CSC 540 - DATABASE MANAGEMENT CONCEPTS & SYSTEMS

FALL 2022 PROJECT 1 - DATABASE APPLICATION DESIGN & IMPLEMENTATION

Draft Project Description - **Due Nov 2nd.**

AUTOR - Auto Repair and Service Management System

Version: 9/12

Introduction

The goal of the project is to design a relational database application for supporting a chain of car repair and service centers by a company. The project should be carried **out in teams of four (4)** unless prior permission is obtained from the instructor**,** and team peer assessments will be part of the overall grade for each student. The project is expected to be executed in stages with interim deliverables submitted, as described in this document.

The description has several details that you should pay attention to. First, it describes the data and application requirements. It then gives information about deliverables and tentative deadlines (deadlines may change slightly) and "getting started" guidelines.

Notes:

1. **You should necessarily assume that this is an imperfect description and will be subject to updates**. Therefore, it is important that you read through the description in the coming days and ask questions to clarify any missing or ambiguous statements.
2. In addition to revisions to help clarify description, **supplementary information about application flow and a set of queries that must be implemented, final submission instructions** will be added to the description in the coming weeks.
3. **You are responsible for testing your application's correctness using test data that you craft**. However, for the demonstration, all teams will use the same data to help keep uniformity in assessing team projects. We will provide this test data about a week to the demos.

Project Specification

Background and Overview

Downtown Auto Car Service and Repair Centers is a chain of car service centers owned by Downtown Auto Inc spread across the United States. They service and repair Honda, Nissan, and Toyota, Lexus and Infiniti cars.

Your task is to design a database system, AUTOR, that allows Downtown Auto corporation and its customers to keep a record of and manage the following information for all its repair centers.This description provides an indication of what important information (italicized) will need to be stored in the database as attributes and relationships. Suggested attributes for the different entity types are listed in the application; however, these suggestions are by no means exhaustive so you may find that you need to introduce additional attributes beyond the ones explicitly mentioned in this document in some cases.

Service Center details

Each service center is identified by *a globally unique ID, address, and a telephone number*. There might be multiple service centers in a given state in the country but each center is independently managed. Each center operates 5 days a week (M-F) from 8 AM to 8 PM. Some (not all) are also open on Saturdays from 9am - 1pm.

The general employee structure in each center has a *manager* who manages all employees, a *receptionist,* and several *mechanics*. For each employee, we store their locally unique *9-digit employee ID, name, address, email address, phone number, and the role at the service center they work at*. (Each employee is associated with only one service center). While the manager and receptionist are contract employees with fixed annual salary, mechanics are hourly paid workers. Each center has its own hourly rate for mechanics.

Each center offers a list of car services e.g. Oil Changes, Check Engine Light Diagnostics & Repair, Engine Repair, Catalytic Converter Repair, Alternator Repair & Replacement, Compressor Repair & Replacement. These services are loosely categorized into two broad categories: a *repair service* or a *maintenance service*. Some services like *Brake Repair* are both maintenance and repair services while others like *Oil Change* are exclusively maintenance services. Repair services are categorized into 6 main subcategories: (i.) *Engine Services* which include individual services like *Belt Replacement*, *Engine Repair*; (ii.) Exhaust Services which include individual services like *Catalytic Converter Repair*, *Muffler Repair*; (iii.) Electrical Services which include individual services like *Alternator Repair*, *Power Lock Repair*; (iv.) Transmission Services which include individual services like *Axle Repair*, *Transmission Flush* (v.) Tire Services which include individual services like *Tire Balancing*, *Wheel Alignment* and (vi.) Heating and Air Conditioner Services which include individual services like *Compressor Repair*. Individual services belong to only one of the 6 repair service subcategories (although as said before they could also be classified as a maintenance service).

Individual services have a globally unique number and name. Each service also has a price and a time estimated for the service job, which is based on the car and the specific auto center. For example, at one center, brake repair in a Honda may cost $50 and take 3hrs while the brake repair for the Lexus may cost $95 and take 4hrs at this same location. At another center, the estimates and times may be different.

Maintenance services are usually provided in “bundles" called schedules. Schedules include a set of individual services in a bundle that are completed together and are priced as a bundle. For example, a maintenance schedule may include *Oil Changes*, *Brake Repair*, *Check Engine Light Diagnostics*, etc. There are three bundles or schedules, Schedules A, B and C. Each schedule has a specific subset of services that are covered and there is a downward inclusion relationship so that Schedule B contains all the services in A and some extra ones and likewise Schedule C contains everything in B and some additional ones. As with the individual services, the pricing and time estimate is dependent on the type of car e.g. Honda vs Toyota and the particular center. Maintenance services are usually done in a rotational manner i.e. after a *Schedule A* maintenance service, the next maintenance will be *Schedule B*, then *Schedule C*. After *Schedule C*, it restarts at *Schedule A*.

