Car Request Services

Group 9

Pragya Shukla

Khang Duong

Table of Contents

Introduction

The overall purpose of this project is to redefine car sales by automating the entire process beginning from selection and ending in a successful deal. This application enhances the customer car shopping experience by keeping a constant focus on the customer's requirements and satisfaction.

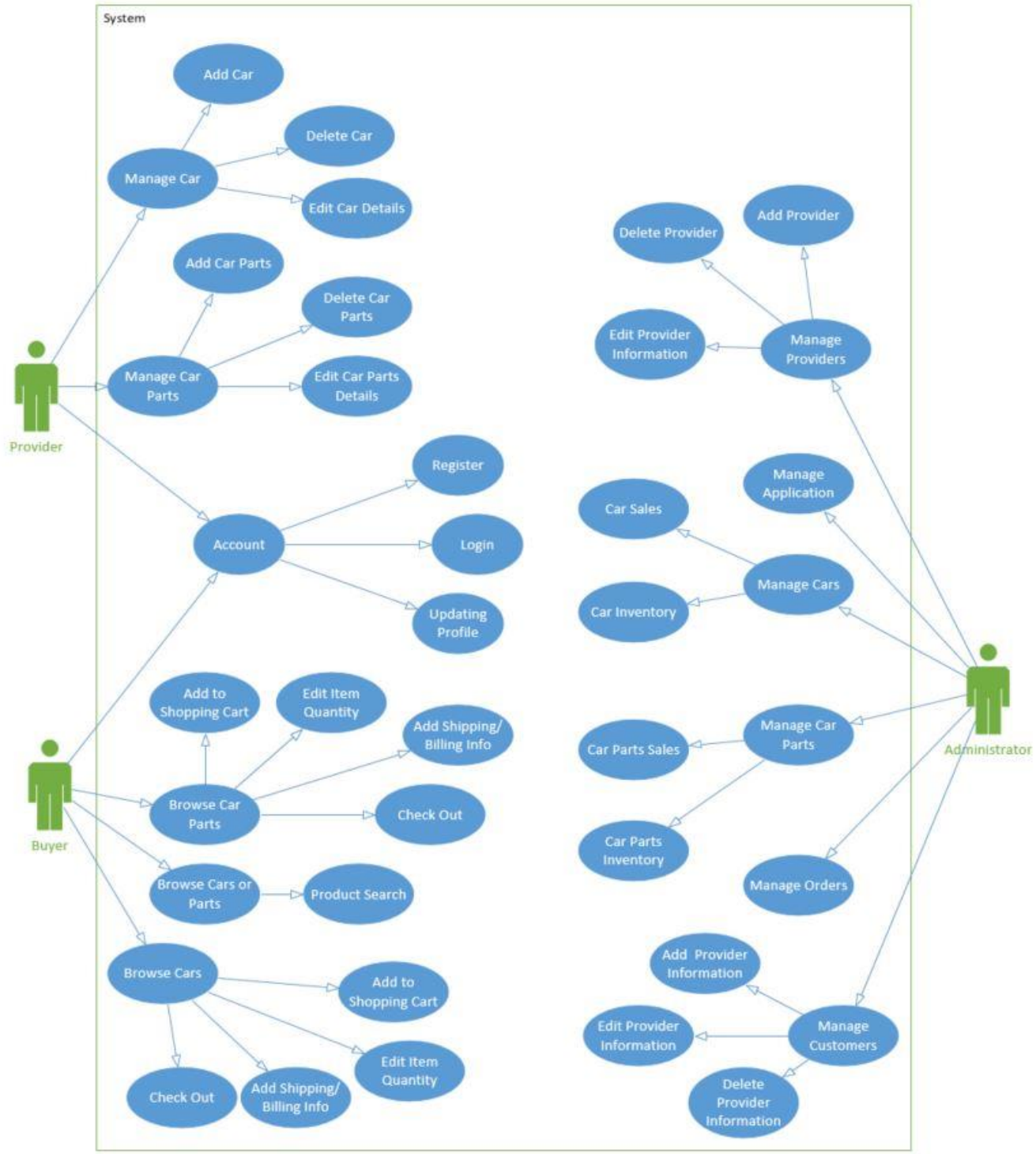
Highlights of this application are:

- Customer can become a member by creating a profile within the application.
- Various providers also register themselves.
- There is an admin which maintains all the profiles.
- Login authentication using social media site like Facebook.
- Customers can search for the car they have in mind using various criteria and start their short-listing process by viewing the exact cars and what they have to offer as specified by the search results.
- Multiple dealers can have their cars showcased with all details and customer can browse and select through them.
- Customers can maintain a profile in which they could shortlist cars they are interested in.
- Apart from cars customers can also buy/sell various car parts.
- A secure billing channel is maintained.

