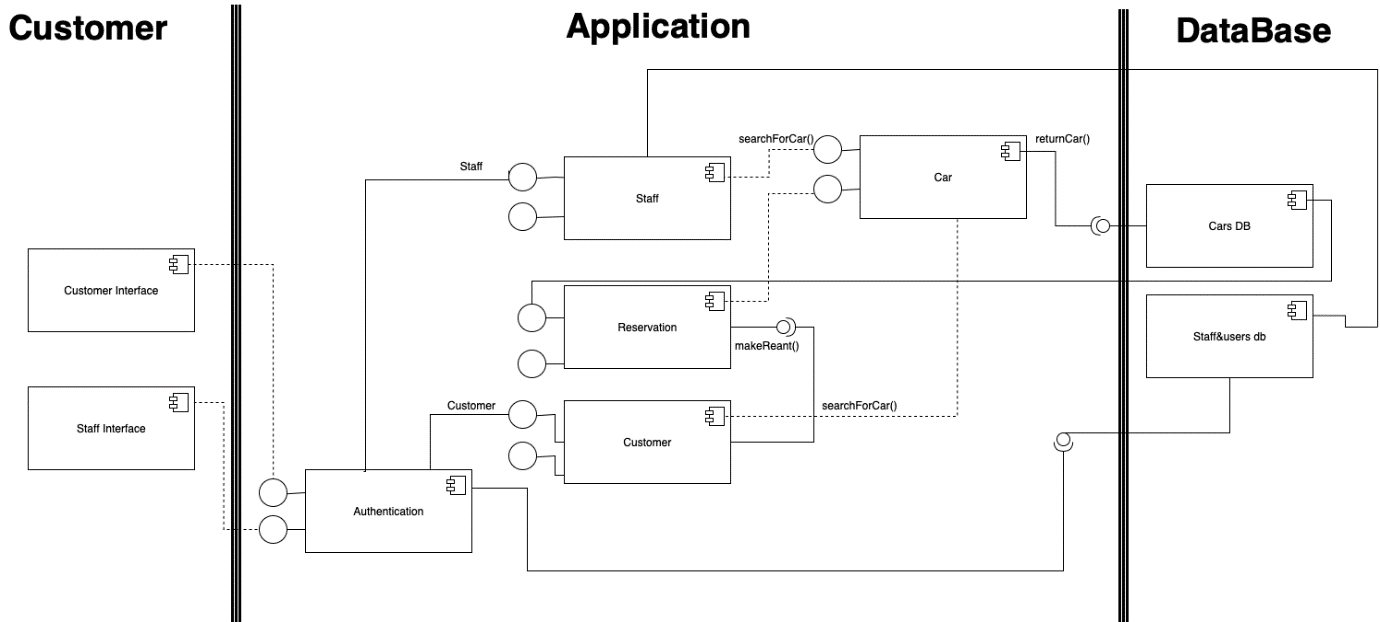
The customer management component is responsible for storing and managing customer information, such as name, address, and contact information. The car management component is used to store and manage information about the cars available for rental, such as make, model, year, and current location. The reservation management component is used to manage customer reservations, including the dates, the car being rented, and the customer's information. The payment management component is used to process payments made by customers for the rental of a car.

The components in the diagram interact with one another through interfaces, which are depicted as connecting lines with either solid or dotted arrowheads indicating the direction of the interaction. The interfaces allow the components to work together to perform various tasks, such as making a reservation, checking the availability of cars, processing payments, and more.

The composite structure diagram provides a clear understanding of the system's architecture, making it easier to maintain and modify the system as needed. It also provides a clear picture of the components that make up the system and their interactions, which is useful when developing the software or making changes to it.

Overall, the composite structure diagram is a valuable tool for software developers, stakeholders, and customers as it provides a clear and concise representation of the internal structure and workings of the car rental system. It helps to ensure that the system is developed and maintained efficiently and effectively, resulting in a high-quality service for customers.

**DEPLOYMENT DIAGRAM**



A deployment diagram in a car rental company is a type of UML diagram that provides a visual representation of the physical components of the system and how they are deployed in a specific environment. The diagram is used to show the relationship between the physical hardware components, such as servers, computers, and mobile devices, and the software components that run on them.

