Unit 4: Database Design & Development- Assessment 1

Design specification

Client: Hire_From_Us

By: Aaron Edge (113612)
Date: 26/02/2018

Introduction

This is a design specification for Hire_From_Us. This document's purpose is to provide a design framework to build a software solution for a car hiring system.

This is a server / client combination system that allows users to easily carry out all the data collection from the customer and securely store the data on the database. The system will keep track of cars that are available to rent and the price to rent them out.

The software solution will add the capability to analyse sales data and calculate sales assistants commission. It is designed to run on the provided windows workstation and use the installed java runtime environment.

User Interface design

Familiarity for user

The graphical user interface should closely mimic the look and feel of the existing paper based system. This can be achieved through laying out the input fields in the same order as they appear on the paper based system.

The sales staff at Hire_From_Us are familiar with conventional user input. Using traditional forms of input like text fields and comboboxes. Will require no additional training on how to use them.

Consistency and standards

Button and other gui elements, should have a consistent look and feel over the entire application. The actions should also be consistent, if the user presses a button they expect it to do the same thing every time.

Error prevention

Measures to prevent user error should be in place within the graphical user interface. Combo boxes with preset values will be used where possible. Check boxes can be used for yes or no inputs.

Users will be prevented from proceeding to the next section of the data entry process if required fields are left empty.

Error Reporting

Feedback in the form of error messages with a short description of the error should be given to the end user. For example if the user has missed a field the system should prompt them with a message telling them what field is empty, or if they

have incorrectly entered their password they system should tell them that they have entered the wrong password.

Data validation

Data validation is the process of checking that input data (usually from the end user) is exists and is usable. The validation process involves checking the input data against a set of predefined rules.

There are several types of data validation such as:

Format check: Ensures that the data is in the expected format.
Length check: Ensures that the data is not too short or too long.
Presence check: Ensures that data is actually entered.
Range check: Ensures that the data falls within a specified range.
Type check: Ensures that the data entered is the correct data type.

One of the main reasons software engineers implement data validation is to help reduce human error. Highlighting the mistake and providing useful feedback to the user is a vital part of the validation process.

Input validation can also be used to prevent malicious attacks on the system, preventing attackers from entering data that could potentially cause damage.

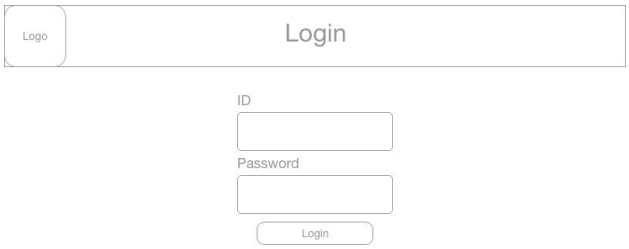
It is good practice to perform data validation as early in the process as is possible. Preferably as soon as it is entered. All data from untrusted sources should be subject to some form of data validation.

Diagrams of proposed screens

Login Screen:

This will be the first screen visible to the end user and the first point of interaction. This screen will allow authorized username and password combinations to proceed to the rest of the application.

It will have feedback to the user when invalid details are provided.



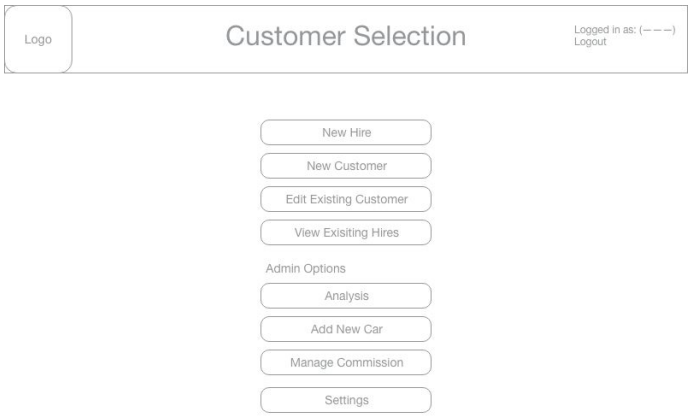
Main menu screen:

This screen will present the user with a choice of tasks they would like to start. From here the sales assistant can:

- Start a new hire.
- Create a new customer.
- Edit existing customers.
- View existing hires.