The web application will be a channel partner and become point of contact to sell cars available in the provider car lot. A major challenge to be overcome is to bridge the gap between various technologies used by different partner dealerships and provide a single point of access to the end consumer.
The application does not intend to limit its partnerships to select few dealerships but instead requires that the solution be flexible and scalable to add and remove providers. One of the ways the requirement can be satisfied is to provide a web-service to provide a unified method to access information available in their own database along with information obtained via the adaptation layer.

Requirement Specification Document (RSD)

Unified Modeling Language - Use Case Diagram



Requirements Table

Login and Signup

ID	Description	Version	Priority
1.0	Using Email/Phone Number for Authentication	1.0	High
1.1	Authentication using Gmail/Facebook	1.0	Low
1.2	Validation for fields	1.0	High
1.3	Check user input for the right data type	1.0	High
1.4	Required fields and Optional fields	1.0	High
1.5	User friendly interface	1.0	Medium
1.6	Security preventing SQL injections	1.0	High

Customer Management

ID	Description	Version	Priority
2.0	Will be able to register and login	1.0	High
2.1	Will be able to update profile	1.0	High
2.2	Can browse through the cars and the parts	1.0	High
2.3	Customer can add to cart	1.0	High
2.4	Customer can add/remove items from cart	1.0	High
2.5	Customers can input payment information	1.0	High
2.6	Customers can checkout	1.0	High
2.7	Customer can filter the cars and car parts	1.0	Medium

Service Provider Management

ID	Description	Version	Priority
3.0	Will be able to register and login	1.0	High
3.1	Will be able to update profile	1.0	High
3.2	Will be able to post and products/services online	1.0	High
3.3	Can add, edit and remove and posts	1.0	High
3.4	Can browse other products from competitors	1.0	High
3.5	Easy to use interface	1.0	High

Billing Management

ID	Description	Version	Priority
4.0	Provide invoice to customers from orders	1.0	High
4.1	Add, edit, remove, and search through all invoices	1.0	High
4.2	Can be updated in case of returns and refunds	1.0	High
4.3	Save and update user's billing data safely in a database	1.0	High
4.4	Check user's debt or pending invoices and send reminders	1.0	Medium
4.5	Email the invoices to customers and providers	1.0	High
4.6	Print the invoices if hard copy is requested	1.0	Medium

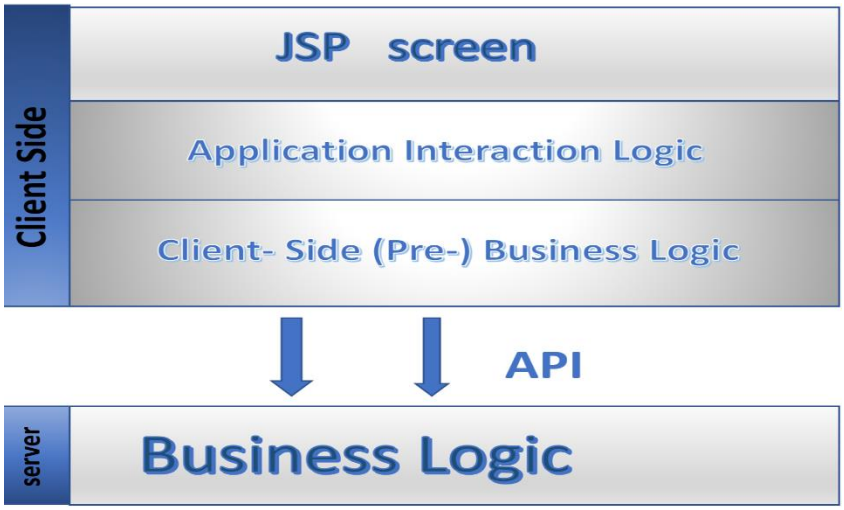
Inventory Manager

ID	Description	Version	Priority
5.0	Store quantity and status of each item	1.0	High
5.1	Add, Edit, Remove and search through inventory	1.0	High
5.2	Update inventory according to orders and posts	1.0	High
5.3	Show weekly inventory reports	1.0	High
5.4	Send out alerts when item is sold out	1.0	Medium