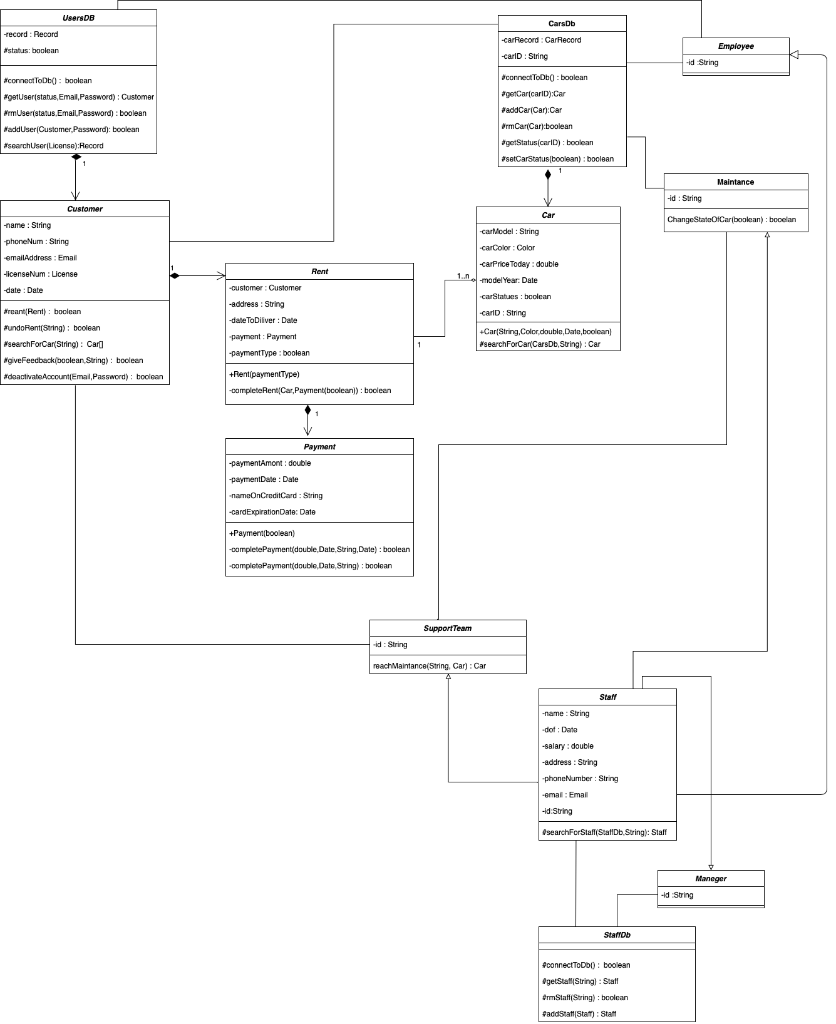
The deployment diagram for a car rental company typically includes components such as web servers, databases, and clients, including desktop computers and mobile devices. These components are represented as nodes in the diagram and their relationships are depicted using connecting lines. The nodes can also have inputs and outputs, represented as ports, which are used to show how data is passed between the components.

The web servers in the diagram are responsible for serving the web pages and applications used by customers to make reservations and manage their account information. The databases are used to store customer information, car information, and reservation information. The clients, including desktop computers and mobile devices, are used by customers to access the web pages and applications provided by the web servers.

The deployment diagram provides a clear understanding of the physical deployment of the components in the system, making it easier to understand how the system functions in the real world. It also provides a clear picture of the relationship between the hardware and software components, making it easier to maintain and modify the system as needed.

Overall, the deployment diagram is a valuable tool for software developers, stakeholders, and customers as it provides a clear and concise representation of the physical deployment of the car rental system. It helps to ensure that the system is deployed effectively, providing a high-quality service for customers.

**OBJECT DIAGRAM**



An object diagram in a car rental company is a type of UML diagram that provides a visual representation of the objects and their relationships in the system at a specific moment in time. The diagram is used to show the state of the objects, including their attributes and their relationships with other objects, at a specific point in time.

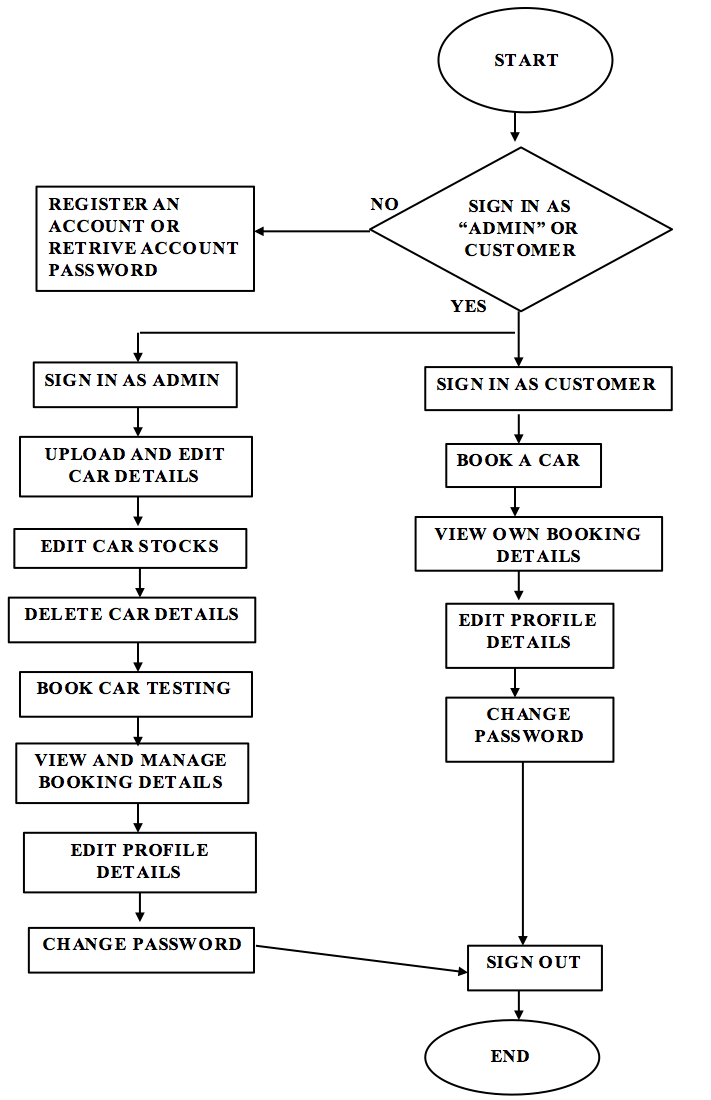
The object diagram for a car rental company typically includes objects such as customers, cars, reservations, and payments. These objects are represented as rectangles in the diagram and their relationships are depicted using connecting lines. The objects can also have attributes, which are represented as labels inside the rectangles.

