

Project Sponsor: Business Owner of Car Dealership

Business Need:

A car dealership wants to improve its processes; this project is to build a platform to help facilitate operations and simplify transactions to get the desired results. Business processes include matching customers with vehicles, tracking sales, inventory management, determining salesperson's commissions, and promotional offers.

Business Requirement:

1. Matching customers with vehicles online via the inventory search function
On-site via salesperson searching the internal and external inventories.
2. To supervise the sale process and trace vehicle options, sales tax, discounts, and financing options.
3. Inventory management involves buying and selling.
4. Salesperson commission is performed every week and is based on sales percentage.
5. Promotional offers refer to the pricing of internal inventory and are performed weekly.

Business Value:

We anticipate the new integrated.

platform to increase sales and provide the best customer satisfaction for all clients. In the long run, we expect to be the most trustable and dependable car dealership in the area.

Architectural Design:

Service-Oriented Architecture (SOA):

Service-Oriented Architecture (SOA) is an architectural approach used in software design and development.

What are the advantages of SOA:

Scalability:

Promotes scalability by allowing services to be deployed and scaled independently. As demand increases, specific services can be scaled horizontally or vertically without affecting the entire system.

Loose Coupling:

Services are designed to be loosely coupled. This means that each service operates independently of the internal details of other services. That makes it easier to modify or replace one service without affecting others.

Enhanced Maintenance and Updates:

Simplifies maintenance tasks and updates. If a service needs to be modified or replaced, it can be done without impacting the entire system.

Technology Neutrality:

meaning that services can be implemented using different technologies and programming languages.

Modularity and Reusability:

This promotes code reuse, as services can be developed independently and then reused in different applications. This modular approach simplifies development and maintenance tasks.

Cost-Efficiency:

Can save cost and time by reducing redundancies.

The Waterfall Model (A predictive SDLC):

Is a linear and sequential approach to software development. It's characterized by distinct phases, each of which must be completed before progressing to the next.

What is the advantage of the waterfall method?

Presence of a Clear Structure.

The sequential nature of the model provides a clear and structured approach, making it easier to understand and manage the project's progress.

Well-Defined Requirements:

It requires detailed documentation of requirements at the beginning, which can help in clearly understanding and meeting the client's needs.

Predictable timeline:

The time and resource requirements for each phase can be estimated upfront, leading to a more predictable project schedule.

The Object-Oriented Approach

The object-oriented approach to systems. A class is a category or classification used to describe a collection of objects. Each object belongs to a class, a natural way to develop requirements.

The object-oriented approach provides UML models that can document dialog designs, including sequence diagrams, activity diagrams, and class diagrams.

The Object-oriented approach was developed to solve the problem of creating graphical user interfaces (GUIs) that involved such “objects” as pull-down menus, buttons, check boxes, and dialog boxes. More recent object-oriented languages include C++, Java, and C#.

User requirements

1. system must be capable of identifying human input and reacting to it.
2. Data should be able to be stored and organized by the system.
3. system must be capable of responding to user input.
4. It should be possible for the system to communicate with other systems.
5. Multiple users should be supported by the system.
6. Users should be able to easily search and filter the inventory.
7. Website and Mobile App: The dealership should have a user-friendly website and mobile app.

System Requirements

1. site is free of problems when looking for a particular item.
2. At the day's end, the system closes all transactions.
3. Each car must have a file, and each user must have an account.
4. When there are multiple vehicles of the same type available, separate files must be created for each one.
5. The price set by the administration's rules must be paid by the customer.
6. Analytics and Reporting Tools: Generates reports on sales performance, inventory turnover and service efficiency for strategic decision-making.
7. Inventory Management: add – edit – delete cars.

Non-Functional requirements

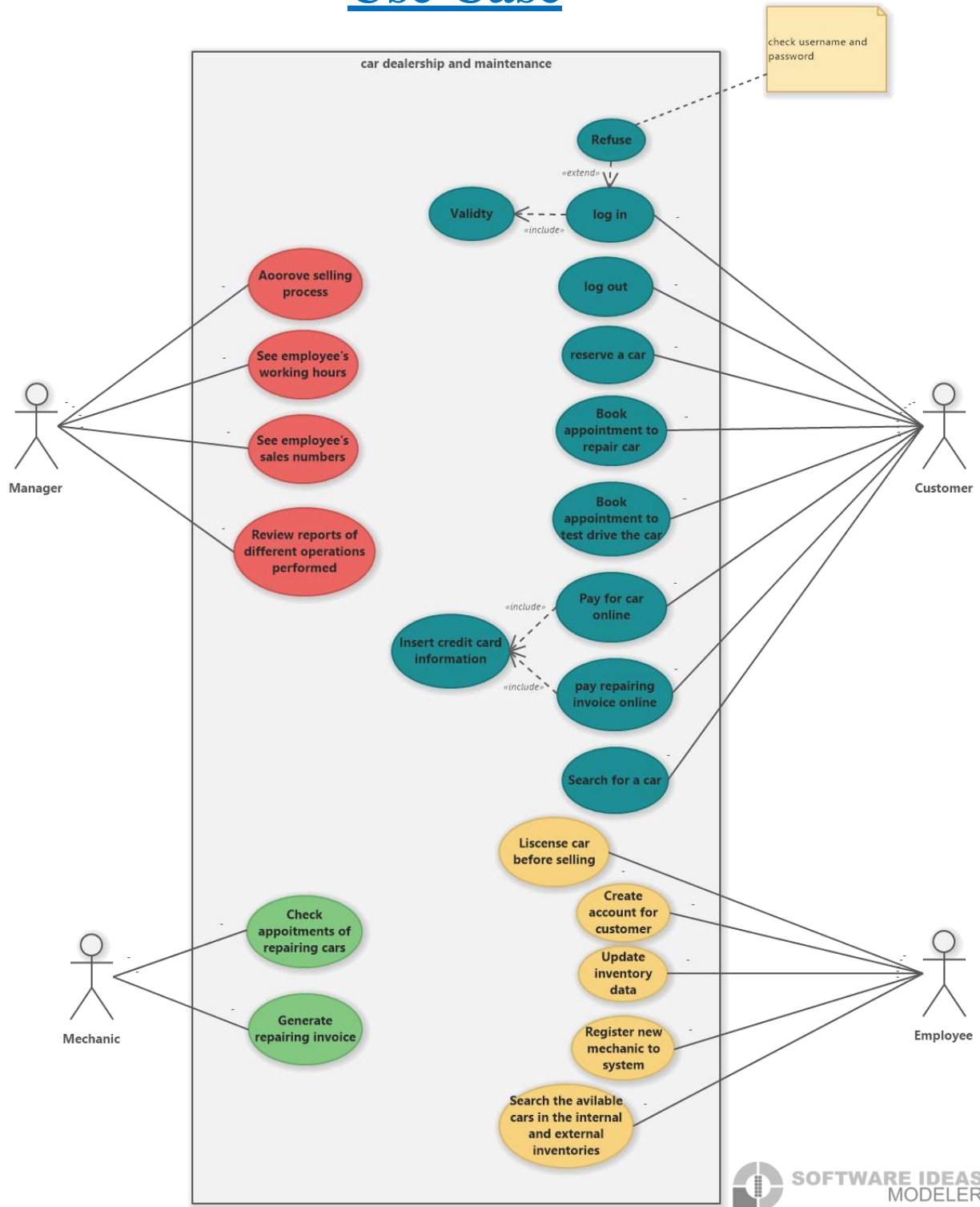
1. Use a web browser or mobile application to access the system.
2. Responsiveness: System should respond quickly to user actions.
3. putting usability first when getting orders from the user.
4. site is free of problems when looking for a particular item.
5. Integration and Scalability: Systems should be scalable to accommodate business growth and integrate seamlessly for efficient data flow across departments.
6. Data Security: protect sensitive customer and financial data.

Functional requirements

1. Each member must log in or out of the system, through the username and password, after verifying the validity of data, and in the event that any of them (name, password) is entered wrong, access to the system will be refused.
2. The customer can reserve the car.
3. The customer can search for a car.
4. The customer can book an appointment to repair the car.
5. The customer can book an appointment to test drive the car before buy it.
6. The customer can pay for a car online, after inserting credit card info.
7. The customer can pay the invoice online, after inserting credit card info.
8. The employee must license the cars before selling them.

9. The employee must create an account for the customer, after checking their data.
 10. The employee must register a new mechanic to the system.
 11. The employee can search for the available cars and spare parts in inventories.
 12. The employee update inventory data.
 13. The manager must approve the selling process.
 14. The manager can see the employees' working hours.
 15. The manager can see the employees' sales numbers.
 16. The manager can review the reports on the different operations performed.
 17. The mechanic will be able to check the appointments for repairing cars.
- The mechanic will be able to generate a repair invoice, after repairing the car.

Use Case



Use Case Description

(The customer log-in to the system)

Element	Description
Use Case Name	Log-in
Use Case ID	UC-1
Priority	High
Actor(s)	Customer
Description	The customer log-in to the system.
Precondition(s)	The customer should land on the log-in page.
Post-condition(s)	The customer logged to the system.
Flow of events	<p>1-The customer landed on the log-in page.</p> <p>2-The system displays a login form.</p> <p>3-The customer enters the valid username and password</p> <p>4-The system authenticates the username and password.</p> <p>5-The system display greets message.</p> <p>6-Customer logged in to the system.</p>
Alternative flows and exceptions	<p>1. Customer entered invalid username and password at the second trial.</p> <p>2. The system validates the username and password and wrong username and password are verified. System prompt the user to enter username and password again.</p> <p>3. The user enters username and password again.</p> <p>4. The system validates username and password, and wrong username and password are entered. A number of trials have exceeded 3 times.</p> <p>5. System prompt user to log-in again after 15 minutes.</p>

(Register a new mechanic)

Element	Description
Use Case Name	Register
Use Case ID	UC-09
Priority	High
Actor(s)	Employee
Description	Register mechanic to the system.
Precondition(s)	The form page is active to registration.
Post-condition(s)	The mechanic is registered with the system and profile data is stored with the system.
Flow of events	<ol style="list-style-type: none">1. Employee select registration operation.2. System display registration form and prompt to fill in profile data.3. The employee fills up the form with profile data.4. The system validates new mechanic information5. The system reverts with registration station.
Alternative flows	<ol style="list-style-type: none">1. New mechanic registration information is not valid.2. The system displays registration failure page.
Exceptions	At any time, the operation may abandon.

(The employee can create an account for the customer)

Element	Description
Use Case Name	Create account
Use Case ID	UC-08
Priority	High
Actor(s)	Employee
Description	Create a new account for customer
Precondition(s)	The form page is active to registration.
Post-condition(s)	A new account is created for customer and profile data is stored with the system.
Flow of events	<ol style="list-style-type: none">1. Employee select create new customer account operation.2. System display registration form and prompt to fill in profile data.3. The employee fills up the form with profile data.4. The system validates new customer information5. The system saves customer data
Alternative flows	<ol style="list-style-type: none">1- New customer information is not valid.2- The system displays account creation failure page.
Exceptions	At any time, the operation may abandon.

