[**Prompt**](http://web.csulb.edu/~dbrown/CECS323/Projects/Automotive%20Repair/CECS%20323%20Term%20Project.pdf)

Phase 1 - Nov 13

**Phase 1 Final - Nov 27, 2017**

**1. The description of your five additional business rules. Please call this BusinessRules.docx, txt, … (Trong Nguyen) rev1: Khai Phung**

**Business Rules**

* Rule 1 : If a steady customer chooses to upgrade their membership to prospective customer. We will give them 3% cash back of their total annual cost.
* Rule 2: We will keep track of the experience of mechanics in years. We also classify cars for servicing into 3 categories based on their difficulty : Easy Maintenance, Medium Maintenance, Hard Maintenance. The following procedure must be followed:
* For Easy Maintenance, a mechanic who has over 1 years of experience can conduct the service.
* For Medium Maintenance, a mechanic who has over 2 years of experiences can conduct the service.
* For Hard Maintenance, a mechanic who has over 3 years of experience can conduct the service.
* Rule 3: For the servicing of a maintenance item, we have a number of mechanics requirement. For example, some basic services, like an oil change, may require only one mechanic while more complex services such as engine cleaning or interior cleaning may require 2 or more mechanics to service the vehicle.
* Rule 4: Some services will require other services to be performed to be performed in order. For example, Cleaning an engine requires an oil change first because we have to dispose the old oil. Another example would be requiring a vehicle alignment after doing tire changes. Some auto shops do free alignment when customer have their tires changed, but this is not what happens at Dave’s Automotive)
* Rule 5: Customers can have their vehicle serviced before the recommended maintenance due date should they so choose. Some customers want to have their car maintained before a long trip or perhaps it fits their schedule better to do the service earlier. However, we also need to update other future service intervals to follow it (other intervals will come sooner than the previously expected time)

**2. The explanation of your denormalization and how you are going to enforce data integrity in spite of the redundancy of the denormalized structure. Please call this document Denormalization.docx, .txt, … (Khai Phung)**

<https://docs.google.com/spreadsheets/d/1LIrqDQIYc61cRlViQvnke2x7wGe9O4Kzb25ueVOWoW8/edit?usp=sharing>

We combine all attributes of Employee, Mechanic, Skills, Mentorship into one table The redundancy helps employer to easily organize his employee. Just by looking at one table, he can get knowledge of worker’s personal information and his skills. In addition, employer also knows which mechanic is teaching another.

We have firstname, lastname,phone as primary key because they are short, meaningful and easy to remember. We also include address of the employee in the table so the employer can look them up if he wants to send mails or contact a mechanic’s family when that mechanic got hurt. The address must is in detail. We have housenumber, streetname,city,state, zipcode. We separated them into sections because it helps finding a specific street/city/state/etc easier. It also prevents user from typing words stick together. We also include masterCertification collumn in the table. Master certification is earned when a mechanic has certifications in all important mechanic skills. We need this column to easily find out which one is the elite mechanic

Some error might happen because of misunderstanding. In the skill box, most of the time there will be more than one data. A mechanic always have multiple skills. To prevent user from enter multiple skill in one row, we create a column called skillnumber, which force the user to enter a new row everytime he wants to input another skill. This helps the database recognize single skills. Not just that, it also eliminate word sticking (words stick together by typo). There is also a column “Certification”for the user enter the certification he earned for a mechanic skill. We don’t need multiple certification rows because 1 certification for each skill is enough

We also created extra columns for the mentee. It has its own firstname, last name,mentoringskill and mentee’s phone so the employer don’t have to look at another table to find them. It can be null because some mechanics don’t have mentee

**3. The normalized UML class diagram – either as a DIA model or draw.io. If you have a use, please check with me first. (Stephen)**

**different tool that you would like to**

**FULL IMAGE:** [**LINK**](https://drive.google.com/file/d/1f3DDxMijq4uWcO2IKKxWuRzEFzk5UXBT/view?usp=sharing)

**4. English description of all classes and associations. Please call this ClassAndAssociationDefinitions.txt, or docx, or … (Hanson)**