The customer object in the diagram represents a specific customer, including their personal information, such as name, address, and contact information. The car object represents a specific car, including its make, model, year, and current location. The reservation object represents a specific reservation made by a customer, including the dates, the car being rented, and the customer's information. The payment object represents a specific payment made by a customer for the rental of a car.

The object diagram provides a clear understanding of the state of the objects in the system at a specific moment in time. This information is useful for debugging, testing, and other activities related to maintaining and modifying the system.

Overall, the object diagram is a valuable tool for software developers and stakeholders as it provides a clear and concise representation of the objects and their relationships in the car rental system at a specific moment in time. It helps to ensure that the system is functioning correctly and efficiently, resulting in a high-quality service for customers.

**PACKAGE DIAGRAM**



A profile diagram in a car rental company is a type of UML diagram that provides a visual representation of the extension of the UML metamodel. The diagram is used to show the customizations made to the UML metamodel to better represent the specific needs of the car rental system.

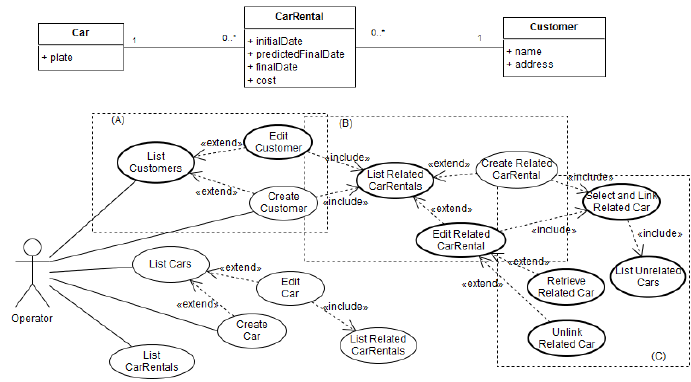
The profile diagram for a car rental company typically includes custom stereotypes, such as "Rental Car" and "Rental Agreement," which are used to extend the UML metamodel. These custom stereotypes are represented as icons in the diagram and are used to describe the specific behavior and characteristics of the objects in the system.

The "Rental Car" stereotype, for example, is used to represent a specific car that is available for rent. The stereotype includes attributes such as make, model, year, and location, which are used to describe the specific characteristics of the car. The "Rental Agreement" stereotype, on the other hand, is used to represent a specific agreement made between the customer and the car rental company. The stereotype includes attributes such as the dates of the rental, the car being rented, and the customer's information.

The profile diagram provides a clear understanding of the customizations made to the UML metamodel to better represent the specific needs of the car rental system. This information is useful for software developers and stakeholders as it provides a clear and concise representation of the customizations made to the UML metamodel, making it easier to understand how the system functions.

Overall, the profile diagram is a valuable tool for software developers and stakeholders as it provides a clear and concise representation of the customizations made to the UML metamodel for the car rental system. It helps to ensure that the system is functioning correctly and efficiently, resulting in a high-quality service for customers.

**PARTIAL DOMAIN AND USE CASE MODELS**



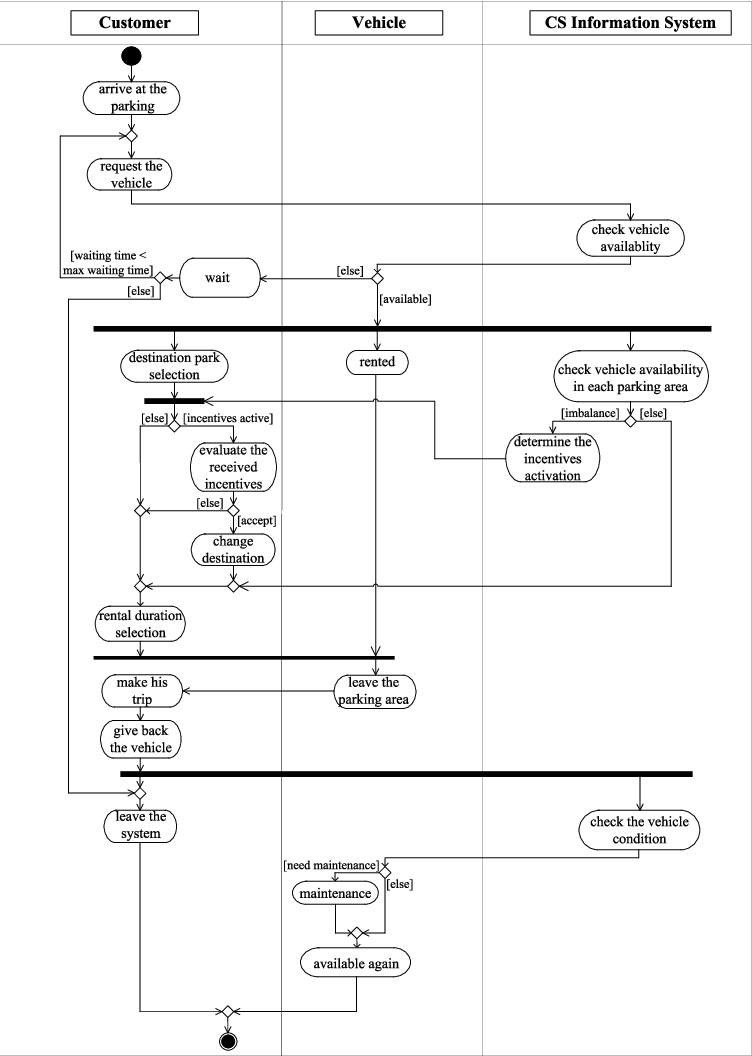
The partial domain and use case models for a car rental company are diagrams used in software development to represent the problem domain of the system and the functional requirements of the system, respectively. These diagrams are used to help software developers and stakeholders understand the system requirements and ensure that the system is being developed to meet the needs of the business.