(The manager can review the reports)

Element		Description
Use Case Name		Review report
Use Case ID		UC-14
Priority		High
Actor(s)		Manager
Description		The manager can review the reports of the different operations performed
Precondition(s)		The reports are in reports section waiting to be opened by manger
Post-condition(s)		The reports are viewed by the manager and saved in archive
Flow of events		<div>1- The manger clicks on “view reports”</div> <div>2- The system recalls all the reports stored in reports section</div> <div>3- The manager selects the reports of specific operation</div> <div>4- The system views the reports to the manager</div> <div>5- The manager selects . “Done reading”</div> <div>6- The system transfer the report to the archive</div>
Alternative flows and exceptions		At any time, the operation may abandon

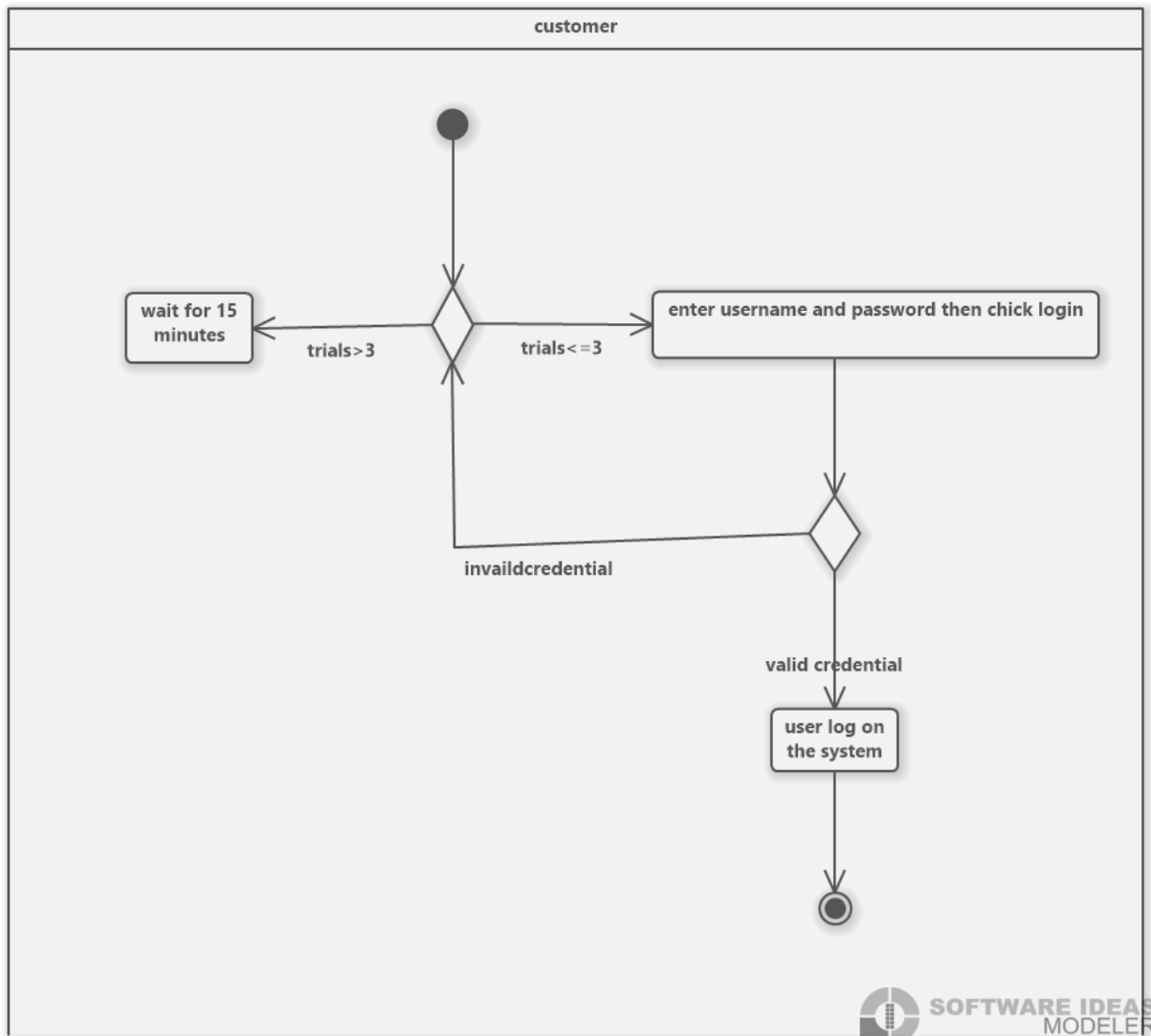
(The customer can book an appointment to repair the car):

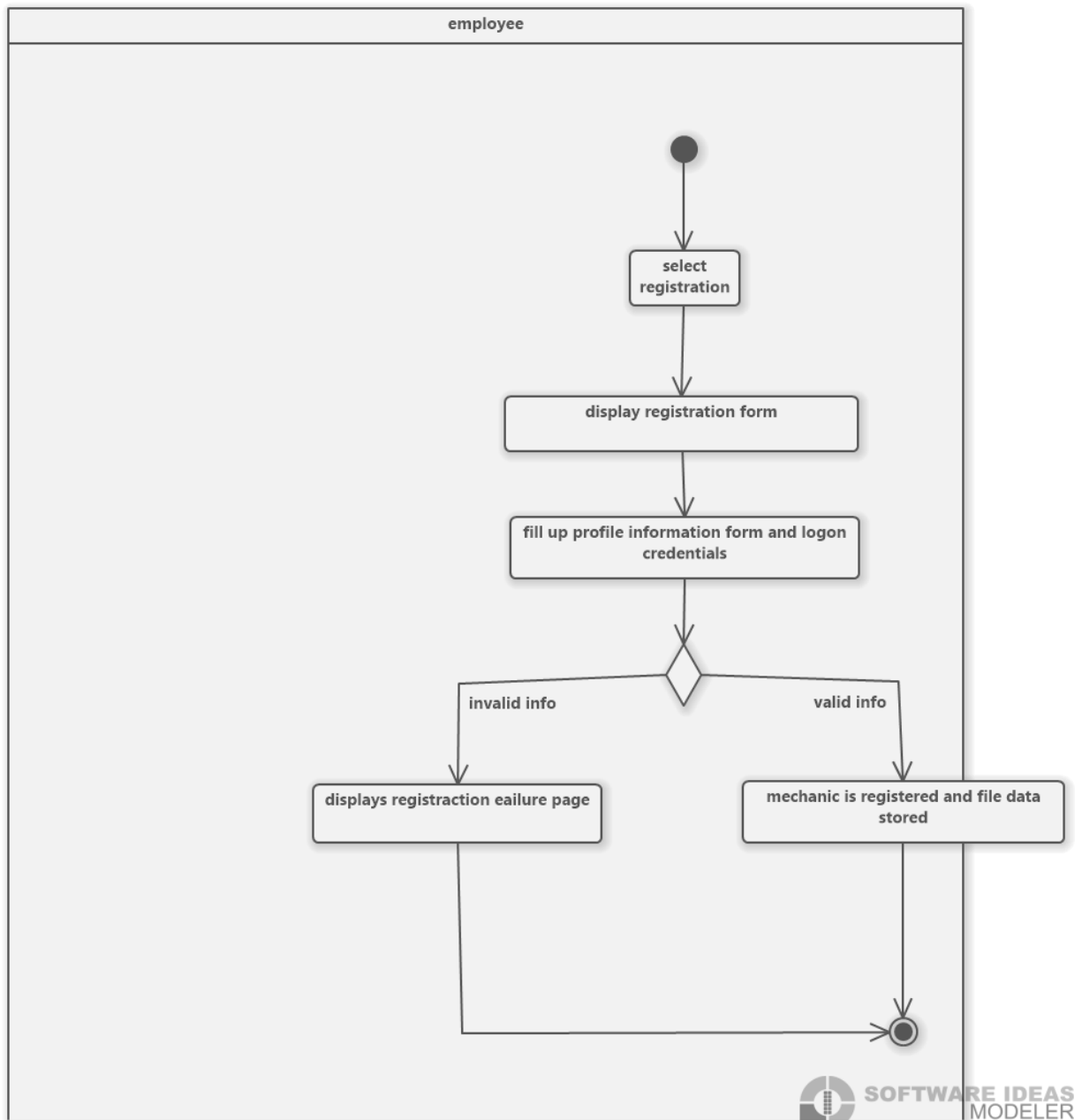
Element		Description	
Use Case Name		Book repairing appointment	
Use Case ID		UC-03	
Priority		High	
Actor(s)		Customer	
Description		The customer can book an appointment to repair the car	
Precondition(s)		The customer logged in and selects appointment	
Post-condition(s)		Appointment booked	
Flow of events		<div>1- The customer logs in the system</div> <div>2-The customer selects " Book repairing appointments "</div> <div>3-The system recalls all available and free appointments</div> <div>4-The customer selects the appointment that is suitable for him</div> <div>5-The system reserves the selected appointment</div>	
Alternative flows and exceptions		<div>1- The customer select two different appointments</div> <div>2- The system displays error message</div>	

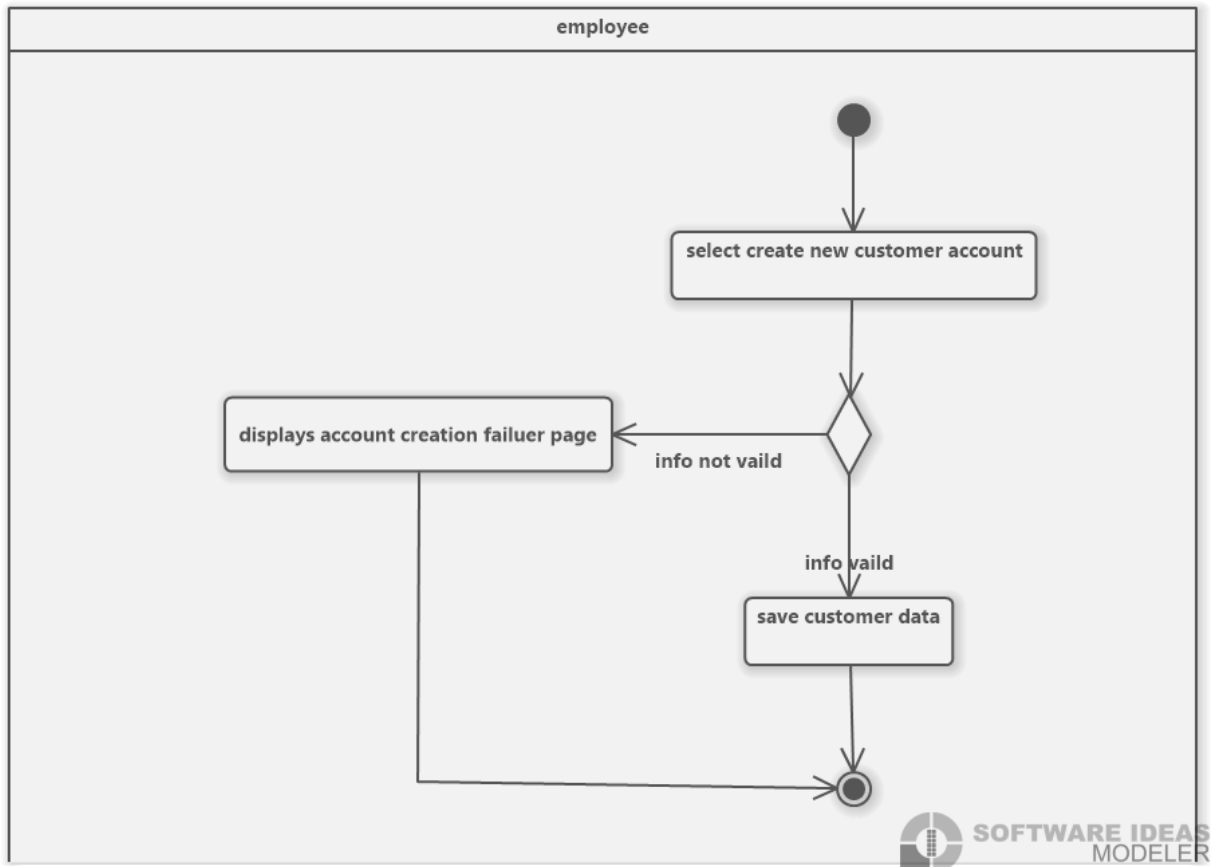
(generate repairing invoice):