**Class and Association Definitions**

1. Customers - An entity that has signed up to receive services from Dave’s.
2. IndividualCustomer - An individual who has signed up to receive services from Dave’s.
3. CorporateCustomer - A company that has signed up to receive services from Dave’s.
4. AddressType - Is a type of location or place of residence where a company is situated.
5. AddressLine - Describes the exact location of an address owned by a company
6. CurrentCustomer - An active customer of Dave’s that is eligible to refer additional customers.
7. SteadyCustomer - A person that has signed up for the generated email notifications.
8. EmailNotification - A generated notification sent to a SteadyCustomer.
9. PremierCustomer - A person who pays on a monthly basis to receive special services from Dave’s.
10. ProspectiveCustomer - A potential candidate to become a current customer.
11. OfferHistory - A recording of the time when an offer was given.
12. SpecialOffer - A unique benefit or perk provided by Dave’s Automotive.
13. Vehicle - A motorized transport brought in by a customer for maintenance.
14. MaintenanceVisit - A record of a meeting between the automotive shop and a customer for the purpose of servicing a vehicle
15. VehicleRepairInterval - A description of the preventive maintenance standard defined by Dave’s Automotive for the make and model of a vehicle.
16. VehicleServiceList - A pairing of a list of services for a designated vehicle repair interval
17. Employee - An individual who is works for Dave’s Automotive in exchange for a salary.
18. ServiceTechnician - An employee who writes up service reports
19. MaintenanceServiceList - A compilation of all the repair tasks to be performed
20. MaintenancePackage - A combination of one or more services
21. PackageItems - A record of the ownership of an item within a maintenance package
22. MaintenanceItem - A specific service that can be performed
23. TaskSequence - The ordered relationship between service items to perform
24. Mechanic - An employee who performs repair work on vehicles
25. MechanicSkills - The specific pairing of a mechanic and a skill they possess.
26. Mentorship - A recorded session of a skill tutoring between mechanics.
27. Skills - A specific ability or knowledge that can be learned and taught.
28. ItemSkills - The specific skill required for a maintenance item
29. VisitService - The set of tasks to be performed for a specific visit
30. MechanicTask - An assignment of a mechanic for maintenance at a customer visit

**Associations:**

Each CorporationCustomers has one or many AddressType.

Each AddressLine has one and only one CorporateCustomer

Each AddressLine is owned by one and only one AddressType

Each AddressType owns one or many AddressLine

Each CurrentCustomers recommends zero or many ProspectiveCustomers.

Each ProspectiveCustomers are recommended by one and only one CurrentCustomers.

Each SteadyCustomer is mailed by zero or many EmailNotification.

Each EmailNotifaction mails one and only one SteadyCustomer.

Each SteadyCustomer receives zero or many OfferHistory

Each OfferHistory is offered to one and only one SteadyCustomer

Each PremierCustomer receives zero or many OfferHistory

Each OfferHistory is offered to one and only one PremierCustomer.

Each ProspectiveCustomer is mailed one or many OfferHistory.

Each OfferHistory mails to one and only one ProspectiveCustomer.

Each SpecialOffer tracks one and only one OfferHistory.

Each OfferHistory tracked one and only one SpecialOffer.

Each EmailNotification indicates one and only one VehicleRepairInterval

Each VehicleRepairInterval indicates one or many EmailNotifications.

Each Customer owns one or many Vehicles.

Each Vehicle is owned by one and only one Customer.

Each Vehicle is applied one and only one VehicleRepairInterval

Each VehicleRepairInterval applies to one or many Vehicles

Each VehicleRepairInterval lists one or many VehicleServiceList

Each VehicleServiceList listed one and only one VehicleRepairInterval

Each VehicleServiceList is listed by one and only one MaintenanceServiceList.

Each MaintenanceServiceList lists one or many VehicleServiceList

Each Vehicles is brought in for zero or many MaintenanceVisit

Each MaintenanceVisit brings in one and only one Vehicle

Each ServiceTechnician writes up one or many Vehicles

Each Vehicle is written up by one and only one Technician

Each MaintenanceVisit has one and only one VisitService

Each ServiceVisit has zero or many MaintenceVisits

Each MaintenanceService List performed one and only one VisitService

Each VisitService performs one or many MaintenanceServiceList

Each Maintenance Package contains one or many packageItems.

Each Packageitem contained in one and only one maintencepackage

Each MaintenanceItems is in one or many PackageItems.

Each PackageItem has one and only one MaintenceItem

Each MaintenanceItems perform before one and only one TaskSequence.

Each TaskeSequence is performed after one and only one MaintenceItem.

Each Mechanic performs one or many MechanicTask

Each MechanicTask was performed by one and only one Mechanic

Each VisitService is performing one or many MechanicTask.

Eah MechanicTask is performed at one and only one VisitSerivce

Each Mechanic participates in one or many Mentorship

Each Mentorship is participated by one and only one Mechanic

Each Mentorship mentors one or many MechanicSkills.

Each MechanicTable is mentored by one and only one Mentorship.