The partial domain model for a car rental company typically includes objects such as customers, cars, and reservations. The objects in the diagram are used to represent the real-world entities in the system and their relationships with each other. The relationships between the objects are depicted using connecting lines, showing how the objects interact with each other.

The use case model for a car rental company represents the functional requirements of the system, detailing what the system must do to meet the needs of the business. Use cases are depicted as ovals in the diagram and they represent the specific actions or processes that the system must perform. These use cases are usually organized into categories, such as customer-related use cases, car-related use cases, and reservation-related use cases.

The use case model provides a clear understanding of the functional requirements of the system and helps to ensure that the system is being developed to meet the needs of the business. This information is useful for software developers and stakeholders as it provides a clear and concise representation of the functional requirements of the system, making it easier to understand how the system functions.

**ACTIVITY DIAGRAM**

****

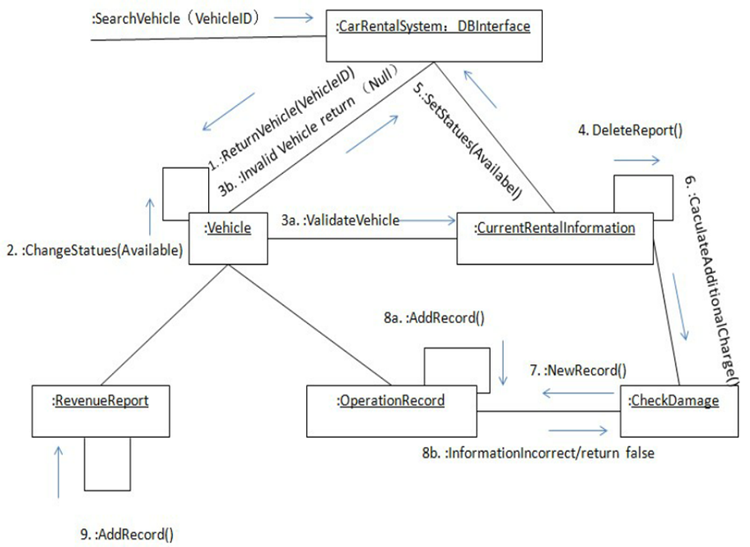
An activity diagram for renting a car is a flowchart-style diagram used to model the activities that take place during the rental process. The diagram shows the flow of control and the interactions between the customer and the car rental system.

One common problem with activity diagrams for renting a car is that they can become complex and difficult to understand if the diagram includes too many steps or details. This can make it difficult for stakeholders and software developers to understand the flow of control in the system and to identify potential issues.

Another problem is that the activity diagram may not accurately reflect the real-world rental process, resulting in the development of a system that does not meet the needs of the business. For example, if the diagram does not include important steps such as verifying the customer's driver's license or calculating the rental fee, the resulting system may not be able to handle these tasks correctly.

A third problem with activity diagrams for renting a car is that they may not be updated to reflect changes in the rental process. For example, if the car rental company introduces a new policy or changes the way it calculates rental fees, the activity diagram may not reflect these changes, leading to confusion and errors in the system.

**COMMUNICATION DIAGRAM**

****

A communication diagram, also known as a collaboration diagram, is a type of UML (Unified Modeling Language) diagram used to model the interactions between objects or components in a system. In the context of a car rental company, a communication diagram can be used to model the interactions between the customer, the car rental system, and the various objects and components involved in the rental process.

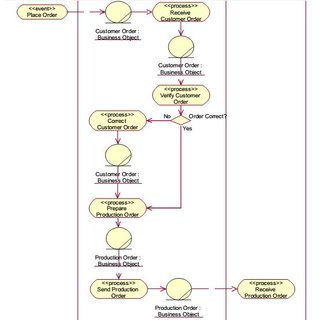
The communication diagram is a more dynamic representation of the system than other UML diagrams, as it shows the flow of control and the interactions between objects and components in real-time. The diagram is comprised of objects represented as rectangles, connected by lines that represent the interactions between the objects.

In the case of a car rental company, the objects in the communication diagram could include the customer, the car rental system, the reservation system, the billing system, and the cars available for rent. The interactions between these objects could include actions such as the customer making a reservation, the reservation system confirming the reservation, the billing system calculating the rental fee, and the car rental system allowing the customer to pick up the car.

The communication diagram provides a clear and concise representation of the interactions between objects and components in the system, making it easier for software developers and stakeholders to understand the flow of control in the rental process. This information can be used to identify potential issues in the system and to ensure that the system is being developed to meet the needs of the business.