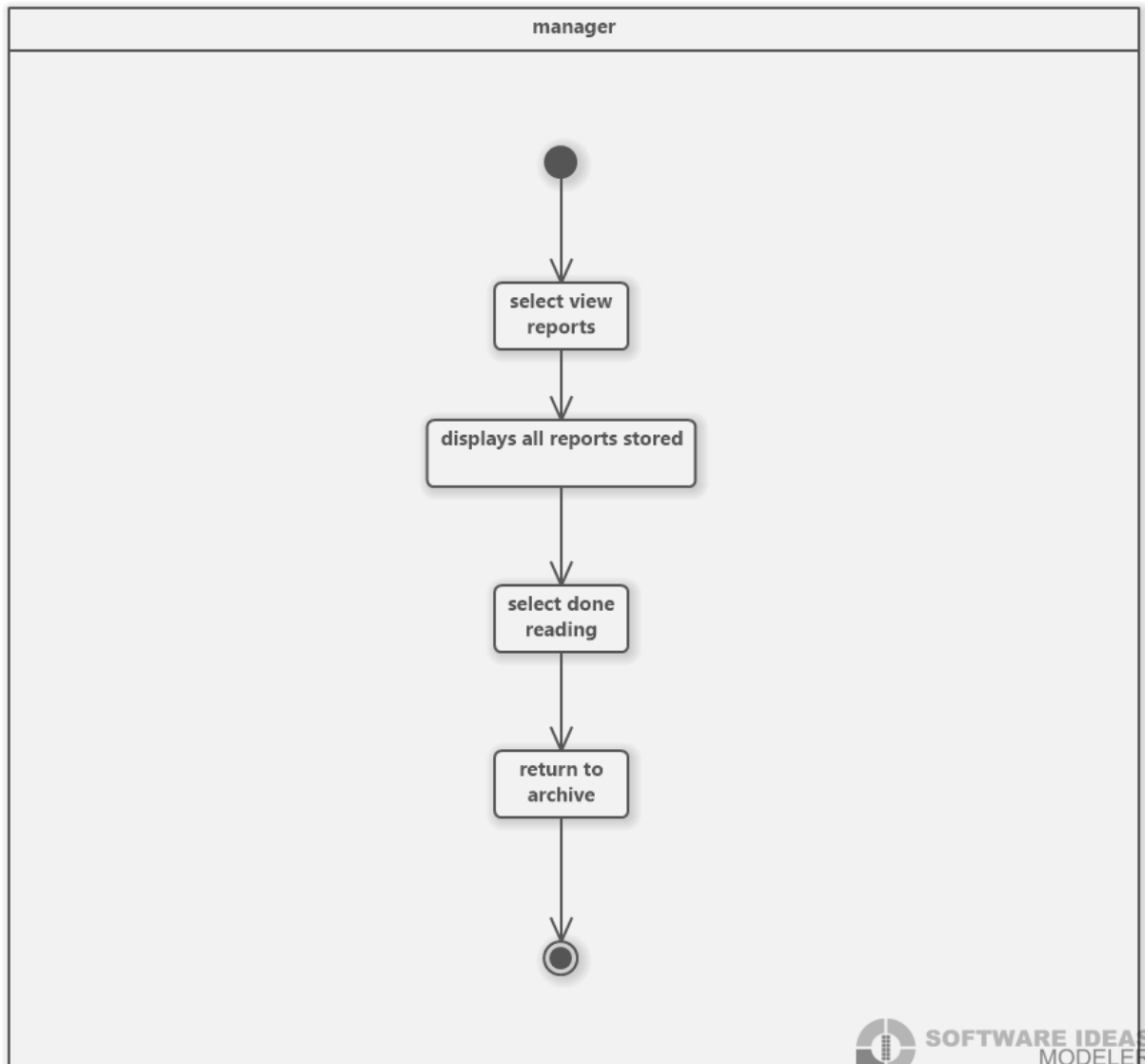
Element		Description
Use Case Name		generate repairing invoice
Use Case ID		UC-016
Priority		High
Actor(s)		Mechanic
Description		The mechanic will be able to generate repairing invoice
Precondition(s)		Services should be handled by a mechanic.
Post-condition(s)		Invoice generated.
Flow of events		<div>1- The mechanic repairs the car</div> <div>2-The mechanic selects “ Done repairing the car”</div> <div>3-The system displays invoice form</div> <div>4-The mechanic fills the invoice form with the name of spare parts and its price</div> <div>5-The mechanic selects “generate invoice”</div> <div>6- The system generates invoice and send it to the customer</div>
Alternative flows and exceptions		<div>1- The mechanic fills the invoice form with invalid information</div> <div>2-The system displays error message</div>