Accounts with higher permission levels will see these additional options:

- View sales analysis data.
- Add a new car to the database.
- Manage sales assistants commission.
- Adjust application and organizational settings.



Customer selection screen:

Once the user has clicked the “New hire” button they will start the process of creating an new hire order. The customer selection screen is the first screen in this process.

This screen will be used to select the customer that is hiring the car. The screen will consist of search field on the left and a table displaying the resulting customers on the right.

At the bottom of the page there will be a next and back navigation system to move the user backwards and forwards through the process. The user will be prevented from going forward in the process until a customer has been selected.

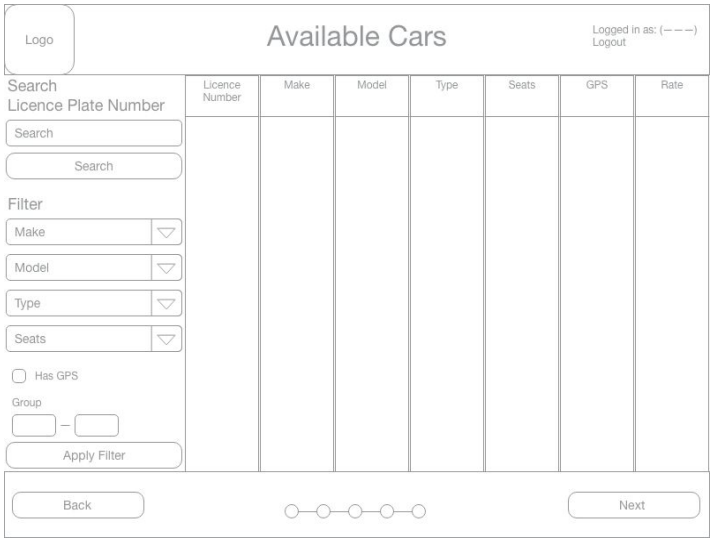
Date selection screen:

The next step in the process is the date selection screen. From here the user will be able to select the date that the hire started and the date that the hire will end.

Available cars screen:

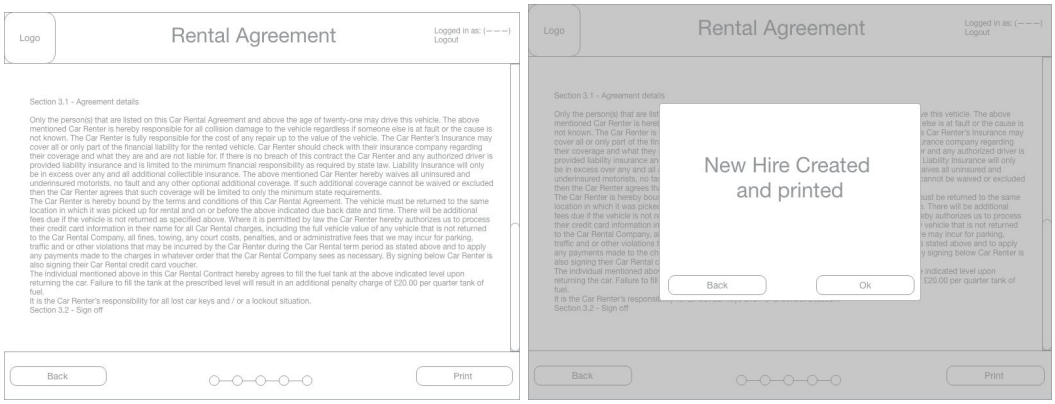
Next step is to select what car the customer would like to hire. To aid with car selection a filter will be provided on the left hand side. The user can filter the results based on several factors, like if the customer needs a gps fitted to the car the user can click the checkbox and the system will only display cars with gps fitted. Several of these filter options will be available like: model, make and number of seats.

The user will also have the ability to search for specific licence plate numbers.