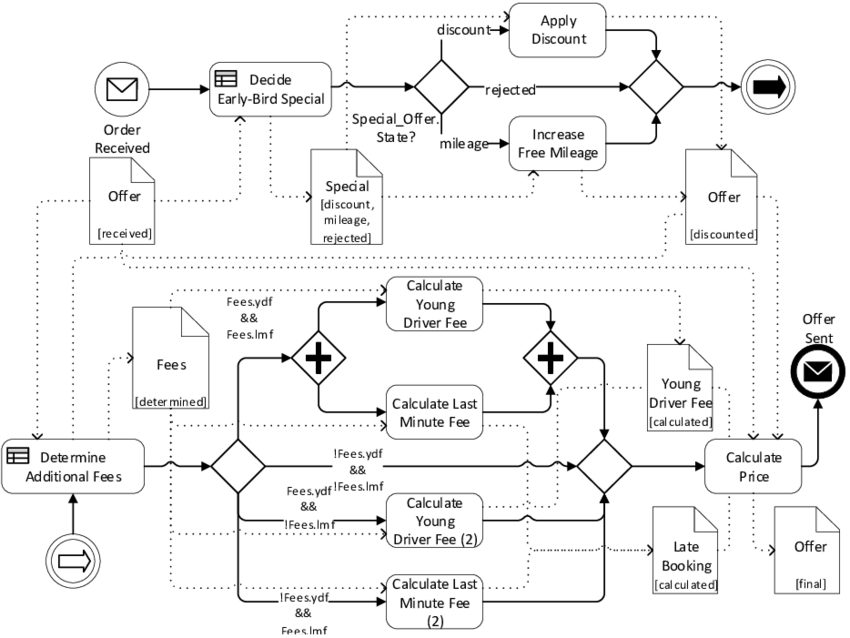
Overall, the communication diagram is a valuable tool for modeling the interactions between objects and components in a car rental system. By providing a clear and concise representation of the flow of control and interactions, the communication diagram helps to ensure that the system is being developed to meet the needs of the business and to provide a high-quality service for customers.

**STATE DIAGRAM**



The state diagram provides a clear and concise representation of the behavior of the car during the rental process, making it easier for software developers and stakeholders to understand the flow of control in the system. This information can be used to identify potential issues in the system and to ensure that the system is being developed to meet the needs of the business.

Additionally, the state diagram can be used to model the behavior of other objects or components in the system, such as the reservation system or the billing system. This can provide a more comprehensive understanding of the system and can help to identify any potential issues or areas for improvement.

**BPMN CASE**

BPMN (Business Process Model and Notation) is a widely adopted notation for modeling business processes. In the context of a car rental company, a BPMN diagram can be used to model the various steps involved in the rental process from start to finish.

The BPMN diagram is comprised of various graphical elements that represent the different activities and events in the process. The elements include tasks, events, gateways, and flows. Tasks represent activities that need to be performed in the process, such as making a reservation or picking up the car. Events represent significant points in the process, such as the customer making a payment. Gateways are used to control the flow of the process, allowing for decisions to be made based on certain conditions. Flows are used to connect the elements in the diagram, showing the order in which the activities are performed.

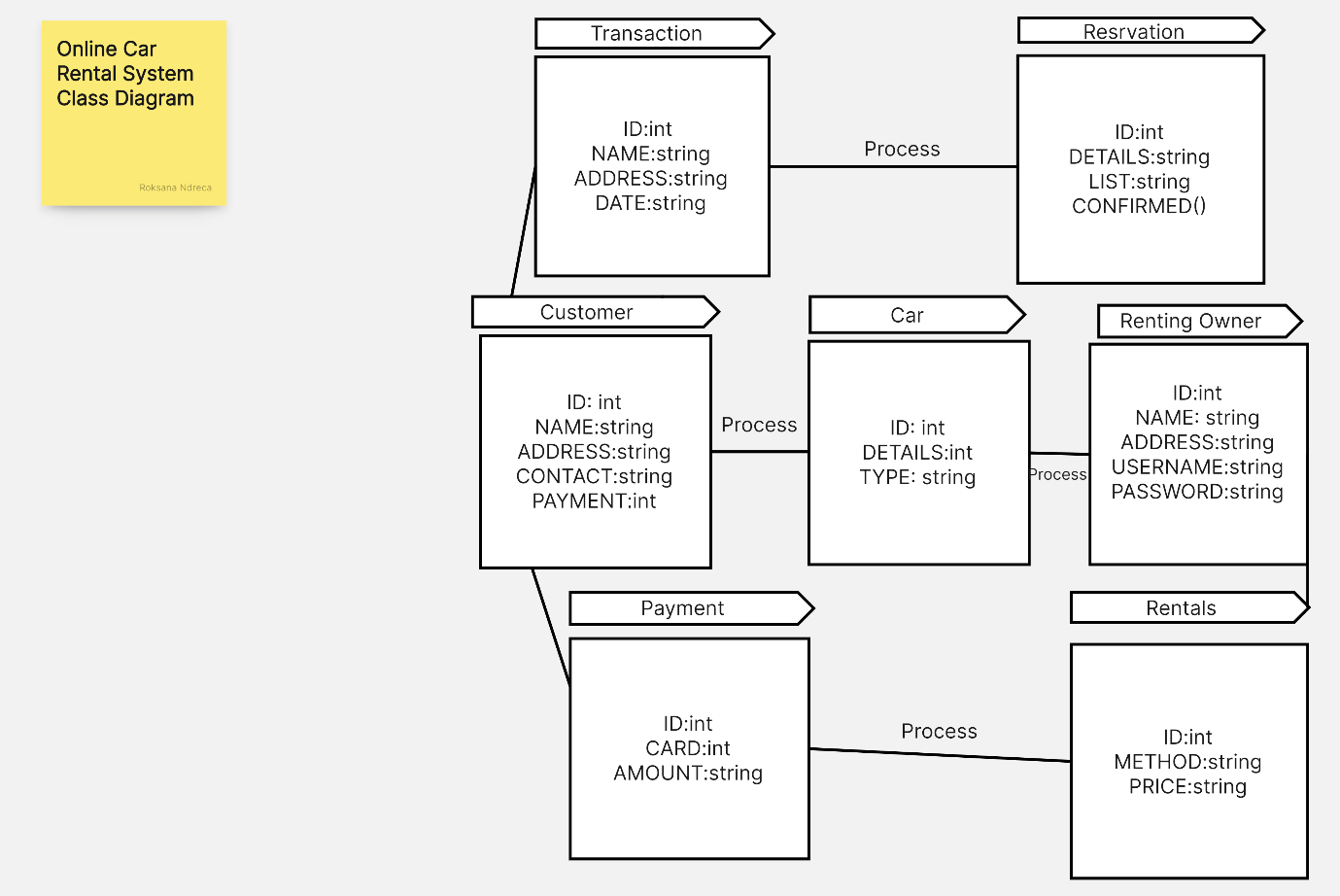
In the case of a car rental company, the BPMN diagram could include steps such as the customer making a reservation, the reservation being confirmed, the customer picking up the car, and the customer returning the car. The diagram could also include decision points, such as the car rental system checking the availability of the car, and events such as the customer making a payment.