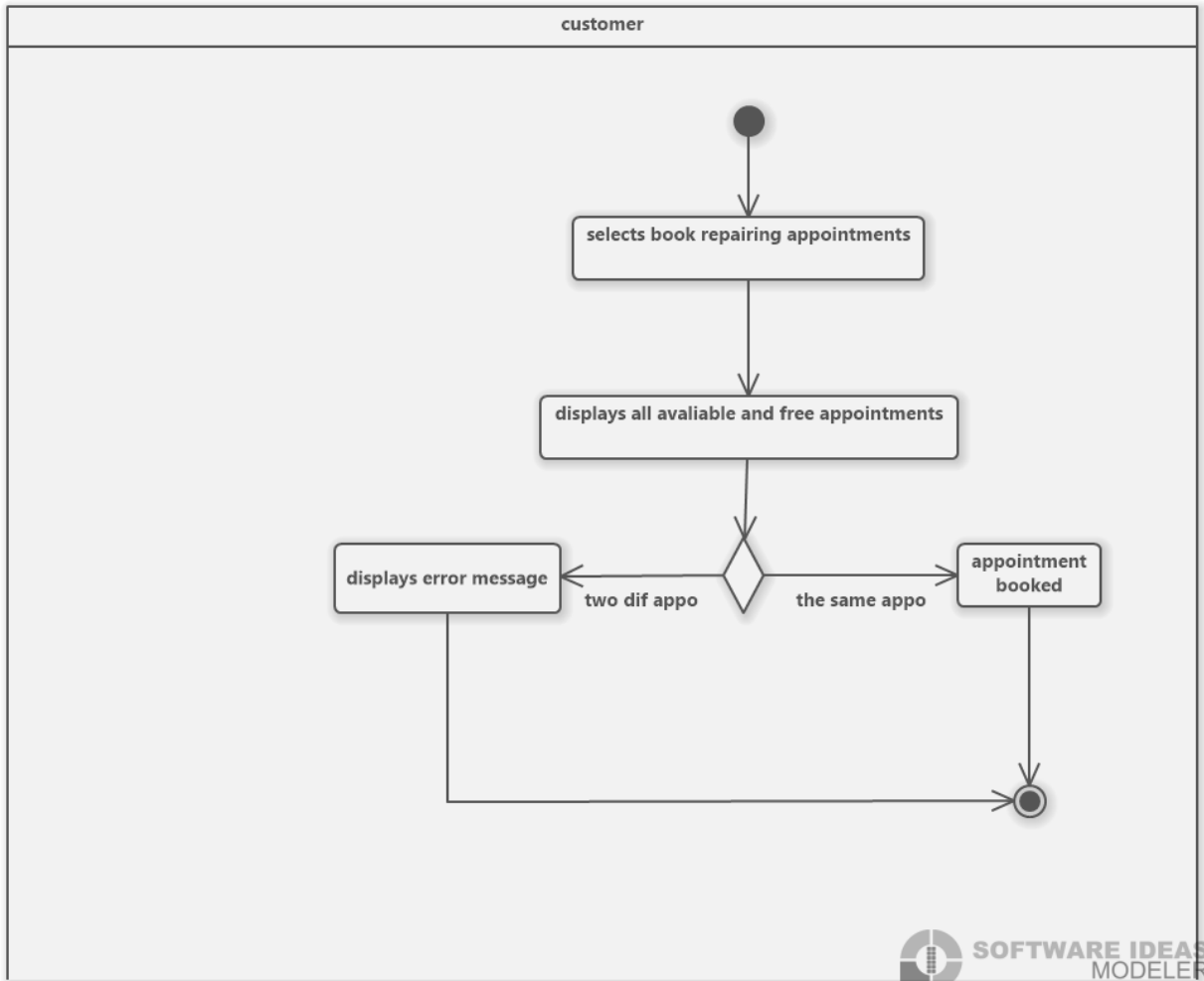
Activity Diagram

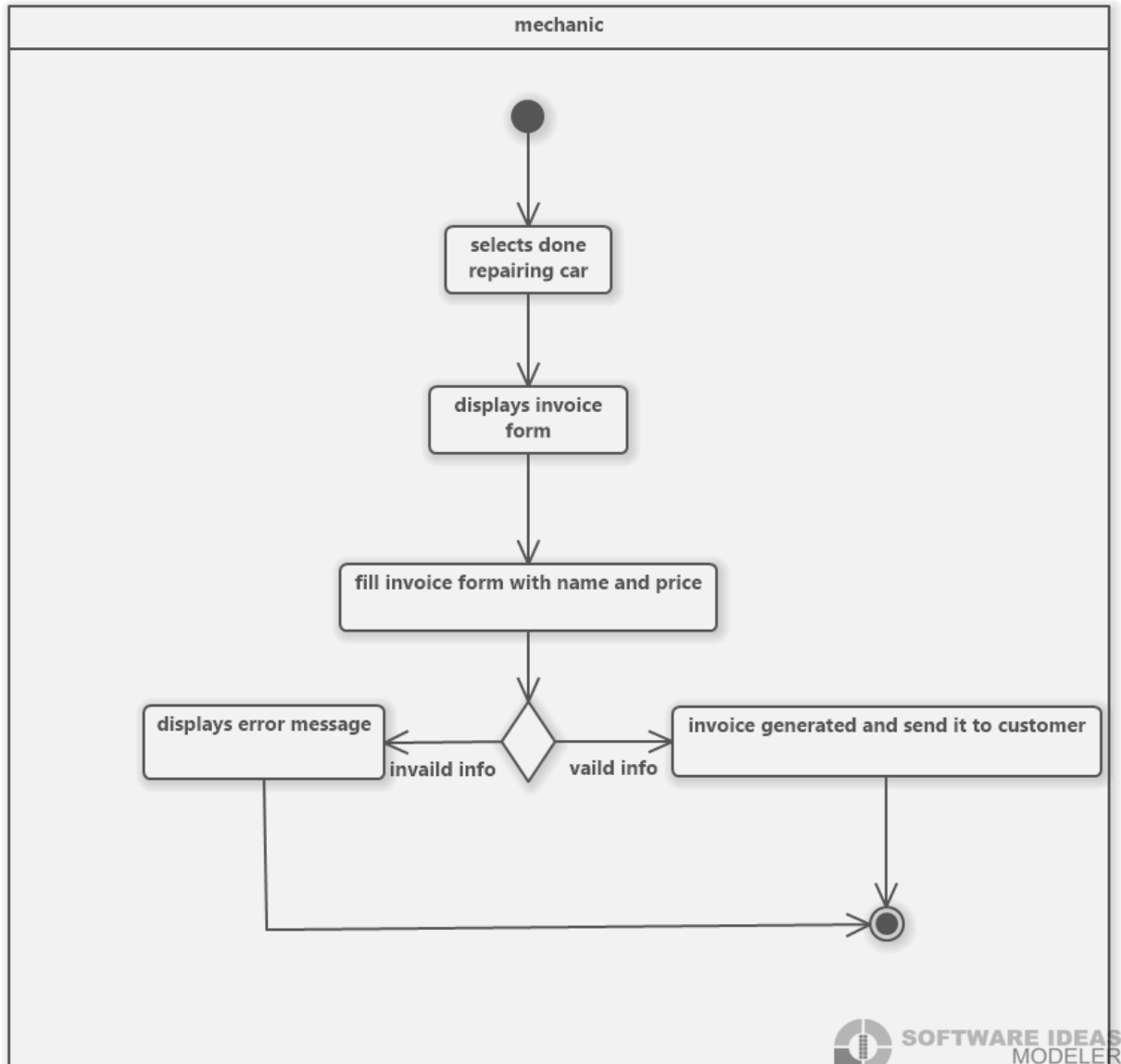






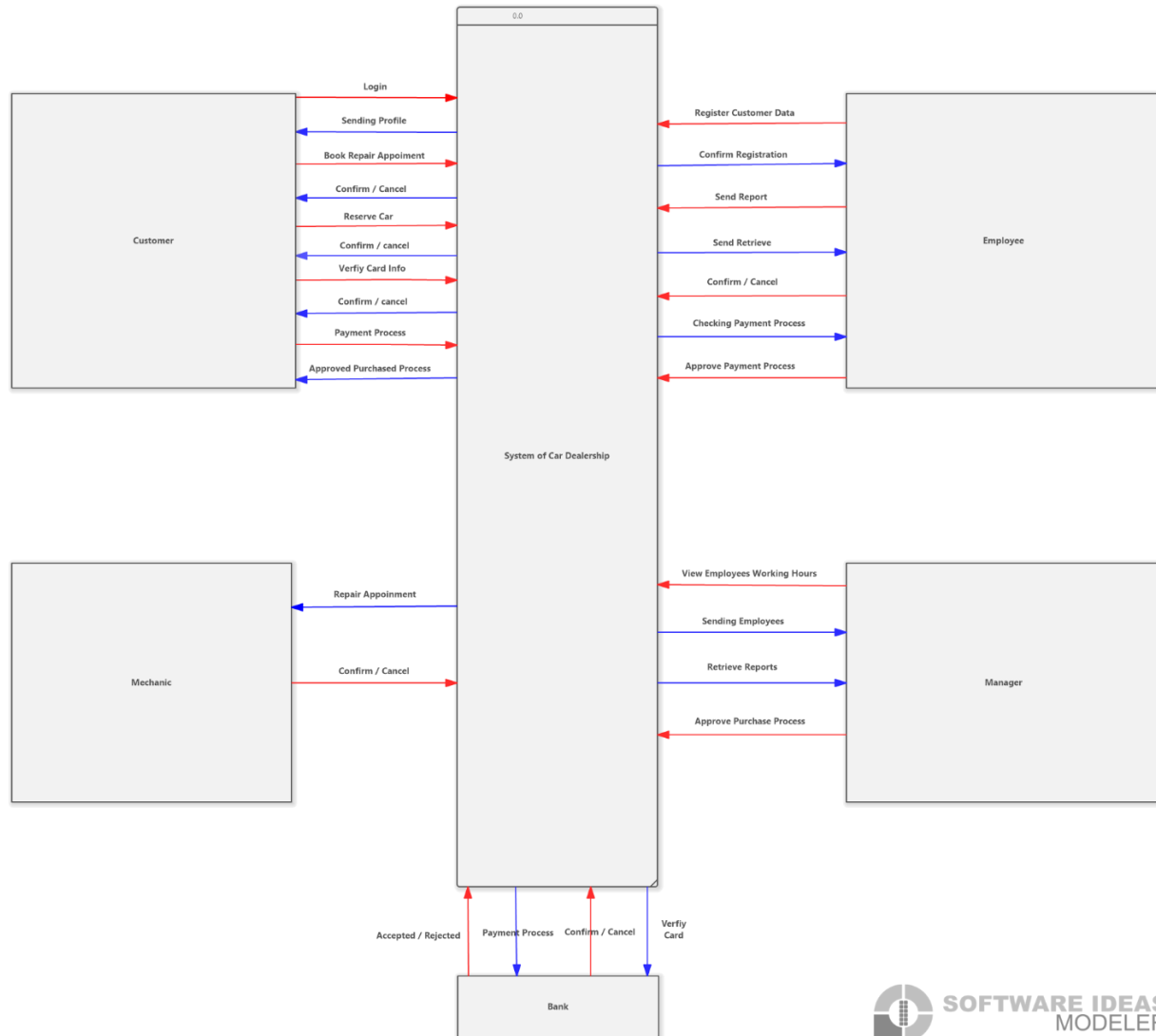




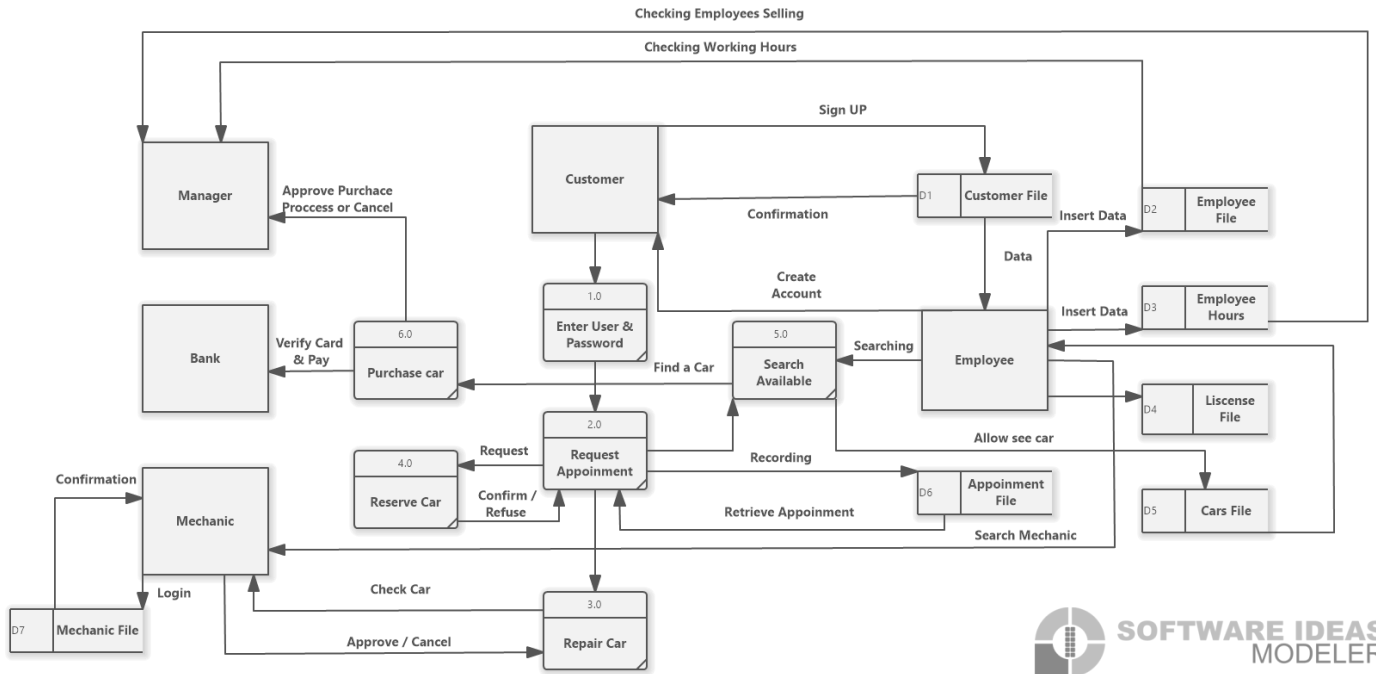


Data flow Diagram

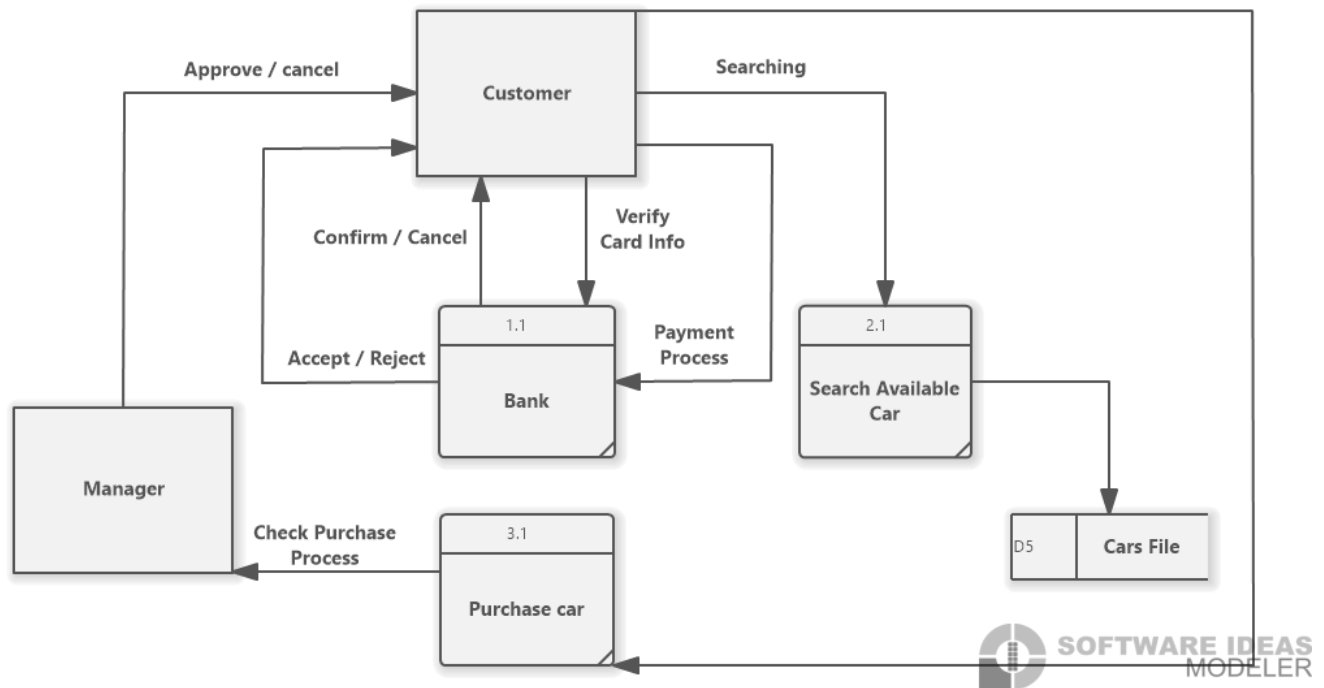
Context Diagram:



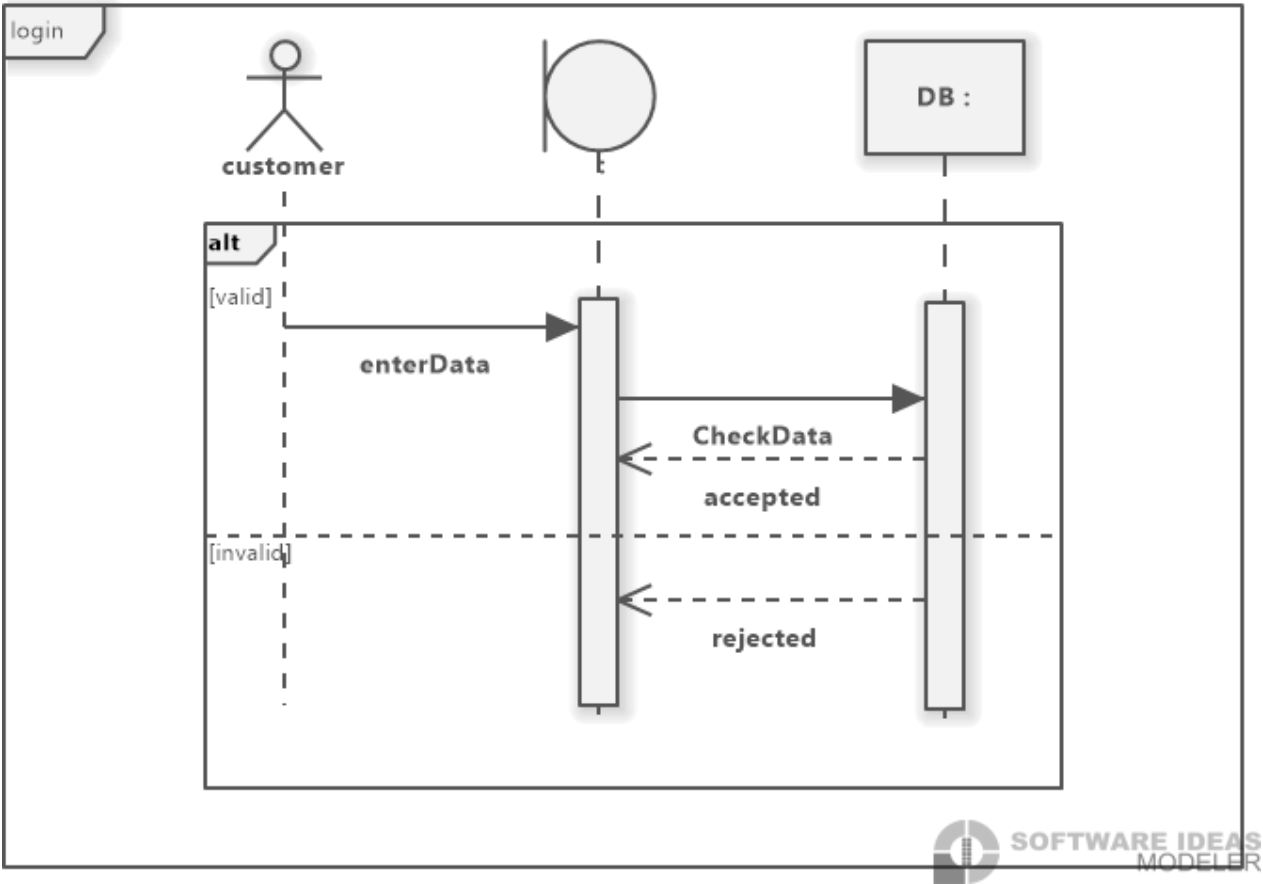
Level 0:

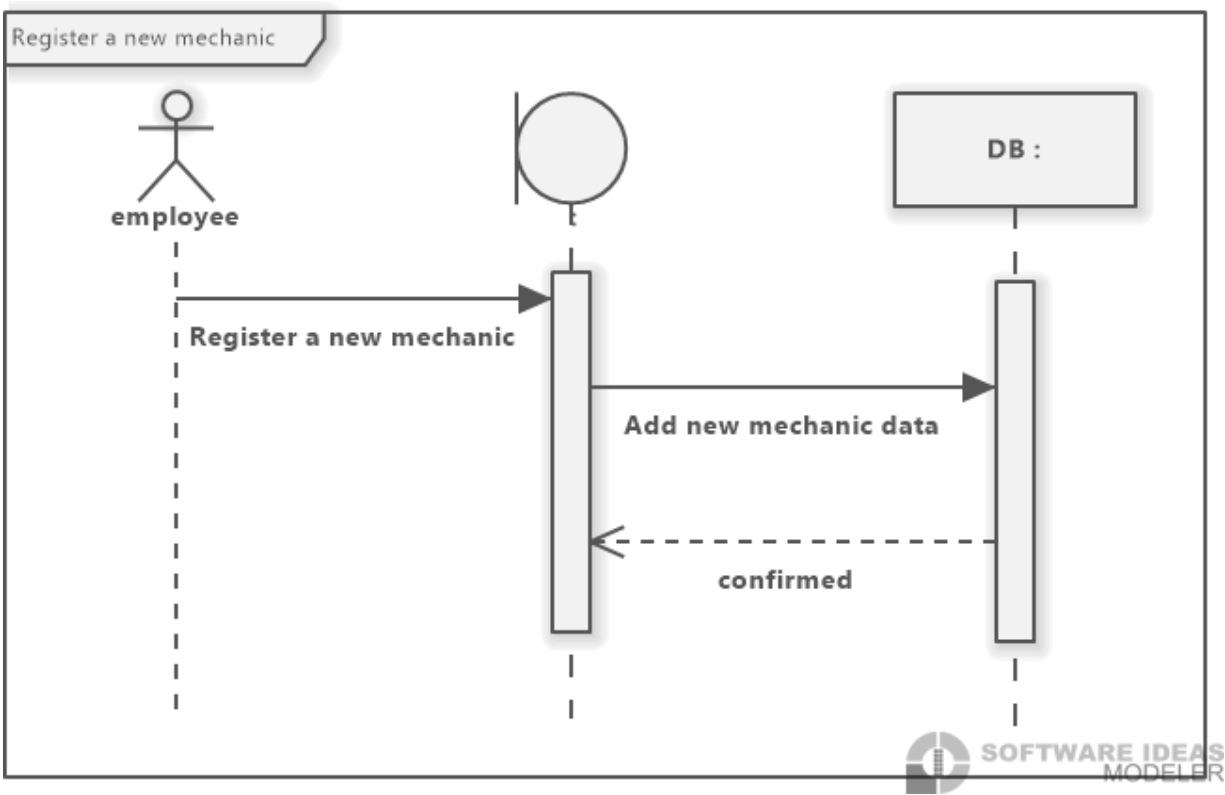


Level 1:



Sequence Diagram





Register a new customer

employee

Register a new customer

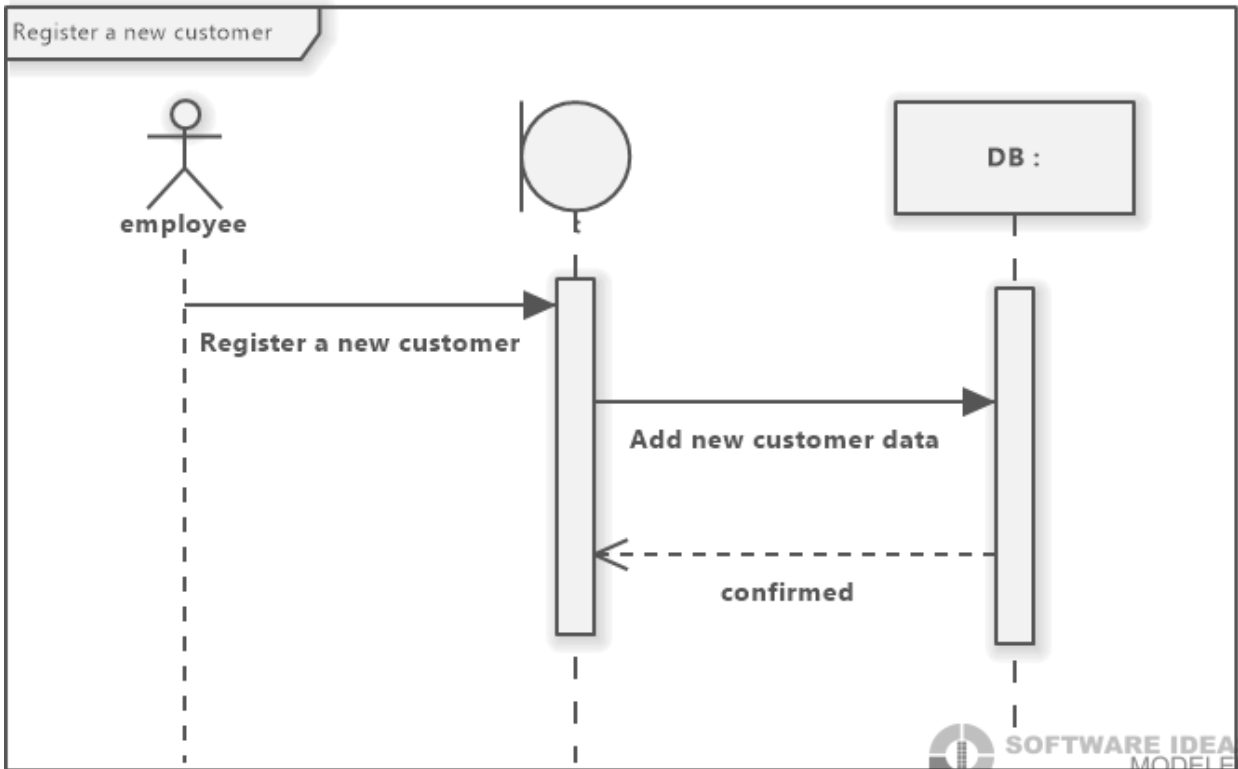
Add new customer data

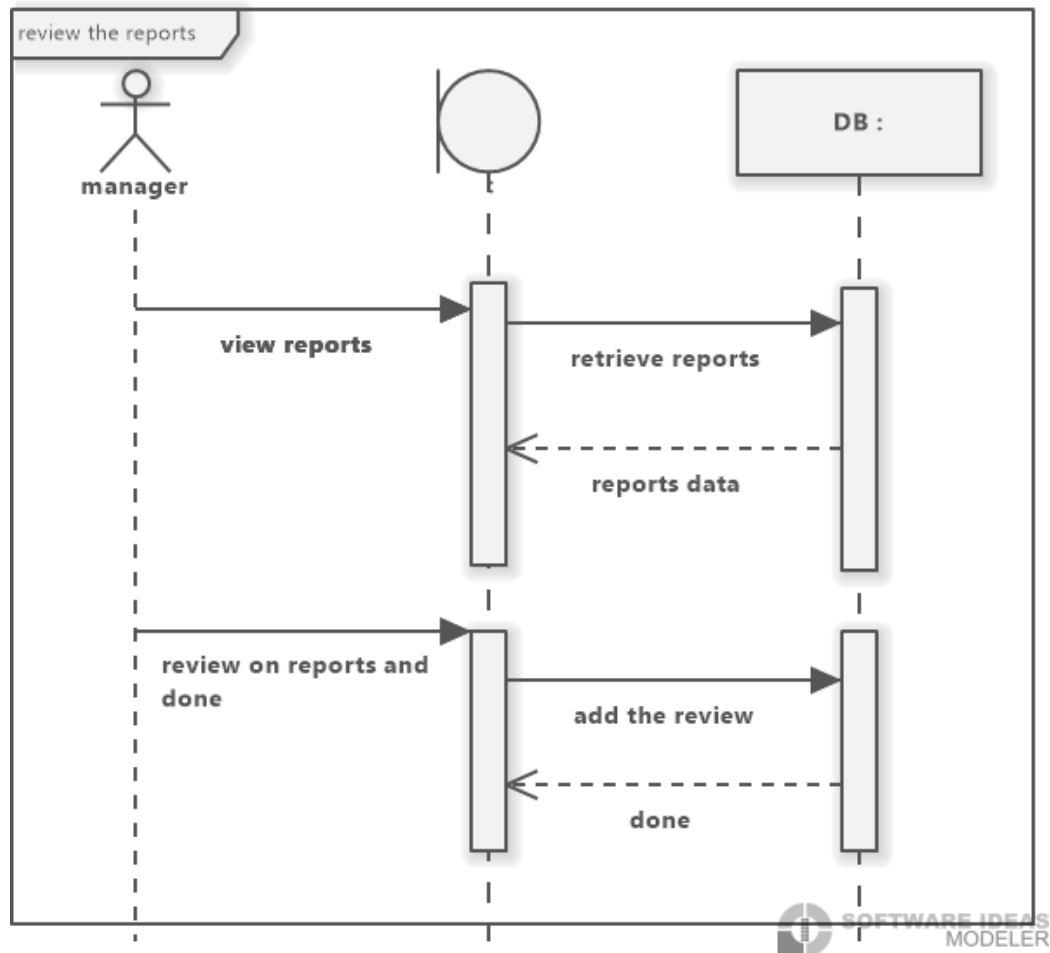
confirmed

DB :