Rental Agreement screen:

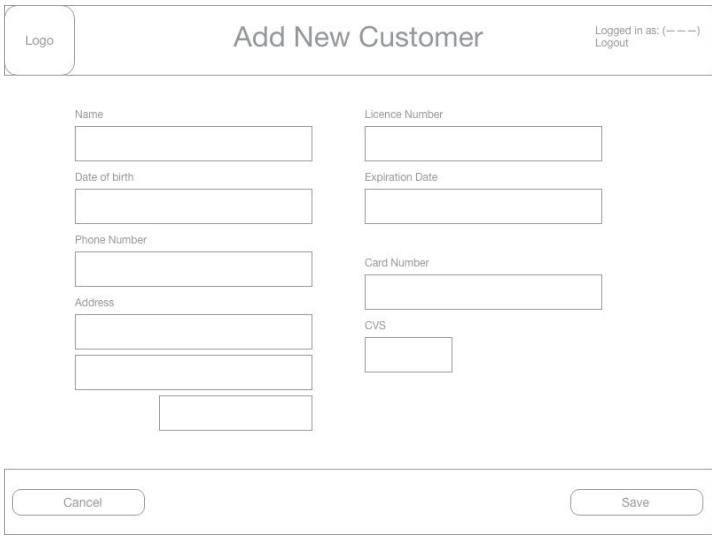
The last screen in the create new hire process is the rental agreement screen. This screen will consist of a scrollable textarea displaying the generated invoice for review by the sales assistant.



Add new customer screen:

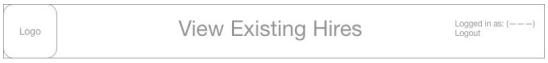
This screen will be accessible from the main menu or from the customer selection screen.

From this screen the user can enter all of the necessary customer details into the input fields. Once they have completed the data entry they can press save, the system will then check the validity of the data and check to see if any fields have been missed. If there is anything wrong with the data the system will provide feedback to the user in the form of an error message. Once the system is happy that all the necessary data is present and valid it will save the new customer to the database.



View existing hire screen:

This screen is accessible through the main menu screen. From here the user can view edit and delete existing hires.



Manage cars screen:

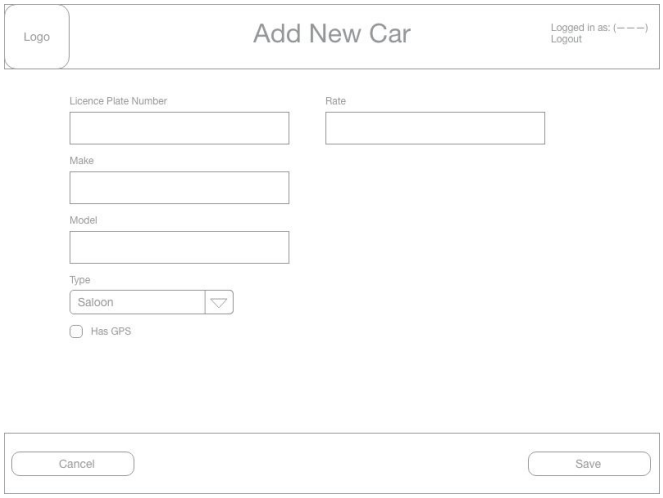
This screen is only available to accounts with management permission levels and is accessible through the “manage cars” button on the main menu.

From this screen the user will be able to manage the details about the fleet of rental cars. They will have the ability to update car details and delete cars from the database.

Add new car screen:

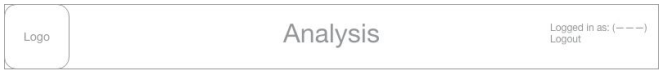
This screen is only available to accounts with management permission levels and is accessible through the “add new car” button on the car management screen.

From this screen the user can enter details about new cars.



Sales analysis screen:

This screen is only available to accounts with management permission levels and is accessible through the “analysis” button only visible to management accounts on the main menu.



Settings screen:

This screen is only available to accounts with management permission levels and is accessible through the “settings” button only visible to management accounts on the main menu.

Manage users screen:

This screen is only available to accounts with management permission levels and is accessible through the “manage users” button only visible to management accounts on the settings screen.

From this screen user management functions can be performed. The manager can update user details and delete users.

Add new user screen:

This screen is only available to accounts with management permission levels and is accessible through the “add new user” button only visible to management accounts on the manage users screen.