Each Mechanic has knowledge of one or many MechanicSkills

Each MechanicSkills is known by one and only one Mechanics

Each MechanicSkill attributes to one and only one Skill

Each Skills is attributed from one or many MecchanicSkills

Each Skills is required for one or many ItemSkills

Each ItemSkills is needed for by one and only one Skills

Each MaintenanceItems requires one or many ItemSkills

Each ItemSkills is required for one and only one MaintenceItems

***Individual Feedback:***

RED = DONT UNDERSTAND

GREEN = DONE

ORANGE = NOT SURE

Team 5 -

Your mechanic class has the mechanicID showing up in the UML. That looks like a surrogate key to me, which says that this doesn't belong in the UML. Your MechanicSkills class has skillNumber in it, which looks like it's a surrogate key to me. Again, not appropriate in the UML. In fact, I'm seeing migrating foreign keys and surrogate keys all through the model. Don't do that.

Mentorship is not normalized. The menteeFirstname, menteeLastname, menteePhone are all functionally dependent just on the menteeID. So that constitutes a subkey. Do you see how to solve that?

Your Certified? Boolean in MechanicSkills is in the wrong place. Remember that Certified? depends upon both the mechanic as well as the skill. Find a better place in your model for this.

I realize that this is preliminary, but don't forget to put the categorization constraints on your categorization relationships.

Clever idea to have a many to many between CorporateCustomer and AddressType. I like it. The only problem that I see is that the diamond at the CorporateCustomer end of that association is not supported in the graphical syntax of the many to many association.

Remember, a category class, like ProspectiveCustomers, cannot be the child in any identifying relationships. Which means that the composition association from CurrentCustomers to ProspetiveCustomers is wrong. You need that association, but it cannot be a composition.

You cannot assign a Mechanic to a MaintenanceItem. Instead, you need to find a way to assign a Mechanic to a MaintenanceItem in the context of a particular MaintenanceVisit.

Your many to many without history between Vehicle and Service Technician isn't really going to work. The way that it stands, a given vehicle would never be able to have the same service technician again. I'm going to say that the association from Vehicle to MaintenanceVisit is identifying, but that Service Technician should come in on a **non** identifying association since the technician association is more informational in nature.

The many to many without history between MaintenanceAction and MaintenanceVisit makes good sense. But you cannot use a composition at either end of that, the language doesn't support that.

I don't see how you are going to enforce the constraint that a given mechanic has to have at least one of the skills needed for a given MaintenanceItem in order for them to be assigned to that MaintenanceItem.

I don't see how you are tracking the redemption of the rewards for referring a prospective customer.

our numRecommendations in CurrentCustomers is derivable. You can count the number of ProspectiveCustomers who have the given currentCustomer's ID as their referring customer.

Your estMileagePerYear in SteadyCustomers is misplaced. Think of it this way. A given customer could have a number of vehicles. Each vehicle will need maintenance done on it at regular intervals, based on the mileage that the customer will put on **that vehicle** in a given year. Find a better place for the estimated mileage.

This is a cosmetic thing, but SpecialOffer is a lookup table, which is a great idea. But just make that class "above" the OfferHistory so that it's clearer that it's a parent.

I believe that deadProspect is a derivable value as well. You could create a view that will count the number of contacts for a given prospective customer and deduce whether they are "dead" if you wanted to encapsulate that logic. (I put numberofContact in prospectiveCustomer class)

The work that has to be done on a given vehicle depends on the model, make, year and the mileage. You need to find a way to capture what is needed at each interval, independently of a particular vehicle, and associate those maintenance requirements to a Vehicle Model instead. That would be far less redundant.

I don't see how you are classifying cars by maintenance difficulty. That is something else that would depend on the model, make and year, and not the specific vehicle. (I put Difficulty attribute in VehicleRepairInterval class)

I don't see how you are supporting business rule 3.(I put numberOfRequiredMechanic attribute in MaintainnanceItem class)

Rule 4 would be wonderful. It's recursive. I don't see any evidence of that in your model.(I creat the table MaintainanceFreeOffer)

How are you going to implement Rule 5? I put TimeBetweenMaintainance in the VehicleRepairInterval class and the derivative attribute nextMaintainanceVisit in the EmailNotification class

**Phase 2 - December 11**

* The DDL(Create table statement) create\_table.sql.