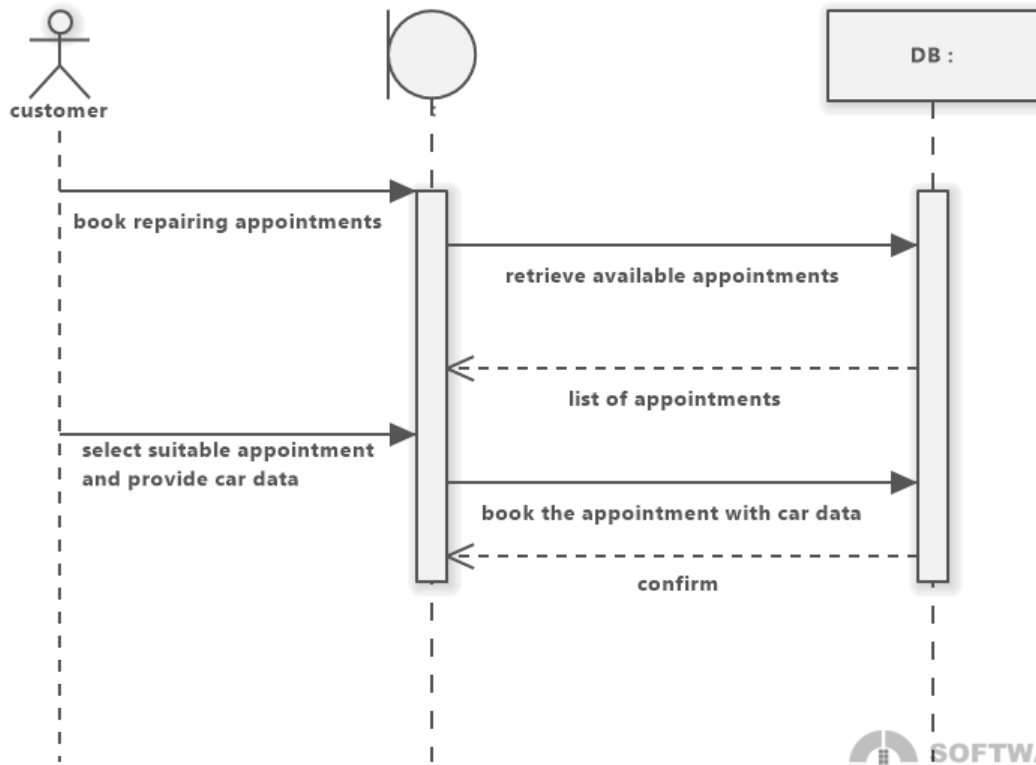


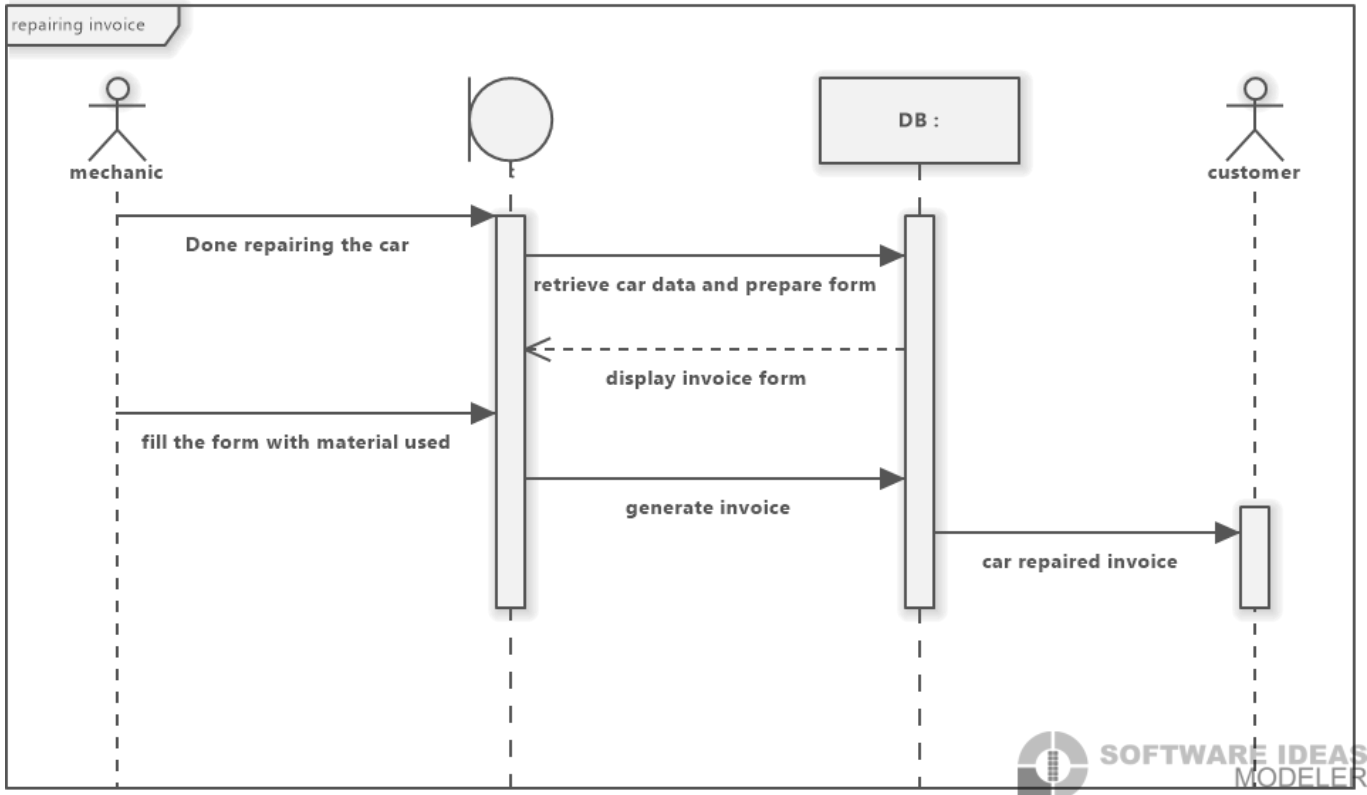
SOFTWARE IDEAS
MODELER



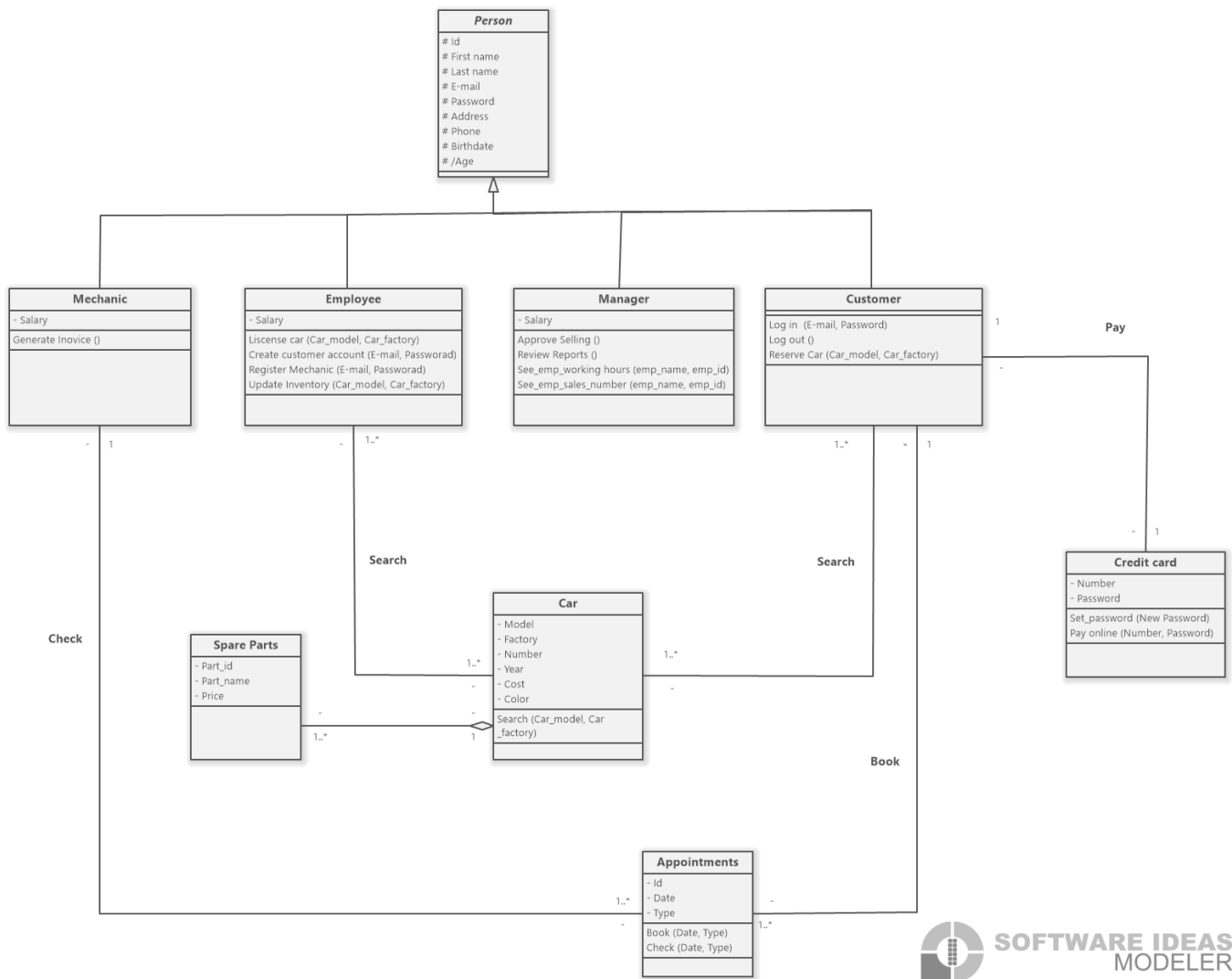


The customer can book an appointment to repair the car

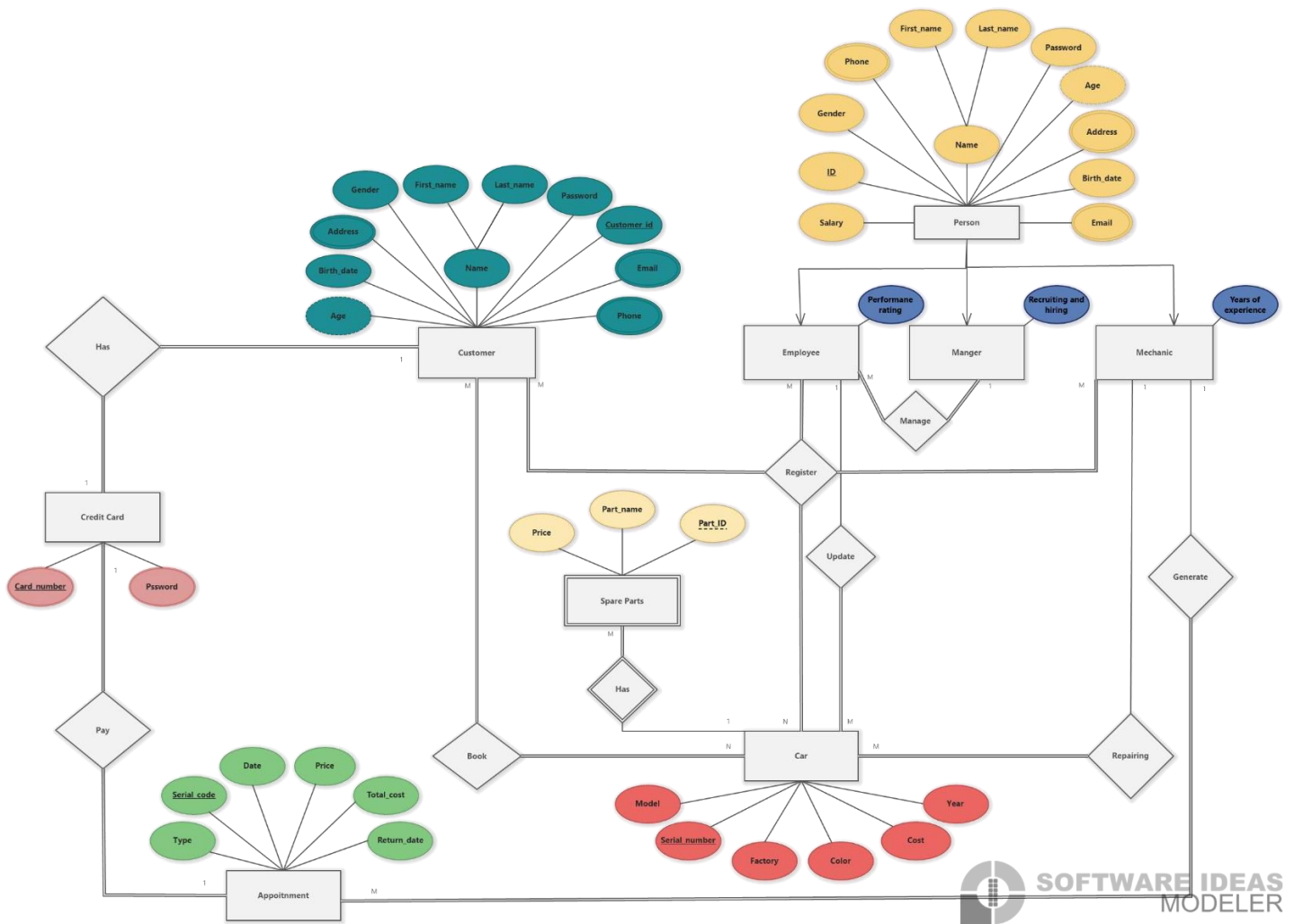




Class Diagram



Entity Relationship Diagram **(ERD)**



Dialog

- System: “Welcome Back! Log in to continue”
- User: *Clicks on log in button*
- System: What is your Username and password?
- User: *Enters his Username and Password*
- User: *Clicks on log in button*
- System: “Verifying your credentials”
- System: “Welcome back, [Username]”
- System: “What would you like to do today, [Username]?”

Storyboard

Car Dealership and Repairing

Welcome Back!

Log in to continue

Log in

Car Dealership and Repairing

 Log in to System

Username :

Password :

Log in

Car Dealership and Repairing

 Log in to System

Username :

Password :

 Verifying your credentials

Car Dealership and Repairing