From this screen the management can add new users to the system setting their username and password for their login and also setting the permission level of the account.

System Design

Structure and relationships

Server Application

- Implemented in Java
- Uses SQL statements to perform operations on the database.

Client Application

- Implemented in Java.
- Sales team interact with the client application only.
- The client application is used to send SQL queries to the server

Hardware platform

This project will require a central server connected to a local area network. The server will be running an SQL database implemented and run by netbeans.

The server will be locked inside a secure case that only the management staff have access to.

The case will be well ventilated and stored in a secure part of the building. The server will be connected to a network switch by cat 5 cables. The switch will also be used to connect to the workstations.

The workstations will be basic machines built to meet microsoft windows 10 minimum specification, so that they can run the windows 10 operating system.

Each workstation will need to be supplied with a mouse and keyboard for input and a screen to display the applications graphical user interface.

A networked printer that has also been connected to the switch using cat 5 cables is also required to print the invoices.

Application platform

The client workstations will have windows 10 operating system installed on them as well as java runtime environment.

The server will have netbeans installed to run and manage the SQL database.

Development

For this project java is the chosen language and netbeans is the integrated development environment.

The code will be written using a combination of event driven and object oriented programing paradigms. This so that classes can be easily modified later and reused if any enhancements to the software is needed.

Processes

- Create, read, delete and update customer data
- Create, read, delete and update user data
- Create , read, delete and update rental car data
- Populate the required information on the rental agreement from the database.
- Calculate the commission to pay each sales assistant.
- Calculate the total cost of rental over a given period
- Add up the profits from the monthly sales to be displayed as sales data.

Algorithms

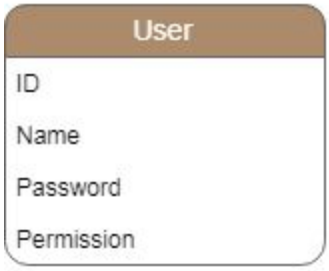
SQL statements will be used to interrogate the database. Getting back the desired results from the database will involve using the “SELECT” and “WHERE” clauses.

The algorithm for calculating the total rental price is: (rental rate * rental duration) + insurance cost

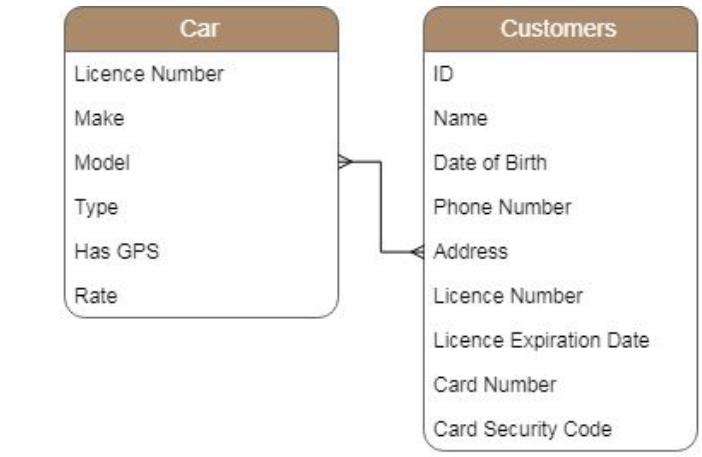
algorithm for calculating the commission is: total rental price * 0.05

