Car Sales Web Portal

Interim Report 2

Team Members

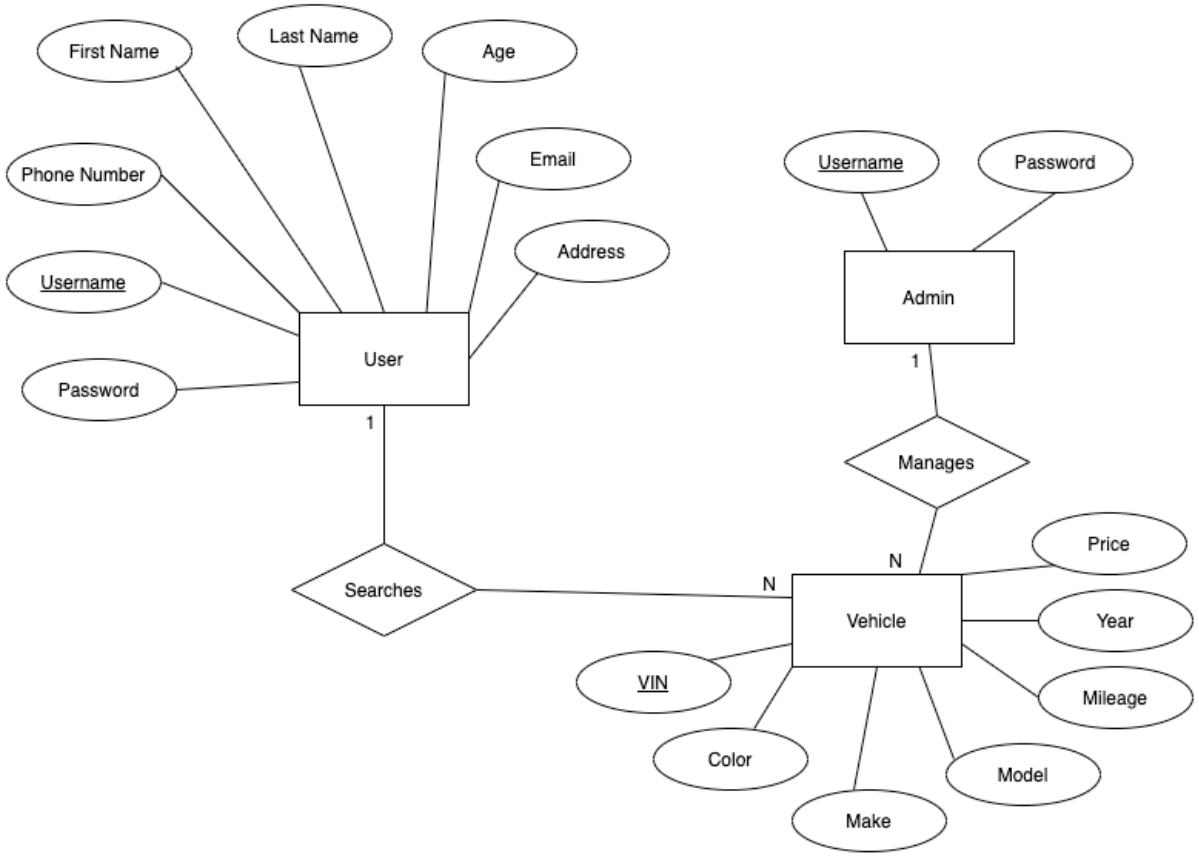
John Michael Hayde: jhayde@clemson.edu

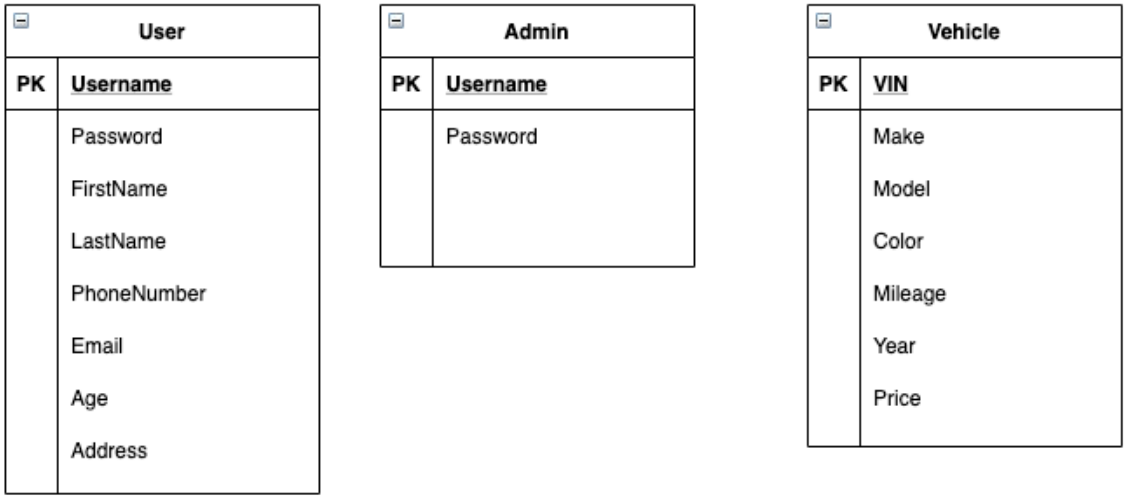
Changes/Updates to Proposal

I am not making any changes to my original proposal. While I have made some changes to my ER Diagrams, I feel that my proposal idea is good and have enjoyed working on building the portal.

Final Version of ER Diagrams and Tables

In the vehicle ER Diagram, I realized that I had left a very important attribute: price. This is very important to know when shopping for vehicles, so I added it into the ER Diagram and corresponding Table. The final versions of the Diagrams and Tables are shown below.