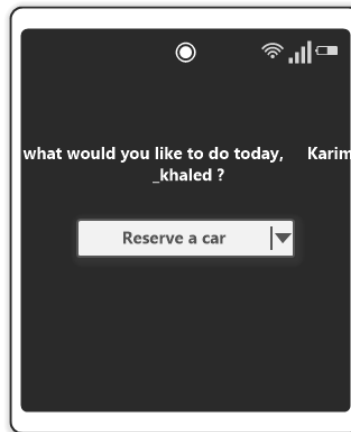
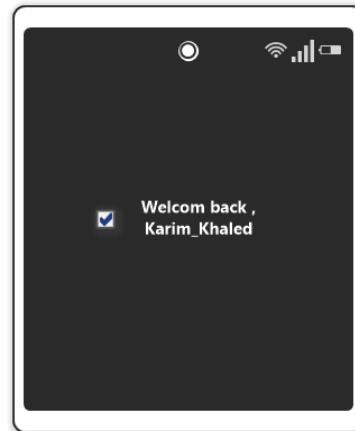
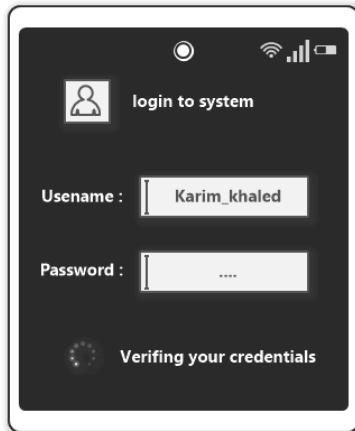
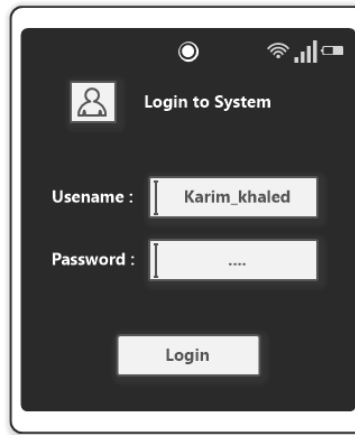
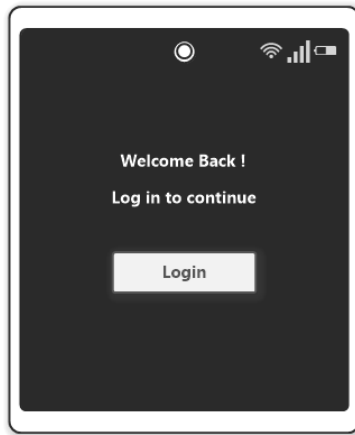
☒ Welcome back,
Mahmoud_Farag

Car Dealership and Repairing

What would you like to do today,
Mahmoud_Farag?

Reserve a car

handheld devices



Report 1
Occupied Appointments Report

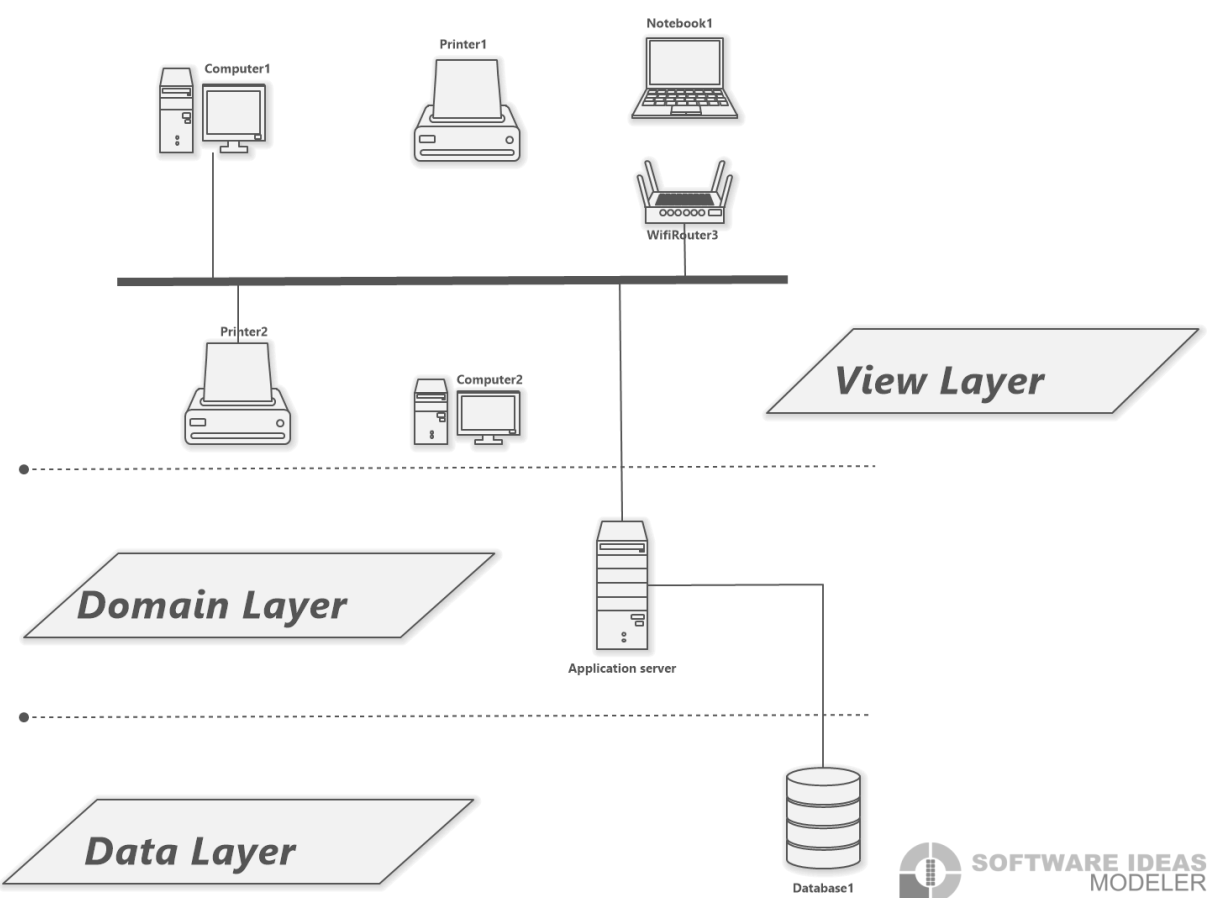
Appointment ID	Customer ID	Date	Type
1	5	6/12/2024	Repair
2	6	7/8/2024	Test-Drive
3	10	2/1/2024	Repair
4	15	3/6/2024	Test-Drive
6	18	8/12/2024	Test-Drive
7	7	6/8/2024	Repair
8	23	9/7/2024	Repair
9	55	2/2/2024	Repair
10	25	5/1/2024	Test-Drive

Available Appointments Report

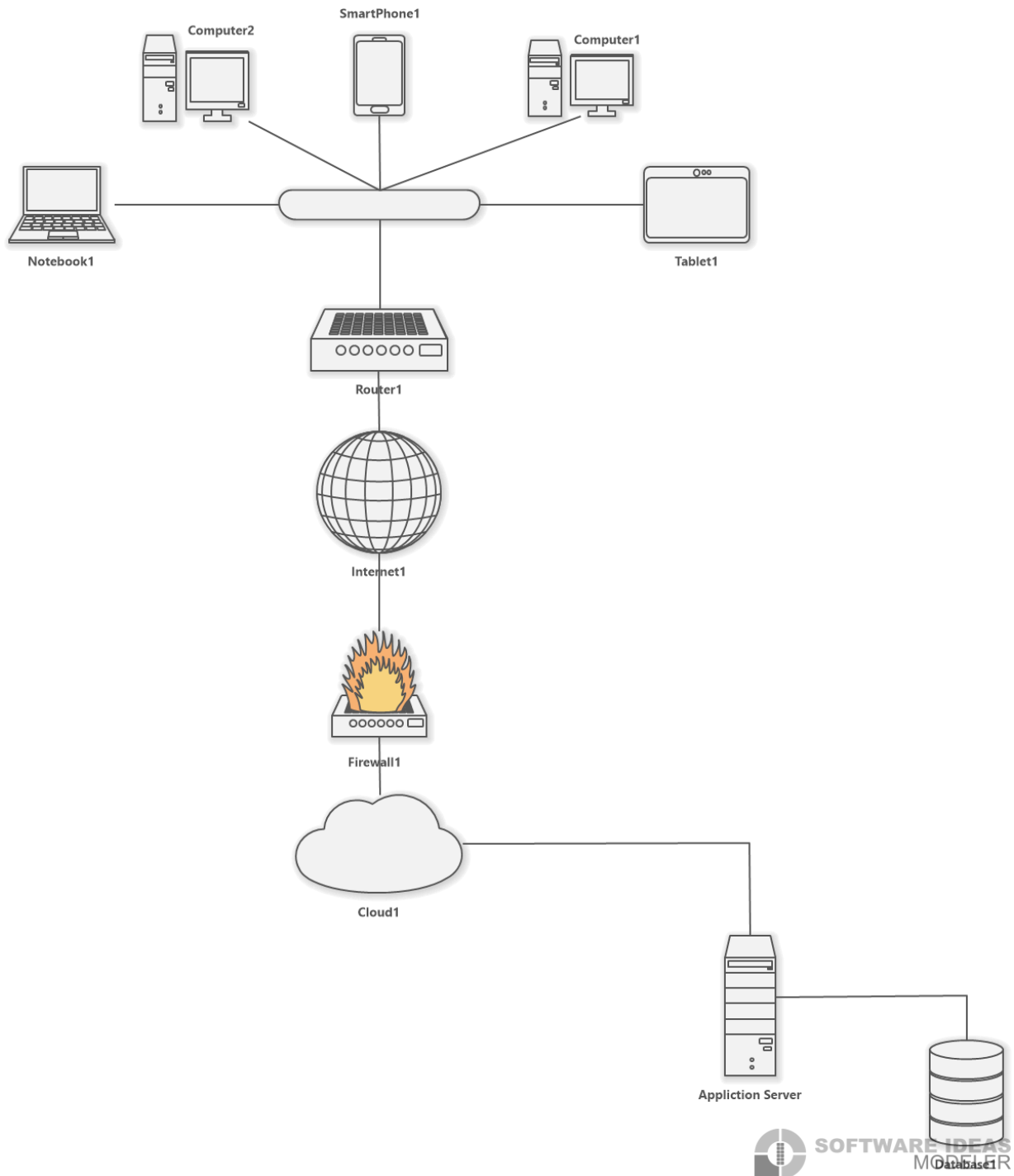
Appointment ID	Date	Type
155	6/2/2024	Test-Drive
134	17/8/2024	Repair
32	22/1/2024	Repair
47	13/6/2024	Repair
61	18/12/2024	Test-Drive
73	26/8/2024	Test-Drive
82	19/7/2024	Repair
19	12/2/2024	Test-Drive
109	25/1/2024	Test-Drive