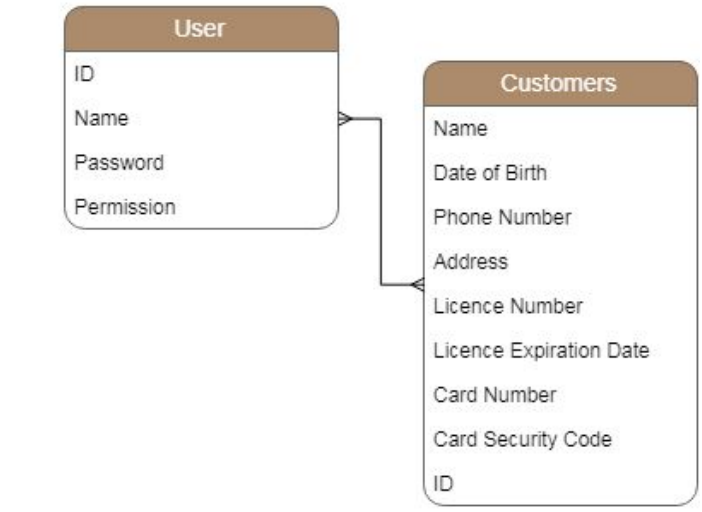
Database Design

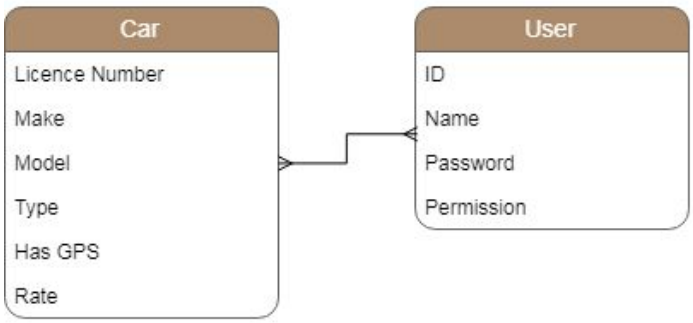
Tables



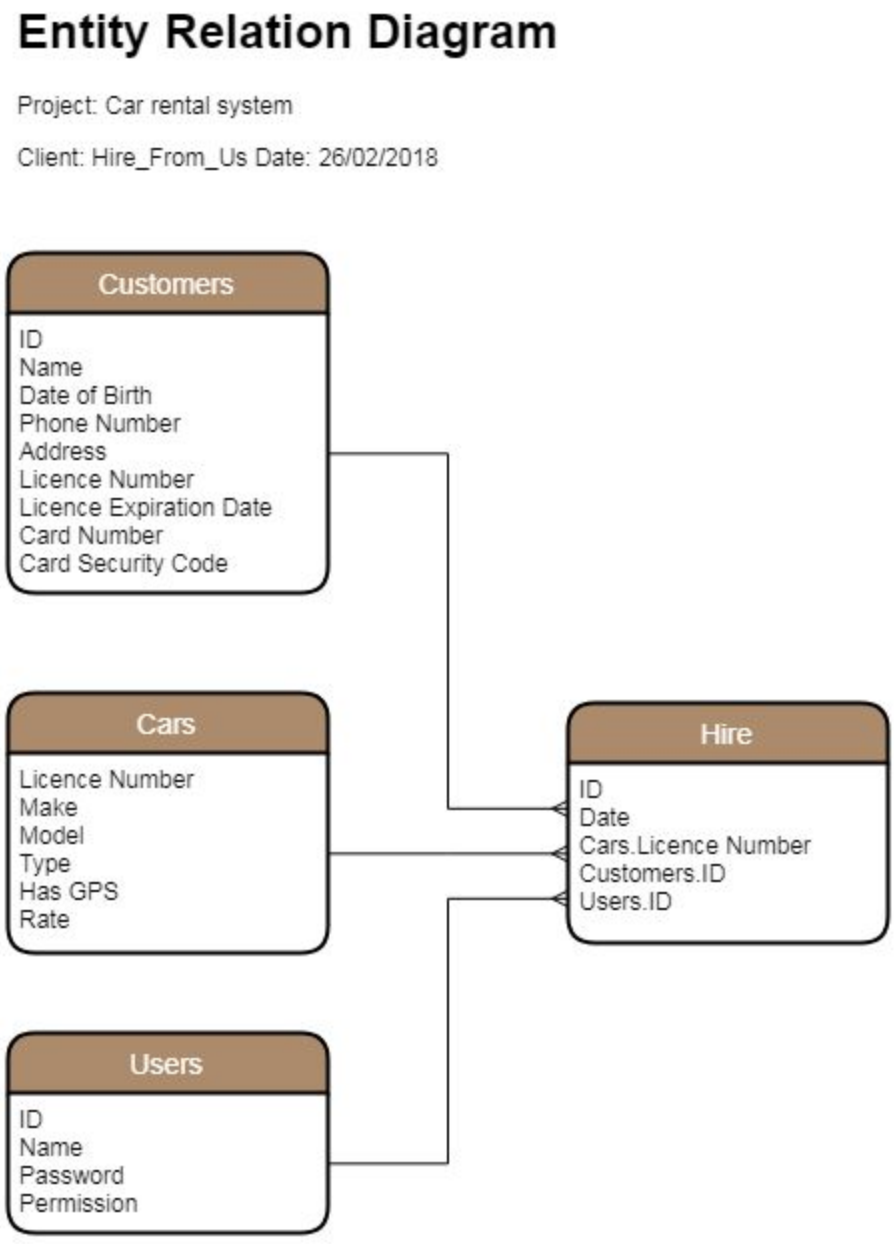
NF2


Car
Licence Number
Make
Model
Type
Has GPS
Rate
Customers
ID
Name
Date of Birth
Phone Number
Address
Licence Number
Licence Expiration Date
Card Number
Card Security Code


User
ID
Name
Password
Permission
Customers
Name
Date of Birth
Phone Number
Address
Licence Number
Licence Expiration Date
Card Number
Card Security Code
ID


Car
Licence Number
Make
Model
Type
Has GPS
Rate
User
ID
Name
Password
Permission

NF3


Entity Relation Diagram
Project: Car rental system
Client: Hire_From_Us Date: 26/02/2018
Customers
ID
Name
Date of Birth
Phone Number
Address
Licence Number
Licence Expiration Date
Card Number
Card Security Code
Cars
Licence Number
Make
Model
Type
Has GPS
Rate
Hire
ID
Date
Cars.Licence Number
Customers.ID
Users.ID
Users
ID
Name
Password
Permission

SQL statements

Create Tables SQL:

```
CREATE TABLE `Customers` (
    `ID` INT(8) NOT NULL AUTO_INCREMENT,
    `Name` VARCHAR(64) NOT NULL,
    `Date of Birth` DATE NOT NULL,
    `Phone Number` INT(32) NOT NULL,
    `Address` VARCHAR(256) NOT NULL,
    `Licence Number` VARCHAR(128) NOT NULL,