Each mechanic works no more than (50hours a week). A mechanic may ask for and may be given time off, as long as there are always at least 5 mechanics present at any given time.

Customer details

A *Customer* has *id* (an integer) that is unique with respect to a specific center, a *first* and *last name*. A customer is associated with at least one vehicle which is identified by globally unique *vin number* (8 alphanumeric characters), *car manufacture*r e.g. Honda, and *current mileage* (integer), *year*. For each customer service event, the services rendered by which *mechanics* and *total amount charge*d and *total amount paid* are recorded. Each customer has a *status* field stating if their account is in good standing or not (i.e. has an outstanding balance).

Car Service Scheduling

A *Customer* can come in for either a scheduled maintenance or a specific repair service or both (for example if additional car components beyond those being serviced require repair). They initiate a request for services by adding the vin number of the car being repaired, the ids of any repair services they want and/or the scheduled maintenance service that they want. For example, a request for a car repair service may include a *Schedule A* maintenance and additional repairs like *Muffler Repair* and *Catalytic Converter Repair*. If the car is not on record at the shop, information about it like the Manufacturer, Current mileage and year are entered first. For cars that have a record in the shop, if based on records the scheduled maintenance selected is not the correct one that is next on the schedule, a prompt is used for the customer to confirm change. Once all selected services are checked/validated, a cost estimate and numbered list of available times is given (up to 1 month ahead). This list of available times is based on total estimated time of job to be done, availability of mechanics. Once the customer selects one of the available times, the relevant number of mechanics are denoted as “booked” for that period. In general, 1 mechanic is needed for a Schedule A service, 2 for Schedule B and 3 for Schedule C. Scheduling will need to avoid double-booking or over-booking mechanics i.e. > 50hrs, or booking them while they are on vacation. Also, no services are booked for 12 - 1 which is lunch break hour. However, a service may span before and after lunch.

For each service event, an *invoice* with unique *invoice id*, *customer id*, *vin of car serviced*, *date of service*, *services provided*, *cost for each service*, *mechanics* that provided the service and *total bill*l (sum of costs) is stored.

High Level Application Flow Requirements

The application should support the correct flow i.e. appropriate sequence of application steps and at each step the appropriate function options, for the appropriate role. Below we describe some of these functions. Some of these functions may require the use of advanced features like Triggers, Stored Procedures, Views, etc. The application can be driven by text-based menus which present numbers next to menu options and users can just enter the desired number for the menu option they desired. (GUIs are not required but are welcome if you wish to). Below description of an example flow. In the coming weeks, we will be providing you a supplementary document providing similar information to the one below to encourage a uniform application flow across projects.

You are also encouraged to include reasonable error handling code e.g. prompt for retry if user enters incorrect input.

Display	Menu	Input	Output
Display the menu	1. Login 2. Sign Up 3. Exit	Enter Choice (1-3)	Go to the appropriate page. If exit is chosen, terminate the program.

Login

Display	Menu	Input	Output
Ask user to input the following details in the order shown below, followed by the menu. A. User ID B. Password	1. Sign-In 2. Go Back	Enter Choice (1-2)	If the user chooses 1, validate credentials and recognize if user is a Manager, Receptionist, or Customer to go to the correct **Landing page**. Print "Login Incorrect" for invalid credentials and ask to enter again. If the user chooses 2, go back to the **Home page**

The following gives an overview of the functionality required for the different roles.

Admin

This is the overall application admin account. It is used to add a new store to system.

- **System Set Up:** This is procedure that sets up all the general information that is universal to all stores like service categories, services, their duration (not prices because that is defined per store).

- **Add New Store**: This menu should allow admin enter two major information items about the store id (either autogenerated or given), address and store manager information (first name, lastname, username, password, salary, and employeeid) and the store's minimum and maximum hourly wage for mechanics. This automatically creates the managers account for that store.

- **Add New Service:** Allows admin to add information about a new service they are offering. Categories will remain fixed, but a new service in a category can be introduced.

For simplicity, in all the cases where accounts are created for other users, they will use their last name as default password. We will also not bother with the resetting password after the user has logged in for the first time.