The BPMN diagram provides a clear and concise representation of the rental process, making it easier for software developers and stakeholders to understand the flow of control in the system. This information can be used to identify potential issues in the system and to ensure that the system is being developed to meet the needs of the business.

Additionally, the BPMN diagram can be used as a communication tool between business stakeholders and software developers, helping to ensure that the system is being developed in line with the business requirements.

Overall, the BPMN diagram is a valuable tool for modeling the rental process in a car rental company. By providing a clear and concise representation of the steps involved in the process, the BPMN diagram helps to ensure that the system is being developed to meet the needs of the business and to provide a high-quality service for customers.

**Class diagram**

****

A class diagram for a car rental company would depict the classes, attributes, and relationships of the entities involved in the car rental system. The classes would include:

Car: This class would include attributes such as the car make, model, year, registration number, and rental status (available/unavailable).

Customer: This class would include information about the customer such as name, contact details, and rental history.

Rental Agreement: This class would describe the details of a rental transaction including the date of rental, date of return, rental fee, and payment details.

Employee: This class would include information about employees, including their name, role, and contact details.

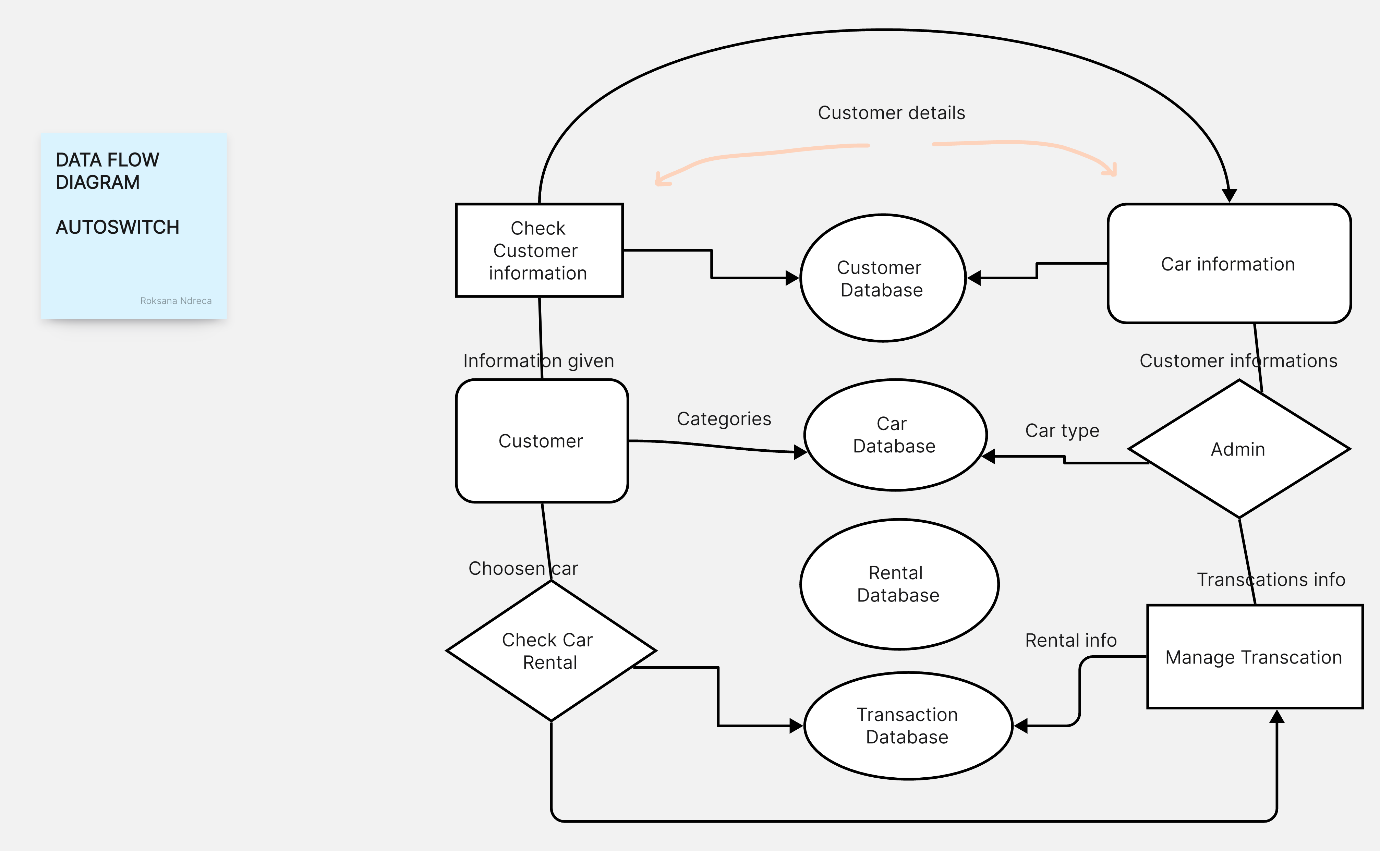
The relationships between these classes would include:

A Car can have many Rental Agreements, while a Rental Agreement is associated with only one Car.