Design of controls and security

Internal Network Diagram



External Network Diagram



Project Management:

System Vision Document

Problem Description:

A car dealership wants to improve its processes; this project is to build a platform to help facilitate operations and simplify transactions to get the desired results.

System Capabilities:

- **Efficient Customer-Vehicle Matching:**
 - Simple online and on-site vehicle search for customers.
- **Inventory management:**
 - Editing prices, quantities and reviews and update
- **Easy to use:**
 - Easy to use system with the capability for searching for cars within
- **Availability:**
 - Available on multi-platforms (website& hand-held devices)

Business Benefits:

- **Enhanced Customer Experience:**
 - Smooth processes for a better customer journey.
- **Increase sales:**
 - Servicing many employees at the same time
- **Reliable Inventory Management:**
 - Accurate and automated inventory tracking.
- **Improved Decision-Making:**
 - Insights for strategic decisions through analytics.
- **Multi-platform:**
 - Easy-to-use website and mobile app.
- **Scheduling :**
 - Access employees schedule (as a manager for rating or as a customer for book mechanic appointment)
- **Integration and Scalability:**
 - accommodate business growth and efficient data flow across departments.
- **Data Security:**
 - Protect customer & employee's data.

Traceability matrix

ID	Requirement	Related Use Case	Fulfilled By	Related Design Components	Test	Description
1	Each member must log in or out of the system, through the username and password, after verifying the validity of data, and in the event that any of them (name, password) is entered wrong, access to the system will be refused.	Log in	Controller.ui	Customer	1.Enter correct username and password, Enter incorrect username and password	Users must be able to log in or out of the system using valid data. Access will be denied if either the username or password is entered incorrectly.
2	The customer can reserve the car.	reserve car	Customer, Controller.ui	Customer	Select a car for reservation, Provide customer details for reservation, Choose reservation dates/times, confirm selection.	Customers should be able to reserve a car.
3	customer can search for a car.	N/A	Controller.ui	N/A	Search for cars by make, model and price	Customers should be able to search for available cars
4	The customer can book an appointment to repair the car	Book appointment to repair car	Controller.ui	Customer	Verify that the customer can access the booking system and schedule a repair appointment, Check that the system displays a list of available time slots for the customer to choose from, confirm that the customer receives a	Customer can book an appointment, System should provide available time slots, Customer should receive confirmation
5	The customer can book an appointment to test drive the car before buying it.	Book appointment to test drive the car	Controller.ui	Customer	Select car for test. Choose preferred appointment date. Provide employee details for appointment. Confirm test drive	Customers should be able to book an appointment for a test drive before purchasing a car.
6	The customer can pay for a car online, after inserting credit card info.	pay for online, Insert credit card information	Controller.ui	Customer	Verify that the customer can access the payment system and initiate the payment process, Test the system's ability to securely handle and process credit card information, and Send payment confirmation	Customer can initiate online payment, System should securely handle credit card information, ensure payment is recorded in the system
7	The customer can pay the invoice online, after inserting credit card info.	Can repairing invoice online, Insert credit card information	Controller.ui	Customer	Select an invoice to pay. Enter valid credit card information. Confirm and process the payment.	Customers should be able to pay the invoice online after entering credit card information.
8	The employee must license the cars before selling them.	License car before selling	Controller.ui	Employee	Verify that the employee can log in to the licensing system, Test that the system and completeness of the licensing information.	Employee can access the licensing system, Generate, and store licensing documents and Send notifications for license renewal

9	The employee must create an account for the customer, after checking their data	Create account for customer	Controller.ui	Employee	Verify customer data, Employees should be able to create an account for customers after verifying their data. Create customer account.	Employees should be able to create an account for customers after verifying their data.
10	The employee must register a new mechanic to the system.	register a new mechanic to the system.	Controller.ui	Employee	employee can log in to the mechanic registration system, the system validates, information entered by the employee for a new mechanic, sends confirmation notification to employee	Employee can access the mechanic registration system, validate mechanic information, Send confirmation to the employee and mechanic
11	The employee can search and spare parts in inventories.	Search the available cars in the internal and external inventories.	Controller.ui	Employee	The employee can select car model/year and choose spare part	Employees should be able to search for available cars and spare parts in the inventories
12	The employee update inventory data.	N/A	Controller.ui	N/A	The employee can log in to the inventory management system, the system updates the inventory records accurately based on the changes made by the employee	Employee can access the inventory management system, Update inventory records in the system
13	The manager must approve the selling process.	must approve the selling process.	Controller.ui	Manager	Initiate the selling process. Manager approval required for completion. Manager approves the selling process.	The manager must approve the selling process before completion.
14	The manager can see the employees' working hours.	See employee working hours	Controller.ui	Manager	the manager can log in to the system for viewing employee working hours, the system displays a comprehensive list of employees and their respective working hours	Manager can access the employee working hours system, show total hours worked for each employee
15	The manager can see the employees' sales numbers	See employee sales number	Controller.ui	Manager	Access employee sales data. Ensure only authorized managers can view data. Verify accurate display of sales numbers	The manager should be able to view employees' sales numbers.
16	The manager can review the reports on the different operations performed	Review the reports of Different operations performed	Controller.ui	Manager	the manager can log in to the reporting system, the system provides a comprehensive list of available reports on different operations	Manager can access the reporting system, provide options to view summary and detailed reports
17	The mechanic will be able to check the appointments for repairing cars.		Controller.ui	check the appointments of repairing cars	the mechanic can log to the appointment system, appointments assigned to the mechanic, confirm that the system notifies the mechanic of new appointments	Mechanic can access the appointment system, display a list of upcoming repair appointments, Provide notifications for new or updated appointments

Work breakdown structure

1. Discover and understand the details of all aspects of the problem.

A. Gather Requirements ~ 8 hours.

- Conduct Stakeholder Interview ~ 2 hour.
- Identify and define use cases ~ 2 hours
- Identify and define information requirements ~ 1 hour.
- Define Workflows and descriptions for the use cases ~ 3 hours.

B. Research and Analysis ~ 2 hours.

- Generates reports on sales performance ~ 1 hour.
- Identify Potential Challenges and Risks ~ 1 hour.

C. Establish Project Plan ~ 6 hours.

- Define Timeline and Milestones ~ 2 hours.
- Allocate Resources ~ 2 hours.
- Develop Communication Plan ~ 2 hours.

2. Design the components of the solution to the problem

A. Architectural & Environment Design ~ 1 day.

- Define System Architecture ~ 3 hours.
- Identify system approach and methodology ~ 3 hours.
- Specify the technology required to support the software application ~ 3 hours.

B. Detailed Design ~ 5 days

- Design (Layout) inputs, output screens and reports 1 day.
- Build database and schema and determine (attributes, keys, Indexes) 2 days.
- Design of user and system interfaces 1 day.
- Design of controls and security ~ 1 day.

3. Build the components and integrate everything into the solutions ~ 2 days.

Development ~ 2 days.

- Code and unit test GUI layer programs

4. Perform all system-level reporting

- Verify all system levels and check for complete ~ 3 days.

Project Schedule

		Sunday	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday	Sunday	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday	Sunday
Planning	System Request															
	SWOT															
Analysis	User Requirement															
	System Requirement															
	Non Functional Requirement															
	Functional Requirement															
	Use case															
	Use case Description															
	Activity Diagram															
Design	Data Flow Diagram															
	Entity Relation Ship Diagram															
	Architectural Design															
	Sequence Diagram															
	Class Diagram															
	User interface															
	Design handheld device															
Implementation																
Project Management	Design of control and security															
	System vision															
	Work breaks down															
	Traceability matrix															
	Project Schedule															
Report																

Phases	Start	End
Planning	Sunday, December 10, 2023	Sunday, December 10, 2023
Analysis	Monday, December 11, 2023	Friday, December 15, 2023
Design	Saturday, December 16, 2023	Tuesday, December 19, 2023
Implementation	Wednesday, December 20, 2023	Thursday, December 21, 2023
Project Management	Friday, December 22, 2023	Sunday, December 24, 2023

Implementation

Class Person:

```
package system.pkg2;
import java.util.Date;
public class Person {
protected int id;
protected String firstName;
protected String lastName;
protected String email;
protected String password;
protected String address;
protected int phone;
protected Date birthdate;
protected int age;
}
```

Class Mechanic:

```
package system.pkg2;
import java.util.Scanner;
public class Mechanic extends Person {
    Scanner input = new Scanner(System.in);

    private double salary;
    public void generateInvoice(double
repairing_cost) {
repairing_cost=input.nextDouble();
System.out.println("Invoice="+repairing_cost+"
L.E");
    }

}
```

Class Employee:

```
public class Employee extends Person{
    private double salary;
    public void liscense_car(String Car_model,String
Car_factory){
    System.out.println("Car model:"+Car_model);
    System.out.println("Car factory:"+Car_factory);
    }
    public void create_customer_account(String
email,String password){
    super.email=email;
    super.password=password;
    }
    public void register_mechanic(String email,String
password){
    super.email=email;
    super.password=password;
    }
    public void update_inventory(String
car_model,String car_factory){
    System.out.println(car_model +" is updated
");
    System.out.println(car_factory +" is updated
");}
}
```

Class Manager:

```
package system.pkg2;
public class Manager extends Person {
    private double salary;

    public void approve_selling(){
        System.out.println("Selling is approved");
    }
    public void review_reports(){
        System.out.println("Reports is reviewed");
    }
    public void see_emp_workinghours(String
emp_name,int emp_id){
        emp_name=super.firstName;
        emp_id=super.id;
    }
    public void see_emp_salesnumber(String
emp_name,int emp_id){
        emp_name=super.firstName;
        emp_id=super.id;
    }

}
```

Class Customer:

```
package system.pkg2;
public class Customer extends Person{
    public void log_in(String email,String Password){
        if (email==super.email &&
            password==super.password){
            System.out.println("Login Successfully");
        }
        else{
            System.out.println("Invalid Credintals");
        }
    }
    public void log_out(){
        System.out.println("Log-out Successfully");
    }
    public void reserve_car(String car_model,String
        car_factory){
        System.out.println("Car with model :
        "+car_model+" and factory : "+car_factory +" is
        reserved ");
    }
}
```


Thank You