Term Project for Fall 2013

YouDrive: Car Rental Service

Home
Contact
Short Vita
Research
Teaching
LSDIS Lab
Links

Problem Statement

A prospective client of my software company asked me to develop a computer system to support his new *car rental service* for young drivers. The client's company serves a small market and only has fewer than a hundred cars. As the traditional car rental companies usually do not rent cars to people under 25, my client decided to start renting cars to young people, using very flexible rental agreements (older drivers will be able to rent, as well).

The rental company will have no permanent staff and the system will only be available online. Drivers should be able to register with the company and approve their driving credentials (usually a driver's license and an address). The registration requires a 6-month membership fee (renewable). Once registered, drivers should be able to request a vehicle for a certain date and time to be picked up at a specific location. Furthermore, the user should be able to select one of three vehicle types: a regular car, a pickup truck, and a luxury car (at a premium cost). If a requested car is not available at a desired location, the system should check if a similar car is available at an alternate location. Currently, my client's company has 7 locations, but more may be added in the future.

Rental locations are just parking lots where the vehicles are available for pickup. Picking up and returning vehicles is on a self-serve basis. Registered users should be able to rent vehicles by the hour, with the maximum of 72 hours (3 days). The rental price should be adjustable, based on the length of the rental.

The administrator of the system should be able to define and maintain a number of vehicle types, enter individual vehicles and specify their types and other important properties.

Different vehicle types should have different per hour prices, defined by the administrator. The administrator should also be able to define rental locations and assign vehicles to them. Furthermore, the administrator should be able to remove a user (customer) for serious violations of the rental agreement.

After a few initial meetings with the customer, we have agreed to create the software system for vehicle rentals. Our initial set of the requirements outlining the concept of such a system is presented later in this document.

As the development of the system must be rushed, we have agreed on the following deadlines for various deliverables, listed in the table below. The contract for the development of the system will be signed once the requirements document is ready.

DELIVERABLE	POINTS	DEADLINE
Requirements document	20	Sep 11, 2013
Analysis document	20	Oct 3, 2013
System Design document	15	Oct 17, 2013
Object Design document	25	Nov 7, 2013
Implementation and System Demonstration	40	Dec 3, 2013

Table 1: Project Deadlines

The system outline: high-level requirements

1. The system must allow the *system administrator* to define and enter into the system *vehicle types*, such as small car, full-size car, truck, or a luxury car. Since vehicles are rented per hour, the administrator must be able to set a *rental price* for each *vehicle type*.

Furthermore, the price should be settable for *hourly* and *daily* rentals (1 day costs less than 24 hrs.). Also, the administrator should be able to set the price the *6-month membership*.

2. The administrator should be able to enter *rental locations* into the system. Each rental location should have a *name*, *address*, and a *vehicle capacity* (the maximum number of vehicles it can hold). A number of vehicles (see below) are assigned to each rental location.
3. The system must allow the administrator to define and enter into the system individual *vehicles*. A *vehicle* should have a defined *vehicle type*, and a number of properties, such as the *make* and *model*, *year*, registration *tag*, current *mileage*, and the time it was *last serviced*. Also, each vehicle's condition is specified (*good, needs cleaning, needs maintenance*, etc.). Each vehicle should be *assigned* to a *rental location*.
4. The administrator should be able to *make changes* to any of the information currently stored in the system. For example, it should be possible to change rental prices, reassign vehicles to different locations, modify vehicle properties, etc. It should be possible to *remove* vehicles, rental locations, etc.
5. A rental system user (a customer) should be able to *register with the system*. To do that, the user must establish the *user name* and *password*, and then provide his/her *driver's license state and number*, *email address*, *residence address*, and a *credit card information* to be used for payments. The user must pay the initial *6-month membership fee*. The user should be able to *modify* this information and extend his/her membership.
6. It should be possible to browse and search rental locations and vehicles there, as well as vehicles alone.
7. The user should be able to place a *reservation* for a vehicle at a selected *rental location*. The reservation must specify a *vehicle type*, vehicle *pickup time* and the *length of the rental*. The system should check if the requested vehicle would be available at the requested time and place and create a reservation. If a request cannot be granted, the system should suggest a similar

rental vehicle at a different location.

8. The user should be able to *cancel an existing reservation* up to one hour ahead of the scheduled pickup time. Otherwise, a minimum charge of one-hour rental should be applied.
9. The user should *notify the system as soon as the car is returned* to the rental location. The user is charged for the vehicle time starting with the reservation time and ending at the return time. A late fee is $50 plus the additional hourly charge. The user may enter information about the condition of the returned vehicle. Also, the user should be able to provide comments about the vehicle and the rental service in general, if desired.
10. The user should be able to terminate the membership at any time. The membership fee is not returnable.
11. The administrator should be able to terminate the membership of a user, if necessary.
12. The system must be *accessible from a common Web browser* (such as the Mozilla Firefox, Google Chrome, and Microsoft Internet Explorer).
13. The system should provide *multi-user access*, assuring correct concurrent behavior. The system should maintain suitable *authorization* information and *validate access*. *User authentication* should be implemented (by checking *user id* and *password*).
14. The system must have an *easy-to-use user interface* (UI) with screens designed for each part of the system's functionality and suitable for different types of users (customers, administrators, managers).
15. The system should use a *persistent data store* (MySQL RDBMS) for all of the relevant data.
16. The system should use accepted *standards* whenever possible (HTML, CGI, ODBC, Servlet API, JDBC, SQL, etc.). The project *must* be coded in either C++ or Java, possibly including other scripting languages, such as PHP and JavaScript, if needed.