A Customer can have many Rental Agreements, while a Rental Agreement is associated with only one Customer.

An Employee can be involved in many Rental Agreements, either as the person who rents the car to the customer or as the person who receives the car back from the customer.

**Data Flow**

****

A data flow diagram (DFD) for a car rental company would represent the flow of data and information within the organization. It would show the processes, inputs, outputs, and entities involved in the car rental system.

The entities involved in the system could include:

Customer: Represented as an external entity that requests car rental services from the company.

Car Rental Company: The main process that carries out the rental transactions and manages the cars and customer data.

Car Inventory: Represented as a data store that contains information about the available cars and their rental status.

Customer Database: Represented as a data store that contains information about the customers, including their rental history.

The processes within the system could include:

Request Car Rental: The customer inputs a request for car rental, including the desired rental period and car type.

Validate Request: The Car Rental Company validates the request by checking the availability of the desired car in the Car Inventory.

Approve Request: If the requested car is available, the Car Rental Company approves the request and updates the Car Inventory to mark the car as rented.

Update Customer Database: The Car Rental Company updates the customer's rental history in the Customer Database.

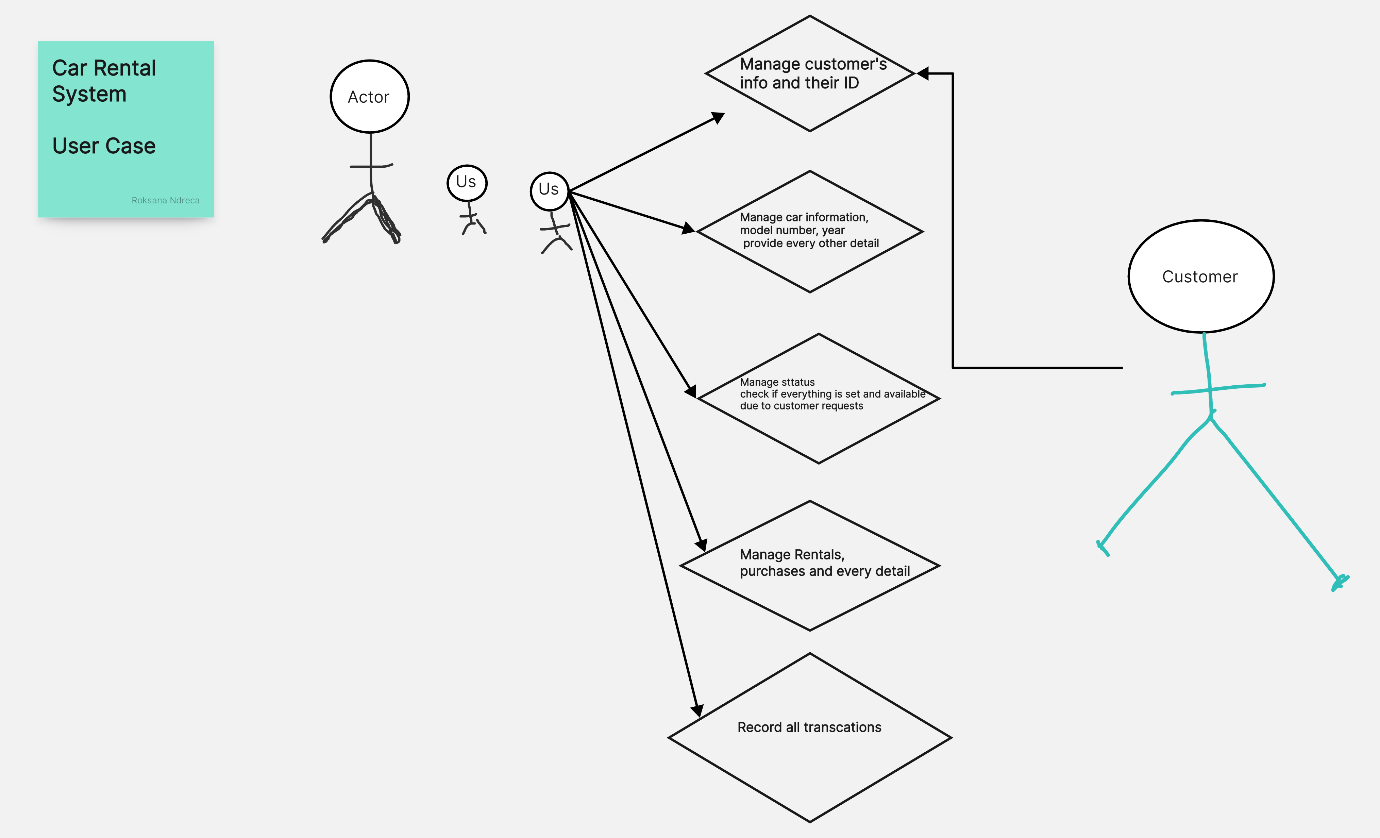
Return Car: The customer returns the rented car to the Car Rental Company, and the rental agreement is terminated.

Update Car Inventory: The Car Rental Company updates the Car Inventory to mark the car as available for rent.