Manager

When logged in as manager, the manager will be able to add other store employees i.e. receptionist and mechanics along with their relevant information

- **Setup Store**: (these can be done in any order. However, only after the store has all necessary components including minimum number of mechanics, is it considered *available* i.e. operational and can accept customer requests to schedule services. While the process is still going on, store is still in *pending* mode

    - **Add employees**. Add information for employee including role, salary (or hourly wage). There should be only one receptionist per store. Once these employees are added, accounts are automatically created for them and they should be able to login.
    - **Select Operational Hrs**: Indicate if the store is open on Saturdays or not.
    - **Set Up Service Prices**:

- **Add New Employee:** A manager should be able to add additional employees later also after initial setup.

Receptionist

The account for receptionist can be created only by the manager. When logged in as a receptionist, the homepage displays the following options:

- **Add New Customer Profile:** Adds basic information for a customer which must include at least one vehicle. This generates a customer account that a customer can log into and update.
- **Find Customers with Pending Invoices:**

Customer

The account for customers shall be created from the application. When logged in as a

customer, the homepage displays the following options:

- **Update Profile:** The logged in customer should be able to update his profile information.
    - **Add Car to Profile:** A customer should be able to register a car with a service center. He should be allowed to only register cars that belong to the list of manufacturers that Downtown Auto works on.
    - **Delete Car from Profile:** When customer sells car they can delete car from profile.
- **View and Schedule Service**: This will have the following sub-menus:
    - **Schedule Service**: This subflow should allow a customer enter the list of service codes (either one after the other or as a set). These indicate the services the customer is interested in. This can include both a maintenance service and some repair services. This procedure will randomly select mechanics from the list that will be available at that time. Scheduling is done in blocks of hour multiples i.e. 8am – 9am for a 1hr service or 8am – 10am for a 2hr service (except lunch hour).
    - **View Service History**: A customer should be able to see service history for his car using the car's vin number. ● **View and Pay Invoices**:
    - This should display a numbered list of invoices and status e.g. *1.) Invoiceid1, paid*. When a customer clicks on the number, they can see the details of the invoice. If the invoice is unpaid, the detailed view also presents a prompt to pay, with yes, no. If yes is selected, the invoice is considered paid.

Mechanic

The account for receptionist can be created only by the manager. When logged in as a

receptionist, the homepage displays the following options:

- **View Schedule**: This will have the following sub-menus
- **Request TimeOff:** Should not be approved if already assigned that timeslot or if number of mechanics will fall below threshold. If not, place in manager's approval queue.
- **Request Swap:** one mechanic can request another to either swap or cover a timeslot. If it is to cover not to swap, the request will only be added to the list of the mechanic being requested if it does not cause him to be overbooked (>50hrs) or a conflict with existing assignment. Only one mechanic can be requested at a time.
- **Approve Swap:** Each mechanic can see their list of pending requests and select approve. Once approved the schedule is changed to reflect the changes.

Sample Queries (TBA)

Project Deliverables

Milestone 0 - Questions on the Forum - on or before Sep 26th

Post any questions for clarification of the project description on the forum

Milestone 1 - Report, Due October 3rd

For the first milestone, you should:

1. Fill in the form for deciding team members as soon as possible.
2. A partial ER-Diagram covering the entity and relationship types and constraints that you have identified so far relevant to Employees, Stores and Services.
3. A translation of your partial E-R model into SQL.
4. Include a table that has two columns. One column has a phrase in the description document that you have identified as a constraint and the other column says how your model so far has captured the constraint e.g. foreign key. You do not need to include those that you have not captured yet since some constraints will require all of SQL to capture.
5. Also include up to 5 functional dependencies that you have identified so far.

Milestone 2 - Final Report. Nov 2nd.

Each team will have to book a timeslot to demo the application to the Professor or Teaching Assistants. The details about the demo will be conveyed later. The following need to be submitted. Submission instructions for final report will be given later.

Grading Rubric (TBA)

Getting Started

I would recommend using an IDE(like Eclipse or IntelliJ) for developing the project and some form of version control like GitHub for sharing project amongst team members. (You can create private repositories on NCSU Github). Using Oracle's SQL-Developer would also help you in writing long queries, triggers and procedures because of advanced features like debugging and static analysis of query.

The following link will help you to connect to the database using the JDBC Driver: Creating a connection using JDBC. Students are encouraged to read more about proper handling of connection and JDBC Best Practices.

Note that points will be deducted for **improper handling** of connection in the java application.