SQL Statements

I have developed SQL statements for all necessary database functionality. They are broken up into 6 major categories: initializing the tables, loading initial data into the tables, inserting a dingle entry into a table, making a query on the table, updating information in the table, and removing information from the table. The statements are listed below.

1. **Initializing the tables**
    a. **User**

```
DROP TABLE IF EXISTS user;
CREATE TABLE user (
  username    VARCHAR(20),
  password    VARCHAR(20),
  phone_num   VARCHAR(20),
  first_name  VARCHAR(20),
  last_name   VARCHAR(20),
  age         INT(3),
  email       VARCHAR(40),
  address     VARCHAR(50)
);
```

b. Admin

```
DROP TABLE IF EXISTS admin;
CREATE TABLE admin (
  username  VARCHAR(20),
  password  VARCHAR(20)
);
```

c. Vehicle

```
DROP TABLE IF EXISTS vehicle;
CREATE TABLE vehicle (
  vin      VARCHAR(20),
  color    VARCHAR(15),
  make     VARCHAR(20),
  model    VARCHAR(20),
  mileage  INT(20),
  year     INT(4),
  price    INT(10)
);
```

2. Loading initial data into the tables

a. User

```
LOAD DATA LOCAL INFILE 'user_sample.txt' INTO TABLE user;
```

b. Admin

```
LOAD DATA LOCAL INFILE 'admin_sample.txt' INTO TABLE admin;
```

c. Vehicle

```
LOAD DATA LOCAL INFILE 'vehicle_sample.txt' INTO TABLE vehicle;
```

3. Inserting a single entry into a table

a. User

```
INSERT INTO user VALUES ('jhayde', 'password', '123456789', 'John', 'Hayde', 22,
'jhayde@clemson.edu', '219 Cedar Creek Circle, Sunset, SC 29685');
```

NOTE: the following statement is what I have developed on the portal so add a new user to the table when they create an account on the website.

INSERT INTO user VALUES('$_POST[username]', '$_POST[password]', '$_POST[phone_num]', '$_POST[fname]', '$_POST[lname]', '$_POST[age]', '$_POST[email]', '$_POST[address]');

b. Admin

INSERT INTO admin VALUES ('root', 'password');

c. Vehicle

INSERT INTO vehicle VALUES ('2FMDK3KC7BBA78129', 'Black', 'Ford', 'Edge', 100000, 2011, 8000);

4. Making a query on the table

The vehicle SQL statement will be altered when it is implemented into the "Search For Vehicles" webpage so that the user can select any or all fields to search the database. I have not yet gotten to that part of the portal build, so I am unsure what the variables will look like and how the SQL will be written on that page.

a. User

SELECT * FROM vehicle WHERE name = jhayde;

b. Admin

SELECT * FROM vehicle WHERE name = 'root';

c. Vehicle

SELECT * FROM vehicle WHERE make = 'Ford';

5. Updating information in the table

These are a sample of how to update each table. I have not yet decided how to handle letting the user update their profile. I may have a form with all of their information filled out, and then let them edit anything, and when they submit it, update all of their information. Or, I may let them select which specific information they would like to change, and edit each field at one time.

a. User

UPDATE user SET password = 'new_pass' WHERE username = 'jhayde';

b. Admin

UPDATE admin SET password = new_root_pass WHERE username = ‘root;

c. Vehicle

UPDATE vehicle SET price = 14000 WHERE make = 'Toyota' AND model = 'Tacoma';

6. Removing information from the table

a. User

DELETE FROM user WHERE first_name = ‘John’ AND last_name = ‘Hayde’;

b. Admin

DELETE FROM admin WHERE username = ‘jhayde’;

c. Vehicle

DELETE FROM vehicle WHERE vin = ‘3TMMU4FN8CM049190’;

Current Progress

Currently, I have generated small data sets and tested my SQL statements with those on both my local platform and the production platform. I have written scripts to generate large amounts of data and have tested loading that data into the tables locally. I have tested my SQL statements against the large data locally.

I have made good progress on the portal. I have developed the homepage, which prompts a user to select one of 3 options: returning user, new user, or admin. The returning user page is written in html and prompts the user to enter their username and password. If this info is correct, it redirects them to the homepage. The new user page asks for all of the users information that is required, verifies the tables is filled out correctly and the passwords match, and stores that information into the user table. This page is built in php. When submit is clicked on the form, ‘add_new_user.php’ is called, which is responsible for adding the information to the table. The admin login simply allows an admin to login. I have mocked this page up this in html.

Tasks to Be Completed

I still have to do a number of tasks. These include:

1. Converting the admin and returning user login pages to dynamically check the login information provided with the data stored in the tables.

2. Generate the large data sets on the production platform and test my SQL statements on the production large scale data.
3. Create a “logged in” homepage that gives the user 3 options: search the database, edit their information, and log out
4. Create the aforementioned pages, especially focusing on the database query page.
5. The database query page will need to display results that the user specifies. These inputs will be the fields of the table that holds the vehicles.
6. Create an admin “logged in” page and give them options to add/remove users, add/remove vehicles, add/remove admin users, and backup the data.
7. Implement a form of data backup for the whole database, decide where this data gets stored (locally, sent to someone via email, posted on a server somewhere).
8. If I have time, use CSS and JavaScript to make the portal look better.

Problems

I have not yet run into many problems. I initially struggled to get the database setup on the production platform, but now that is up and running. It has also taken me some time to get used to PHP and how it interacts with HTML elements, but I am getting the hang of that as well. The only true problem I have left to tackle is how to back up the database. I have never done this before and have found multiple different methods to do so. I have to decide which option will be the best for my system and implement it.