Service Register

ID	Description	Version	Priority
6.0	Keeps track of posts by service providers	1.0	High
6.1	Keeps track of orders from customers	1.0	High
6.2	Update inventory from orders and posts	1.0	High
6.3	Update Customer and Provider information per transaction	1.0	High
6.4	Store dates of purchases and posts	1.0	High
6.5	Store status of each car and car parts	1.0	High

Design Phase

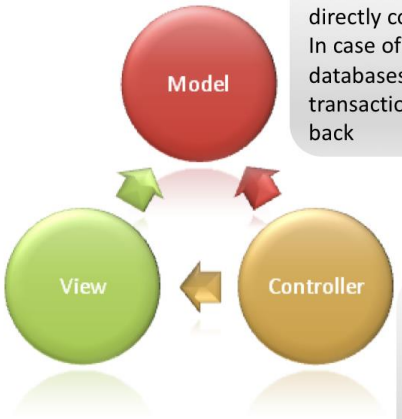


The web application created is a 3 tier MVC (Model, View, Controller) architecture-

- View – JSP screens with all the presentation logic for the end user.
- Model – The last layer interacting with the database performing all SQL related operations and business logic.
- Controller- It act as an interaction bridge between view and model.

- User management module for registration and user based logins(Consumer, Provider, admin).
- Customer Management module for creating a profile and selecting various services.
- Billing module for performing charging algorithms for all the services used.
- Maintaining an inventory of all the cars for sale from all the registered providers.
- Maintaining inventory for car parts for sale.

Architecture

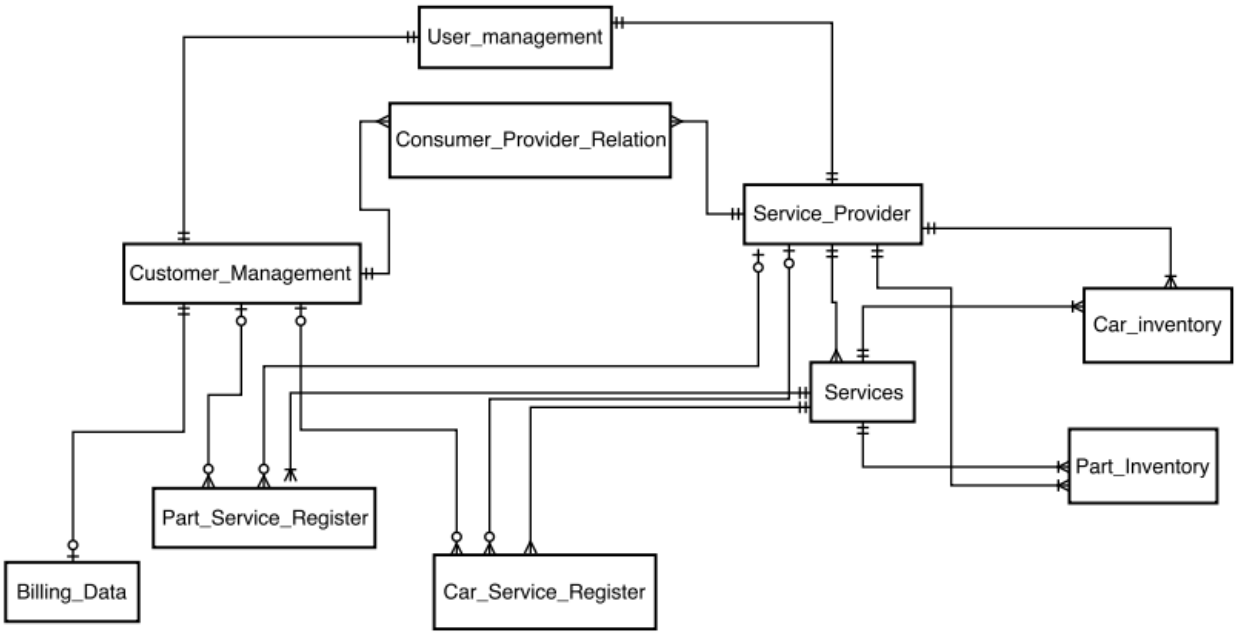


This module has all the business logic and directly communicated with the database. In case of a new user or car inventory databases are updated and all the transactions are calculated and stored back

These are JSP screens for client/Customer interaction with a secured login authentication. It provides all the responses to the user.