CREATE TABLE customers (  
 first\_name VARCHAR(20) NOT NULL,  
 last\_name VARCHAR(20) NOT NULL,  
 phone VARCHAR(20) NOT NULL,  
 street VARCHAR(50),  
 zipcode VARCHAR(5),  
 CONSTRAINT customers\_pk PRIMARY KEY (first\_name, last\_name, phone));

Create table individualCustomer

Create table corporateCustomer

Create table addressType

Create table currentCustomer

Create table steadyCustomer

Create table emailNotification

Create table premierCustomer

Create table prospectiveCustomer

Create table offerHistory

Create table specialOffer

Create table vehicleRepairInterval

Create table vehicle

Create table serviceTechnician

Create table maintainanceAction

Create table maintainancePackage

Create table maintainanceItem

Create table mechanic

Create table mechanicSkills

Create table mentorShip

* The Relation Scheme diagram relation.xml,

Draw on the same link of the uml model [**LINK**](https://drive.google.com/file/d/1f3DDxMijq4uWcO2IKKxWuRzEFzk5UXBT/view?usp=sharing)

* The Views create\_view.sql.

A view is simply any SELECT query that has been given a name and saved in the database.

CREATE VIEW customerView AS  
 SELECT employeeID, empFName, empLName, jobTitle, managerID FROM Employees;  
 GRANT SELECT, UPDATE ON job\_view TO managers;

CREATE VIEW mechanicView

CREATE VIEW mentorshipView.

CREATE VIEW currentCustomerView

CREATE VIEW prospectiveCustomerView?

CREATE VIEW maintainanceVisitView

CREATE VIEW \_\_

CREATE VIEW \_\_

CREATE VIEW \_\_

* The DML (The insert statements used to populate the tables) insert.sql.

INSERT INTO orders (cfirstname, clastname, cphone, orderdate, soldby)  
 VALUES ('Tom', 'Jewett', '714-555-1212', '2005-09-11',  
'Patrick');

* Queries

1. List the customers. For each customer, indicate which category he or she fall into, and his or hercontact information. If you have more than one independent categorization of customers, please  
indicate which category the customer falls into for all of the categorizations.  
2. For each service visit, list the total cost to the customer for that visit.  
3. List the top three customers in terms of their net spending for the past two years, and the total  
that they have spent in that period.  
4. Find all of the mechanics who have three or more skills.  
5. Find all of the mechanics who have three or more skills in common.  
a. Please give the name of each of the two mechanics sharing 3 or more skills.  
b. Please make sure that any given pair of mechanics only shows up once.  
6. For each maintenance package, list the total cost of the maintenance package, as well as a list of  
all of the maintenance items within that package.  
7. Find all of those mechanics who have one or more maintenance items that they lacked one or  
more of the necessary skills.  
8. List the customers, sorted by the number of loyalty points that they have, from largest to  
smallest.  
CECS 323 Term Project.docx 1/11/2017 10:51:00 AM 6  
9. List the premier customers and the difference between what they have paid in the past year,  
versus the services that they actually used during that same time. List from the customers with  
the largest difference to the smallest.  
10. Report on the steady customers based on the net profit that we have made from them over the  
past year, and the dollar amount of that profit, in order from the greatest to the least.  
11. List the three premier customers who have paid Dave’s Automotive the greatest amount in the  
past year, and the sum of their payments over that period. Be sure to take into account any  
discounts that they have earned by referring prospective customers.  
12. List the five model, make, and year that have caused the most visits on average to Dave’s  
automotive per vehicle in the past three years, along with the average number of visits per  
vehicle.  
13. Find the mechanic who is mentoring the most other mechanics. List the skills that the mechanic  
is passing along to the other mechanics.  
14. Find the three skills that have the fewest mechanics who have those skills.  
15. List the employees who are both service technicians as well as mechanics.  
16. Three additional queries that demonstrate the five additional business rules. Feel free to create  
additional views to support these queries if you so desire.

* Each attribute in the UML model must have a description for it in Phase 2. Please call this  
  AttributeDescription.docx, .txt, …

??

First name is the first word in a person’s name?

* Trigger Code

The MySQL trigger is a database object that is associated with a table. It will be activated when a defined action is executed for the table. The trigger can be executed when you run one of the following MySQL statements on the table: INSERT, UPDATE and DELETE and it can be invoked before or after the event.

mysql> delimiter //  
mysql> CREATE TRIGGER agecheck BEFORE INSERT ON people FOR EACH ROW IF NEW.age < 0 THEN SET NEW.age = 0; END IF;//  
Query OK, 0 rows affected (0.00 sec)  
mysql> delimiter ;

Steven: availability  
Monday: Morning til - 2pm  
Tuesday: Morning til - 2pm  
Wednesday: Morning til - Noon  
Thursday: Morning and 8pm after  
Friday: All day  
Saturday: after 3pm   
Sunday: ??   
Wed = morning

Trong after 5 everyday