Generate Invoice: The Car Rental Company generates an invoice for the customer, based on the rental agreement details.

The DFD would also show the inputs and outputs of each process, and how the data flows between the entities and processes in the system.

**User Case**

****

A user case diagram for a car rental company would depict the interactions between the system and the different types of users who interact with it. The diagram would consist of actors and use cases, which represent the different types of interactions that occur in the system.

The actors in the system could include:

Customer: Represented as a person who rents a car from the car rental company.

Employee: Represented as a person who works for the car rental company and is responsible for managing the rental transactions.

The use cases in the system could include:

Rent Car: Represented as a process in which the customer selects a car, reserves it, and rents it for a specified period of time.

Return Car: Represented as a process in which the customer returns the rented car to the car rental company.

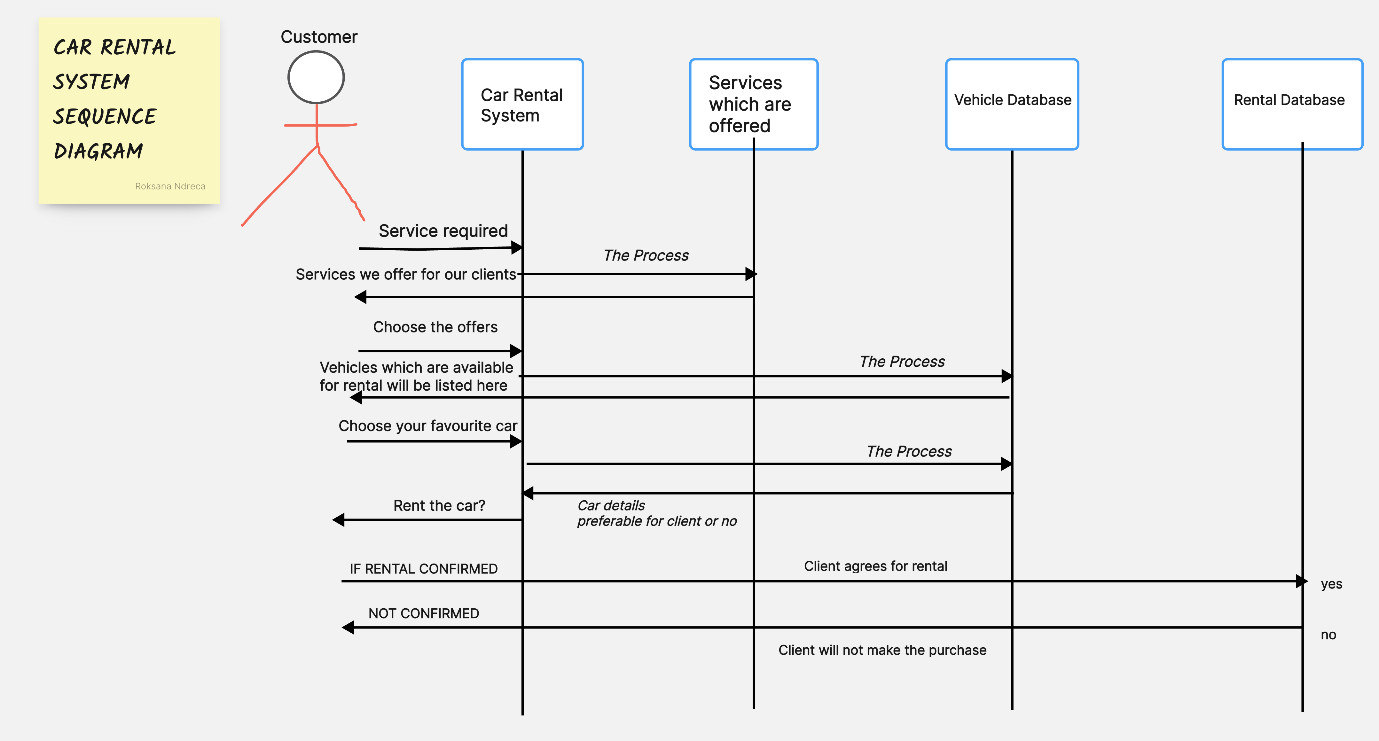
Manage Inventory: Represented as a process in which the employee manages the car inventory, including adding new cars, updating information about existing cars, and marking cars as available or unavailable.

Manage Customer Database: Represented as a process in which the employee manages the customer database, including adding new customers, updating information about existing customers, and viewing customer rental history.

Generate Invoice: Represented as a process in which the system generates an invoice for the customer, based on the rental agreement details.

The user case diagram would show the interactions between the actors and the use cases, and would provide a high-level view of the different types of interactions that occur in the system.

**System Sequence**

****

A system sequence diagram (SSD) for a car rental company would depict the interactions between the system and the actors involved in a specific scenario or use case. The diagram would show the sequence of messages exchanged between the actors, and the order in which they occur.

For example, a SSD for the use case "Rent Car" could include the following actors and interactions:

Customer: Sends a request to rent a car to the Car Rental Company system.

Car Rental Company: Validates the customer's request by checking the availability of the desired car in the Car Inventory.

Car Rental Company: Approves the customer's request if the car is available, and updates the Car Inventory to mark the car as rented.

Car Rental Company: Sends a confirmation message to the customer, including the rental agreement details.

Customer: Sends a message indicating that they have received the confirmation and agree to the rental agreement.

Car Rental Company: Updates the customer's rental history in the Customer Database.

The SSD would provide a detailed view of the interactions between the system and the customer, and would help to understand the steps involved in the process of renting a car from the car rental company.