This module is a bridge between view and model receives the requests from customers or receives the forms and validates them. The data received from screen is in the form of XML parsing and retrieval data is performed before it can be passed to model layer

Entity Relationship Diagram

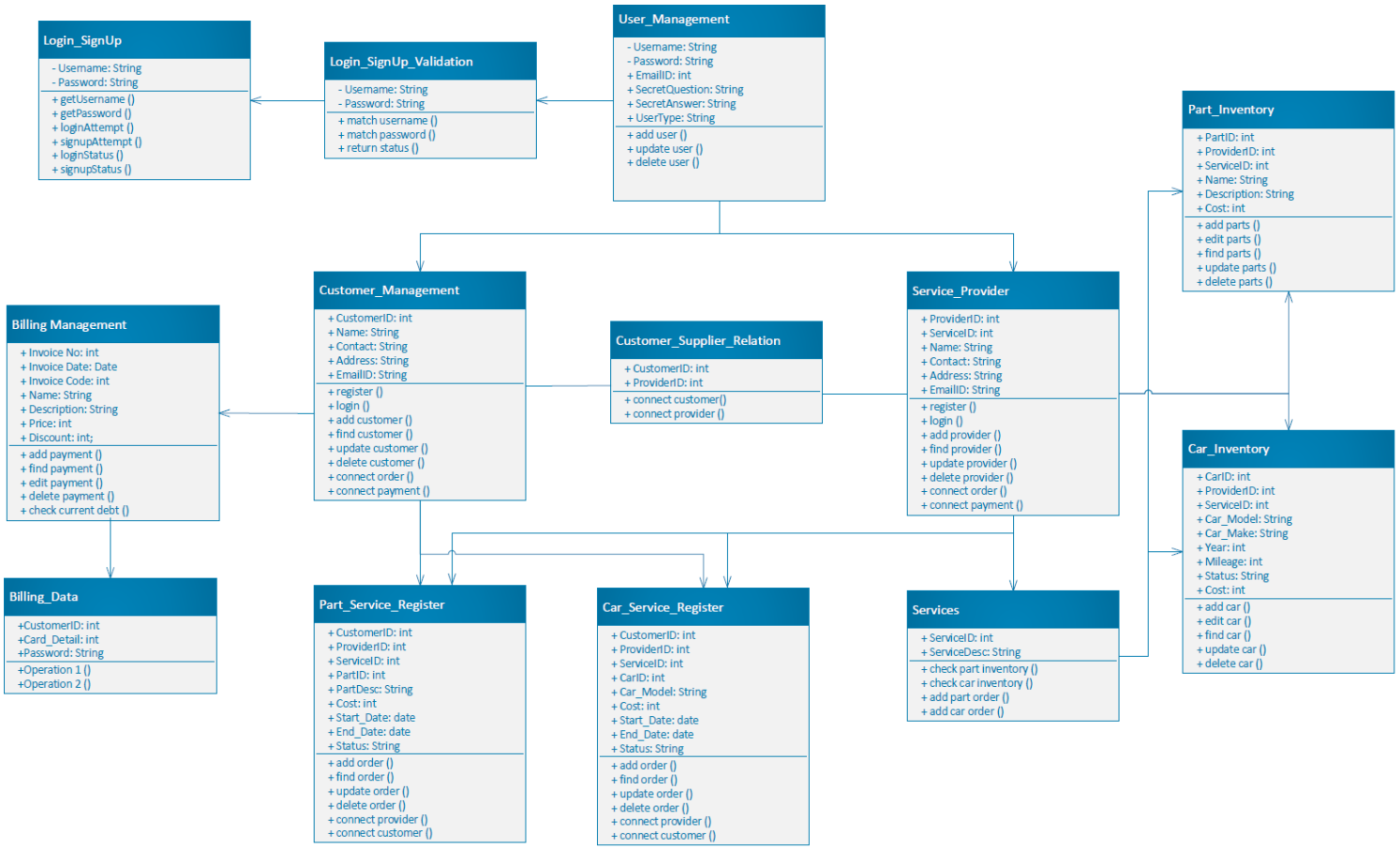


SQL Command

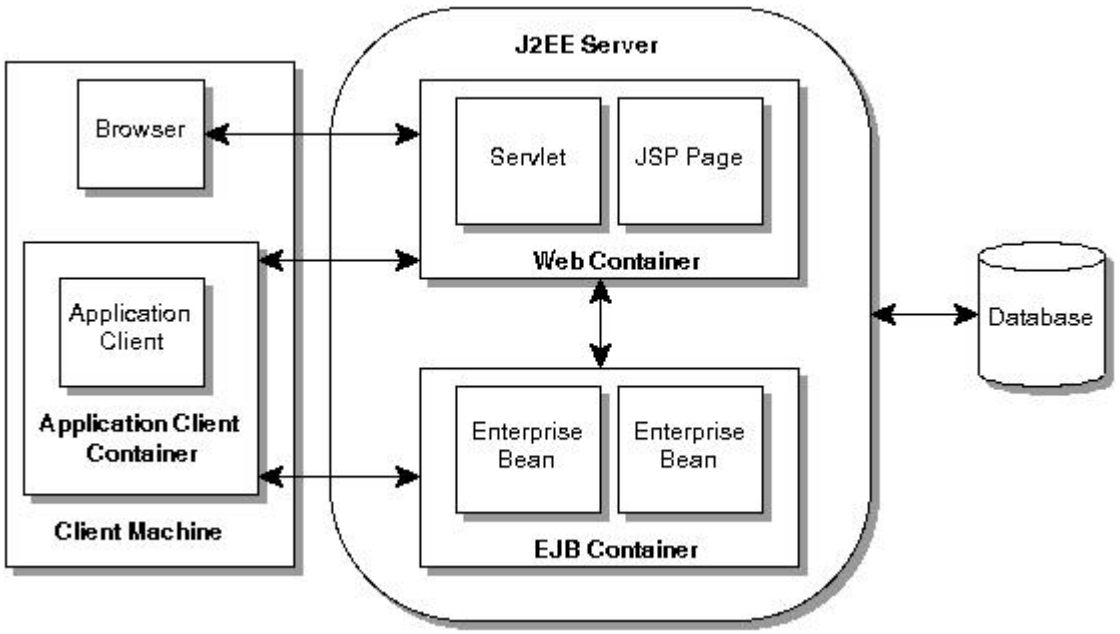
```
 CREATE TABLE USER_MANAGEMENT
  ( "USERNAME" VARCHAR(30) NOT NULL,
    "PASSWORD" VARCHAR2(30 ) NOT NULL,
    "EMAILID" VARCHAR2(50) NOT NULL,
    “SECRET_QUESTION” VARCHAR2(200) NOT NULL,
    “SECRET_ANSWER” VARCHAR2(200) NOT NULL,
    “USER_TYPE” VARCHAR2(20) NOT NULL
     CONSTRAINT " USER_MANAGEMENT_PK" PRIMARY KEY (USERNAME, PASSWORD)
);

CREATE TABLE CUSTOMER_MANAGEMENT
  ( "CUSTOMERID" VARCHAR(30) NOT NULL,
    "NAME" VARCHAR2(30 ) NOT NULL,
    "CONTACT" VARCHAR2(12) NOT NULL,
    “ADDRESS” VARCHAR2(200) NOT NULL,
    “EMAILID” VARCHAR2(50) NOT NULL
     CONSTRAINT " USER_MANAGEMENT_PK" PRIMARY KEY (CUSTOMERID)
);

CREATE TABLE SERVICE_PROVIDER
  ( "PROVIDERID" VARCHAR(30) NOT NULL,
    "SERVICEID" VARCHAR(30) NOT NULL,
    "NAME" VARCHAR2(30 ) NOT NULL,
    "CONTACT" VARCHAR2(12) NOT NULL,
    “ADDRESS” VARCHAR2(200) NOT NULL,
    “EMAILID” VARCHAR2(50) NOT NULL