    `Licence Expiration Date` DATE(128) NOT NULL,
    `Card Number` INT(32) NOT NULL,
    `Card Security Code` INT(8) NOT NULL,
    PRIMARY KEY (`ID`)
);

CREATE TABLE `Cars` (
    `Licence Number` VARCHAR(16) NOT NULL,
    `Make` VARCHAR(64) NOT NULL,
    `Model` VARCHAR(64) NOT NULL,
    `Type` VARCHAR(64) NOT NULL,
    `Has GPS` BOOLEAN(1) NOT NULL,
    `Rate` DECIMAL(32) NOT NULL,
    PRIMARY KEY (`Licence Number`)
);

CREATE TABLE `Users` (
    `ID` INT(16) NOT NULL AUTO_INCREMENT,
    `Name` VARCHAR(64) NOT NULL,
    `Password` VARCHAR(8) NOT NULL,
    `Permission` INT(8) NOT NULL,
    PRIMARY KEY (`ID`)
```

```
);

CREATE TABLE `Hire` (
    `ID` INT(16) NOT NULL AUTO_INCREMENT,
    `Start Date` DATE NOT NULL,
    `End Date` DATE NOT NULL,
    `Licence Number` INT(16) NOT NULL,
    `Customer ID` INT(16) NOT NULL,
    `User ID` INT(16) NOT NULL,
    PRIMARY KEY (`ID`)
);

ALTER TABLE `Hire` ADD CONSTRAINT `Hire_fk0` FOREIGN KEY
(`Licence Number`) REFERENCES `Cars`(`Licence Number`);

ALTER TABLE `Hire` ADD CONSTRAINT `Hire_fk1` FOREIGN KEY
(`Customer ID`) REFERENCES `Customers`(`ID`);

ALTER TABLE `Hire` ADD CONSTRAINT `Hire_fk2` FOREIGN KEY (`User
ID`) REFERENCES `Users`(`ID`);
```

Appendix

https://www.draw.io/

http://markup.su/highlighter/