     CONSTRAINT " SERVICE_PROVIDER_PK" PRIMARY KEY (PROVIDERID, SERVICEID)
);


CREATE TABLE CONSUMER_PROVIDER_RELATION
  ( "CONSUMERID" VARCHAR(30) NOT NULL,
    "PROVIDERID" VARCHAR(30) NOT NULL
    CONSTRAINT " CONSUMER_PROVIDER_RELATION_PK" PRIMARY KEY (PROVIDERID, SERVICEID)
);


CREATE TABLE SERVICES
  ( "SERVICEID" VARCHAR(30) NOT NULL,
    "SERVICE_DESC" VARCHAR(30) NOT NULL
    CONSTRAINT " SERVICES_PK" PRIMARY KEY (SERVICEID)
);


CREATE TABLE CAR_SERVICES_REGISTER
  ( "CONSUMERID" VARCHAR(30) NOT NULL,
   "PROVIDERID" VARCHAR(30) NOT NULL,
```

```
    "SERVICEID" VARCHAR(30) NOT NULL,
   "CARID" VARCHAR(30) NOT NULL,
    "CAR_MODEL" VARCHAR2(20) NOT NULL,
    "COST" DECIMAL(13,3) NOT NULL,
    “START_DATE” DATE NOT NULL,
    “END_DATE” DATE NOT NULL,
    “STATUS”  VARCHAR2(10) NOT NULL
     CONSTRAINT " CAR_SERVICES_REGISTER _PK" PRIMARY KEY (CONSUMERID, PROVIDERID,
SERVICEID, CARID, START_DATE)
);

CREATE TABLE CAR_INVENTORY
  ( "CARID" VARCHAR(30) NOT NULL,
   "PROVIDERID" VARCHAR(30) NOT NULL,
    "SERVICEID" VARCHAR(30) NOT NULL,
   "CAR_MODEL" VARCHAR(30) NOT NULL,
    “CAR_MAKE” VARCHAR2(20) NOT NULL,
    “MILEAGE” NUMBER(8) NOT NULL,
    “YEAR_MAKE” NUMBER(4) NOT NULL,
   "COST" DECIMAL(13,3) NOT NULL,
    “KM_RUN” NUMBER(10) NOT NULL,
    “STATUS”  VARCHAR2(10) NOT NULL
     CONSTRAINT " CAR_INVENTORY_PK" PRIMARY KEY (CARID, PROVIDERID, SERVICEID)
);

CREATE TABLE PART_INVENTORY
  ( "PARTID" VARCHAR(30) NOT NULL,
   "PROVIDERID" VARCHAR(30) NOT NULL,
    "SERVICEID" VARCHAR(30) NOT NULL,
   "NAME" VARCHAR(30) NOT NULL,
    “DESCRIPTION” VARCHAR2(100) NOT NULL,
    "COST" DECIMAL(13,3) NOT NULL
     CONSTRAINT " CAR_INVENTORY_PK" PRIMARY KEY (PARTID, PROVIDERID, SERVICEID)
);


CREATE TABLE BILLING_DATA
  ( "CONSUMERID" VARCHAR(30) NOT NULL,
    "CARD_DETAIL" VARCHAR(30) NOT NULL,
     “PASSWORD” VARCHAR(30) NOT NULL
    CONSTRAINT " BILLING_DATA_PK" PRIMARY KEY (CONSUMERID)
);
```

UML Class Diagram



High Level Architecture Diagram



Web Services Design

Web services APIs

1. ArrayList<String> getCarModelByProvider(String model);
2. ArrayList<String> getCarPartsByProvider (String parts);
3. ArrayList<String> getOrdersbyCustomer (String order);
4. ArrayList<String> getInvoicebyCustomer (String invoice);
5. ArrayList<String> getInvoiceWeeklySummary (String summary);
6. ArrayList<String> getCarInventoryReport (String report);
7. ArrayList<String> getCarPartInventoryReport (String city);
8. ArrayList<String> getConsumerRegisterSupplier (String register);
9. ArrayList<String> getCarStatus (String car status);
10. ArrayList<String> getPartStatus (String part status);